

The image features a blue header bar at the top. Below it is a photograph of a large stadium under construction, showing the tiered seating areas and the surrounding landscape. The text "SAP Business Workflow Eventos" is overlaid on the right side of the image.

# **SAP Business Workflow Eventos**

# Objetivos del Curso

## Objetivos

*Son objetivos de este curso*

### § Eventos

- § Entender que es un evento y como crear eventos
- § Comprender los distintos tipos de eventos
- § Saber como definir condiciones de inicio de eventos

### § Desarrollo

- § Saber desarrollar: programas que llaman eventos, check function modules y receiver function modules.

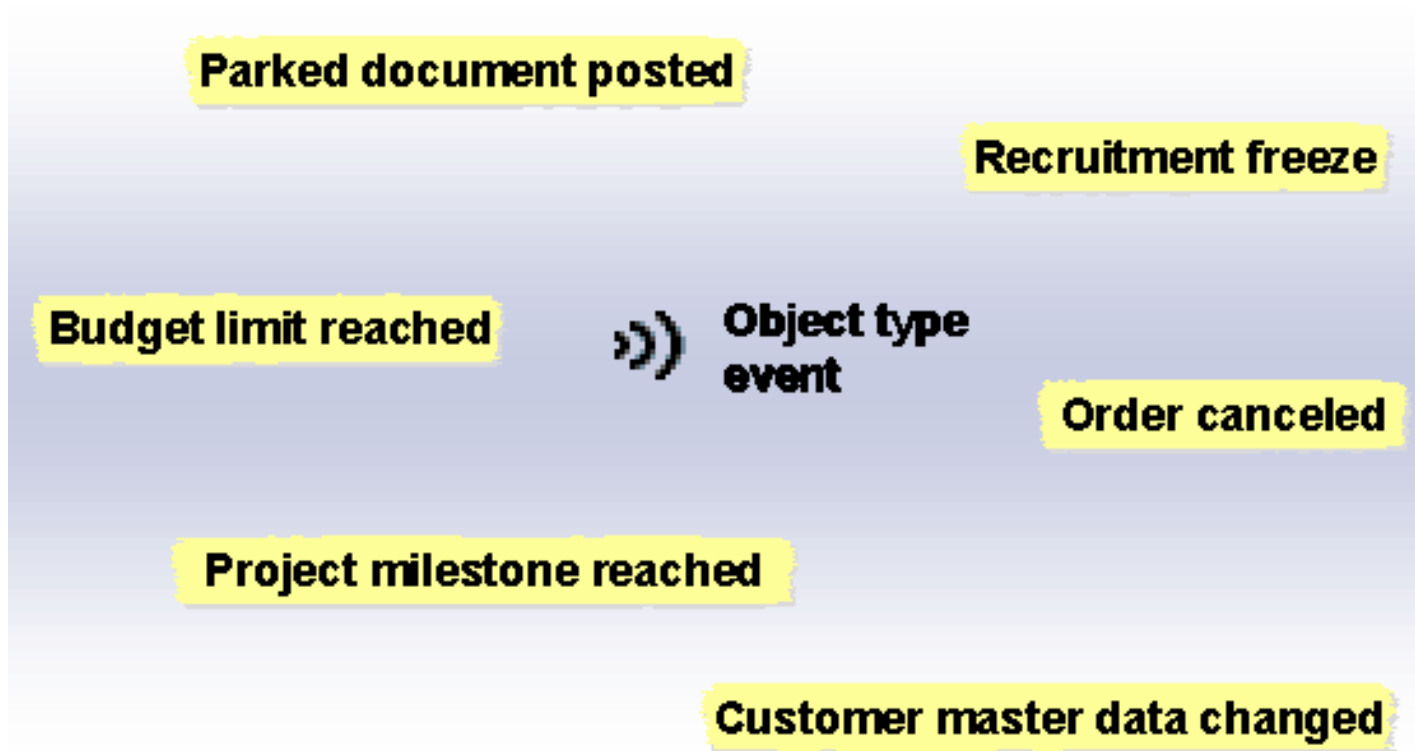
# Contenidos

- **Definición de Eventos**
- Lanzando eventos desde las aplicaciones SAP
  - Lanzando eventos con “change documents”
  - Lanzando eventos con mensajes
  - Lanzando eventos con cambios de status
  - Otros tipos de lanzamiento de eventos
- Relacionar un evento a un workflow
  - Definición en el Workflow Builder
  - Event Linkage
  - Condiciones de Inicio
- Desarrollos en Eventos
  - Llamar un evento de un programa
  - Desarrollo de módulos de funciones para eventos

# Definición de Eventos

- Dado que los workflows son procesos de negocio, **es vital** para una aplicación de negocio poder comunicarse con los workflows.
- Por ejemplo una aplicación de negocio necesita informar:
  - Cuando **comienza** un proceso de negocio
  - Cuando **termina** un proceso de negocio o una actividad dentro del proceso
  - Cuando una actividad o proceso que ha comenzado **ya no se necesita**
  - Cuando dada una circunstancia **ha cambiado el ambiente** en el cual el proceso se ejecuta.
- Para poder comunicarse la aplicación de negocio utiliza **EVENTOS**.
- Simplemente dicho un evento **es el cambio de estado de una instancia de un objeto de negocio** (Business Object).
- Por ejemplo cuando un usuario cambia el maestro de materiales para el material XXXY entonces el Business Object XXXY lanzara el evento “Changed”.

# Definición de Eventos



# Definición de Eventos

- Para usar un evento como interfase entre la aplicación y un workflow se necesita lo siguiente:
  - **Definición del Evento**
    - Es el nombre técnico del evento definido en un tipo de objeto. Se definen como un verbo en pasado (creado, modificado, liberado, etc.).
    - Además el evento esta definido por sus parámetros. Los parámetros por defecto de un evento son: su nombre, el tipo de objeto, la instancia del objeto y el creador del evento. No obstante se pueden definir parámetros adicionales que deben acompañar el evento.
  - **Creador del Evento**
    - Es el programa, workflow, persona que ha creado el evento.
  - **Receptor del Evento**
    - Es el termino genérico que se usa para denominar a todo aquello que reaccionará ante el evento. Normalmente son workflows o tareas de espera.
  - **Linkage del Evento**
    - El linkage especifica la relación entre el evento y su receptor. Se pueden a su vez especificar las reglas que gobiernan esta relación. Las reglas determinan cuando y como el receptor recibirá el evento.

# Definición de Eventos

- Creando un evento
  - Los eventos se crean en el Business Object Repository
  - Debemos especificar el tipo de objeto para el cual queremos crear el evento
  - Al definir eventos nunca deberemos codificar nada.
  - Los datos que deben ingresarse son
    - El nombre del evento
    - Los parámetros del evento
  - Podemos ver en el business object BUS2105 (solicitud de pedido) el evento “released” y veremos si definición y un parametro que posee asociado que es el codigo de liberación.

Visualizar el tipo de objeto BUS2105

Business object BUS2105 ✓ Solicitud de pedido

- Interfaces
- Campos clave
- Atributos
- Métodos
- Eventos
  - PurchaseRequisition.released Liberar solicitud de
  - Evento released

Evento: released

Tipo objeto: BUS2105

Release: released

Status: released

Textos

| Denominación | Tip. objeto | Rel. carácter |
|--------------|-------------|---------------|
| Released     | BUS2105     | 43            |

El objeto desencadenante

Parámetros

| Parámetro | Relación |
|-----------|----------|
| Released  | Released |

Propiedades de parámetro

Relación

Relación

| Relación | Relación |
|----------|----------|
| Released | Released |

Relación

| Relación | Relación |
|----------|----------|
| Released | Released |

Relación

| Relación | Relación |
|----------|----------|
| Released | Released |

# Contenidos

- Definición de Eventos
- Lanzando eventos desde las aplicaciones SAP
  - Lanzando eventos con “change documents”
  - Lanzando eventos con mensajes
  - Lanzando eventos con cambios de status
  - Otros tipos de lanzamiento de eventos
- Relacionar un evento a un workflow
  - Definición en el Workflow Builder
  - Event Linkage
  - Condiciones de Inicio
- Desarrollos en Eventos
  - Llamar un evento de un programa
  - Desarrollo de módulos de funciones para eventos



## Lanzando Eventos desde Aplicaciones SAP

- Antes de que un evento sea lanzado por una aplicación, la creación del evento debe programarse en el programa de la aplicación.
- Afortunadamente en muchos de los programas standard de SAP están ya definidos los programas que lanzan los eventos y solo es necesario realizar el event linkage y determinadas configuraciones de customizing (ver la documentación de cada modulo para saber como lanzar eventos – <http://help.sap.com>).
- No obstante puede que para un proceso de negocio particular **tengamos que crear un evento nuevo**. En este caso **deberemos definir como se lanzara el evento** a partir de la aplicación.
- En el caso que el evento deba lanzarse desde un programa propio (de cliente) podremos programar el lanzamiento del evento muy fácilmente utilizando las APIs que SAP provee para tal caso.

## Lanzando Eventos desde Aplicaciones SAP

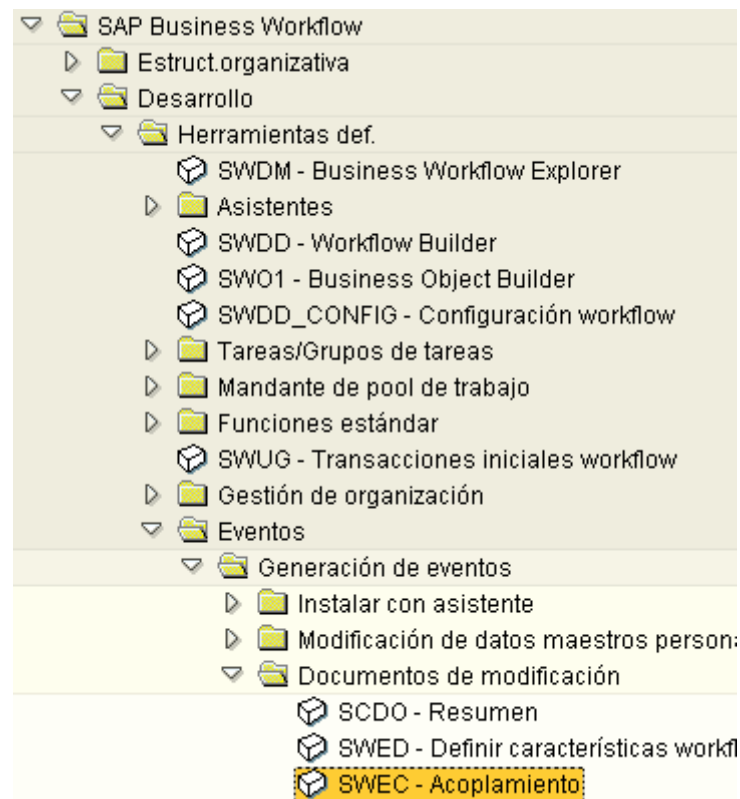
- En el caso que debemos lanzar un nuevo evento desde un programa standard de SAP tenemos las siguientes posibilidades:
  - A través de documentos de cambio (change documents)
  - A través del sistema de gestión de status
  - A través de control de mensajes
  - Utilizando el sistema de información logística (LIS)
  - A través de los datos maestros de HR
  - A través de Business Transaction Events (Solo para Finanzas)
  - A través de customizing específico de cada aplicación.
- Solo los tres primeros casos son los más usados, el resto son específicos para determinados módulos (HR – FI) y para casos aislados (por ejemplo la liberación de un pedido de compras).

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Change Documents**
  - Muchas aplicaciones de negocio en SAP utilizan documentos de cambio para dejar registro de las modificaciones hechas (generalmente transacciones de mantenimiento de datos maestros).
  - Los documentos de cambio definen la operación que provoca el cambio (modificación, creación o borrado) y registran los datos del objeto de negocio que ha cambiado en forma de tablas con el valor antiguo y el nuevo.
  - Los documentos de cambio SOLO se escriben cuando un campo designado como “*relevante para change document*” cambia.
  - Antes de definir un evento basado en un documento de cambio deberemos controlar que el cambio sera escrito como un documento de cambio, controlando el customizing de los campos o bien haciendo pruebas.

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Change Documents**
  - Para crear un evento de este tipo vamos por la siguiente opción de menú



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Change Documents**

- Debemos indicar,

- El **código de documento** de cambio
- El **business object**
- El **evento**
- **Bajo que actividad se lanzará**
  - Creación
  - Modificación
  - Borrado

- Luego podremos restringir aun mas bajo que circunstancias queremos que se lance el evento, especificando **campos de la tabla de campos relevantes, su valor antiguo y su valor actual.**

Modificar vista Eventos para doc.modificación: Resumen

| Objeto de negocio | Tip. de negocio | Evento      | Al crear                 | Al modificar             | Al borrar                |
|-------------------|-----------------|-------------|--------------------------|--------------------------|--------------------------|
| MEANF             | ECN             | DELETED     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| ANL4              | BU51022         | CHANGED     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| BNF               | BU52022         | PRERELEASED | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| BELE              | BESE            | CHANGED     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| BELE              | BNPF            | CHANGED     | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Modificar vista Restric.campo: Resumen

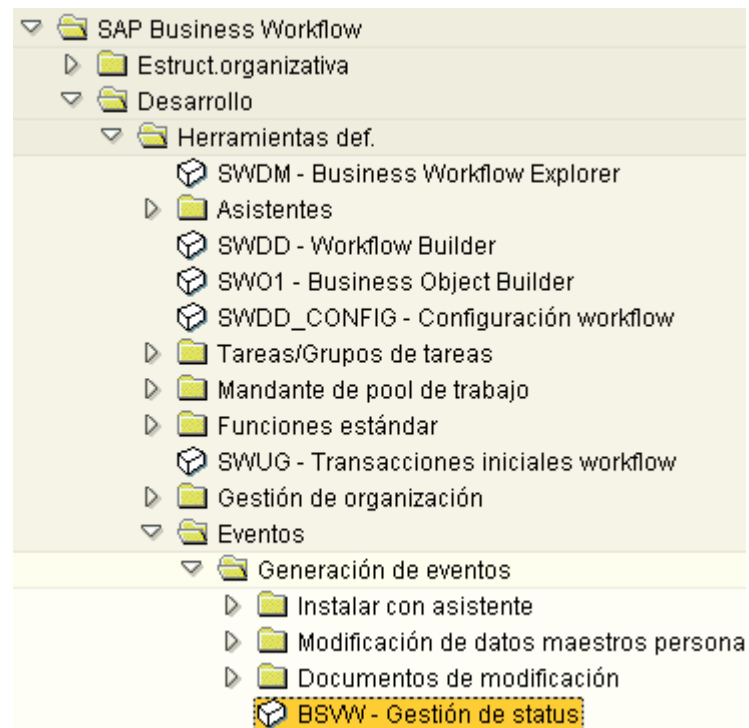
| Tabla | Nombre campo | Valor antiguo | Valor nuevo |
|-------|--------------|---------------|-------------|
| COAN  | TRM2         | X             | X           |

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Cambios de Status
  - Si una aplicación de negocio utiliza el sistema de gestión de status, podremos configurar el lanzamiento de eventos a partir de un cambio de estatus del sistema.
  - Las aplicaciones de negocio SAP que generalmente usan este sistema son las logísticas, principalmente PP, PM, PS, QM, etc.
  - El sistema estándar viene por defecto con status predefinidos llamados “status de sistema”, no obstante y por customizing pueden definirse nuevos status (de cliente).
  - Los status de sistema siempre son fijados por el sistema automáticamente, mientras que los de cliente tienen que ser fijados por el usuario.

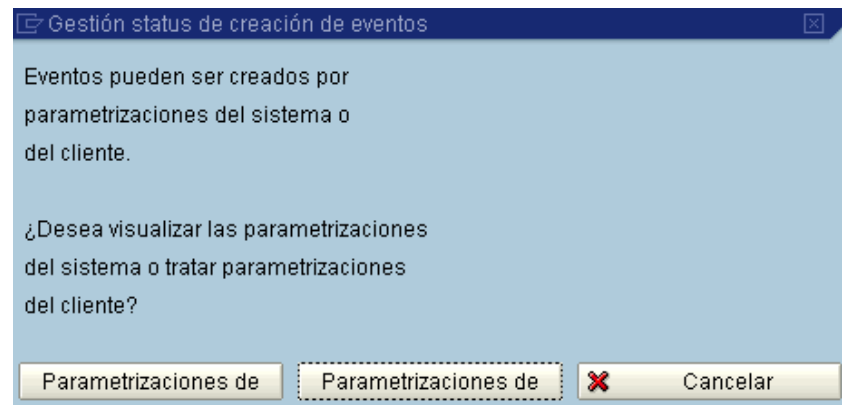
# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Cambios de Status
  - Para crear un evento de este tipo vamos por la siguiente opción de menú



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Cambios de Status
  - Primero debemos seleccionar con que tipo de status trabajar, de sistema o de usuario.

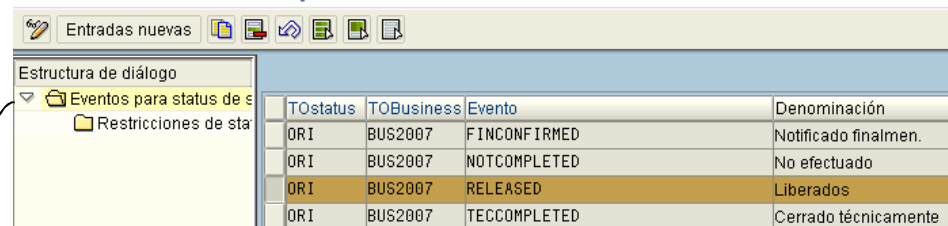




# Lanzando Eventos desde Aplicaciones SAP

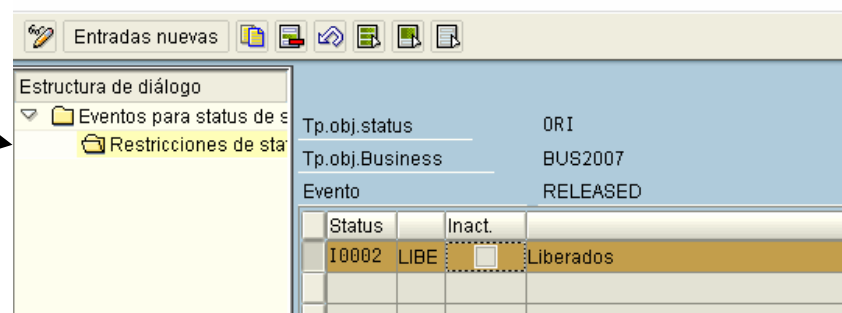
- Lanzando Eventos: Cambios de Status
  - Luego deberemos seleccionar el tipo de objeto y su evento .
  - Finalmente activarlo.

Modificar vista Eventos para status de sistema: Resumen



| TObstatus | TObBusiness | Evento       | Denominación         |
|-----------|-------------|--------------|----------------------|
| ORI       | BUS2007     | FINCONFIRMED | Notificado finalmen. |
| ORI       | BUS2007     | NOTCOMPLETED | No efectuado         |
| ORI       | BUS2007     | RELEASED     | Liberados            |
| ORI       | BUS2007     | TECCOMPLETED | Cerrado técnicamente |

Modificar vista Restricciones Back (F3) us: Resumen



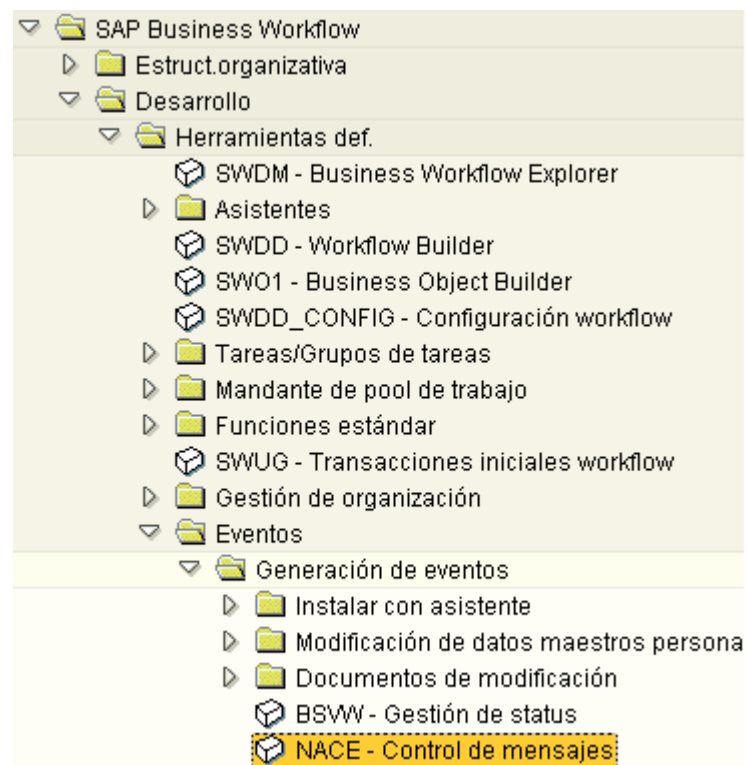
|                 |                          |           |
|-----------------|--------------------------|-----------|
| Tp.obj.status   |                          | ORI       |
| Tp.obj.Business |                          | BUS2007   |
| Evento          |                          | RELEASED  |
| Status          | Inact.                   |           |
| I0002 LIBE      | <input type="checkbox"/> | Liberados |

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Control de Mensajes**
  - Si una aplicación de negocios usa control de mensajes para intercambiar información entre los distintos involucrados en el proceso de negocio podemos configurar un mensaje para lanzar eventos.
  - Cuando el sistema de control de mensajes se ejecute, cualquier mensaje configurado será lanzado.
  - Por ejemplo al crear una orden de ventas o un pedido de compras se utilizan mensajes para imprimir la orden. También podremos usar el mismo sistema para lanzar eventos.
  - El tipo de mensaje que se utilice debe tener como medio de transmisión “9 – Eventos de Workflow”.
  - La configuración de mensajes se hace a través de la transacción **NACE**.

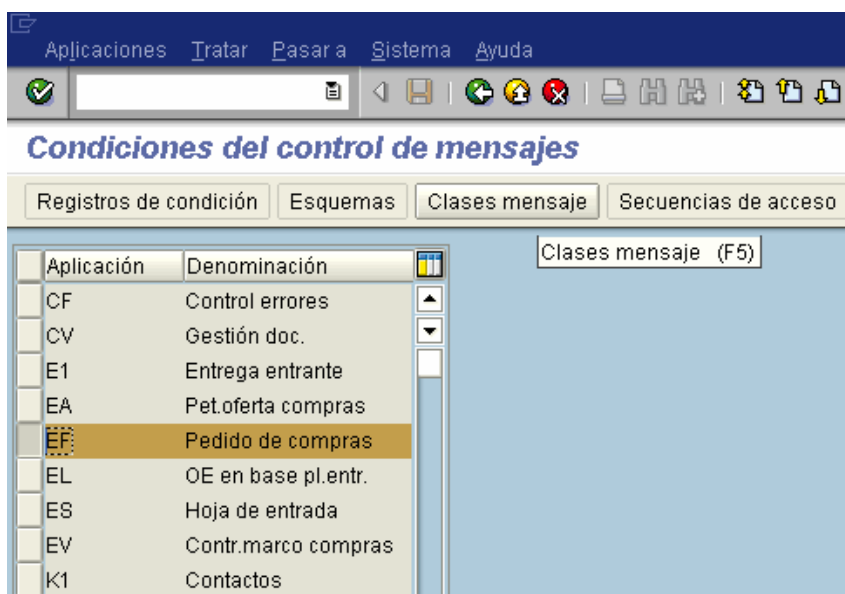
# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Control de Mensajes**
  - Para configurar el mensaje primero entramos en la transacción NACE



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Control de Mensajes**
  - Luego seleccionamos el objeto de negocio con el que deseamos trabajar y presionamos el botón “**Clases de Mensaje**”
  - **NOTA:** previamente deben estar configuradas las secuencias de acceso y los esquemas de mensaje (consultar la documentación de SAP)



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Control de Mensajes**
  - Creamos una nueva clase presionando el botón “Entradas Nuevas”
  - Creamos el mensaje con un nombre “EVEN”
  - Colocamos sus atributos

The image shows two overlapping SAP screenshots. The left screenshot, titled "Modificar vista Clases de mensajes: Resumen", displays a table of message classes. The right screenshot, titled "Modificar vista Clases de mensajes: Detalle", shows the configuration for a new message class named "EVEN".








| Cl.mensa... | Denominación         |
|-------------|----------------------|
| ACE1        | Declaración import.  |
| AUFB        | Reclam.confir.pedido |
| ET01        | Etiqueta de material |

The right screenshot shows the "Clase de mensajes" field set to "EVEN" and the "Evento" field set to "Importación". Below, there are several checkboxes for attributes, including "Suceso con error" (checked), "No modificable", "Evento multilínea" (checked), "Mensaje dependiente de material", and "Mensaje de log de transacción".

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: **Control de Mensajes**
  - Colocamos los textos y los interlocutores
  - Finalmente ingresamos la rutina de proceso
    - Rutina RVNSWE01
    - Form CREATE\_EVENT

**Modificar vista Rutinas de proceso: Detalle**

Entradas nuevas        Formulario

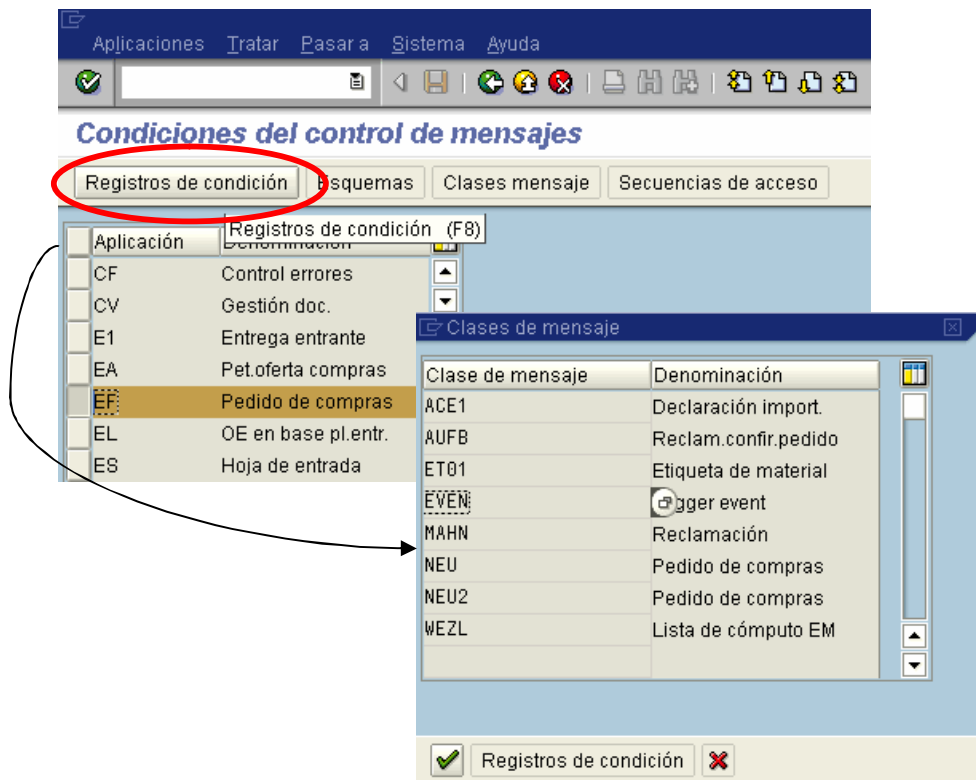
Estructura de diálogo

- Clases de mensajes
  - Título y textos de mai
  - Rutinas de proceso**
  - Funciones de interlo

|                |              |                                 |
|----------------|--------------|---------------------------------|
| Clase mensaje  | EVEN         | Trigger event                   |
| Aplicación     | EF           | Pedido de compras               |
| Medio de envío | 9            | Eventos (SAP Business Workflow) |
| Módulo dispos. |              |                                 |
| SmartForm      |              |                                 |
| Proceso 1      |              |                                 |
| Programa       | RVNSWE01     |                                 |
| Rutina Form    | CREATE_EVENT |                                 |
| Formulario     |              |                                 |

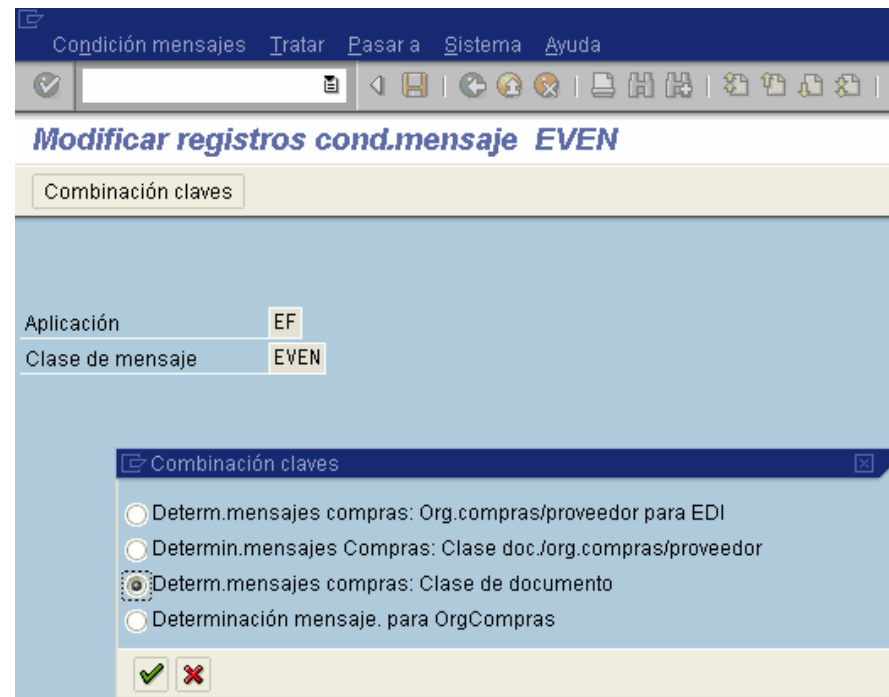
# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Control de Mensajes
  - Una vez definido la clase de mensaje configuramos los registros de condición seleccionando la clase de mensaje creada (EVEN)



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Control de Mensajes
  - Seleccionamos la combinación de claves deseada





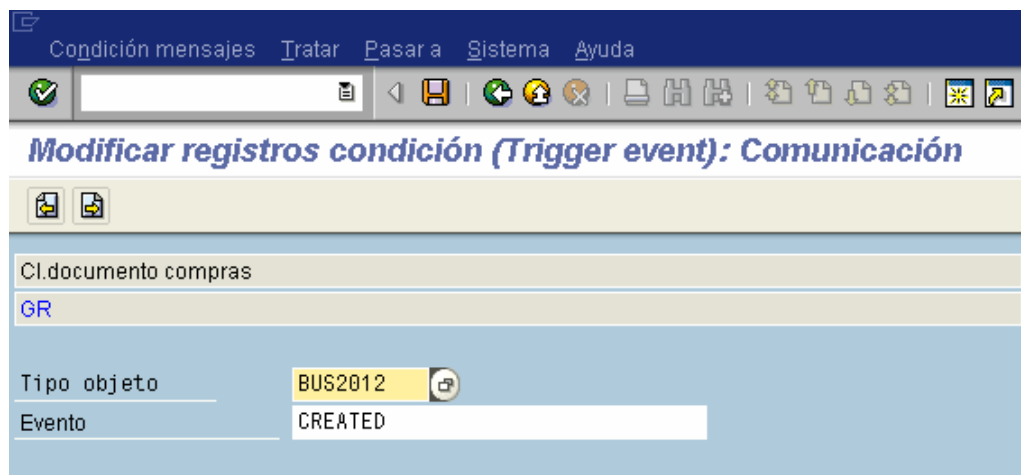
# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Control de Mensajes
  - Creamos los registros con:
    - Clave de Selección (en el ejemplo es el tipo de documento)
    - Función de interlocutor
    - Tipo de mensaje (9 – Eventos Workflow)
    - Tipo de salida (4 – Inmediatamente al grabar el documento)

| Cl...                               | Denominación | Funci | Interl. | M... | Ev... | Idio... |
|-------------------------------------|--------------|-------|---------|------|-------|---------|
| GR                                  |              | PR    |         | 9    | 4     | ES      |
| NB                                  |              | PR    |         | 9    | 4     | ES      |
| NI                                  |              | PR    |         | 9    | 4     | ES      |
| <input checked="" type="checkbox"/> |              |       |         |      |       |         |

# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Control de Mensajes
  - Presionamos el botón “Comunicación”
  - Aquí indicamos el **tipo de objeto** y el **evento** que deseamos que el mensaje lance.
  - En el ejemplo vemos que lanza el evento “**Created**” del objeto “**Pedido**”



# Lanzando Eventos desde Aplicaciones SAP

- Lanzando Eventos: Otros formas de lanzar eventos
  - Además de las formas explicadas (las mas importantes) podremos definir otras maneras de lanzar eventos no tan habituales como ser:
    - Utilizando el LIS (sistema de información logística) podremos definir reportes que lancen excepciones. Al lanzar la excepción podremos configurar el sistema para que adicionalmente llame un evento (transacción **AWUW**).
    - A través de **cambios en los datos maestros del módulo HR**. Para este caso debemos relacionar un infotipo de HR al correspondiente tipo de objeto y activar el evento.
    - Utilizando **Business Transaction Events**. Los business transaction events se utilizan para lanzar eventos relacionados a las aplicaciones financieras de SAP como contabilidad de mayores, cuentas a pagar y a cobrar, etc.
    - Utilizando las **APIs de Workflow** en programas ABAP. Este punto lo veremos en detalle mas adelante durante la presentación.

# Contenidos

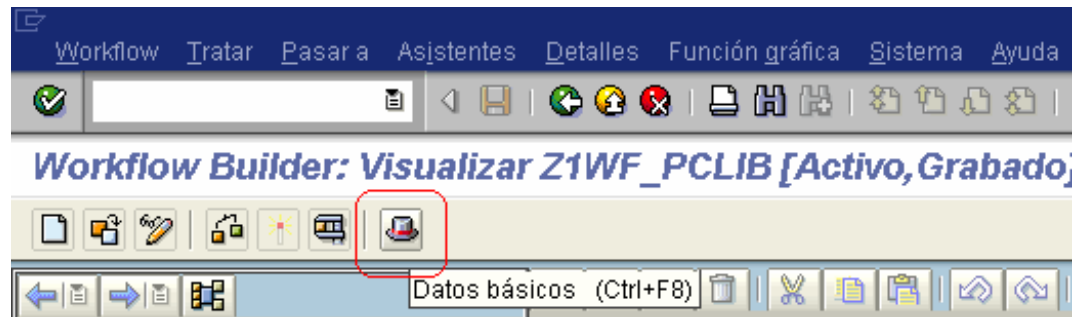
- Definición de Eventos
- Lanzando eventos desde las aplicaciones SAP
  - Lanzando eventos con “change documents”
  - Lanzando eventos con mensajes
  - Lanzando eventos con cambios de status
  - Otros tipos de lanzamiento de eventos
- Relacionar un evento a un workflow
  - Definición en el Workflow Builder
  - Event Linkage
  - Condiciones de Inicio
- Desarrollos en Eventos
  - Llamar un evento de un programa
  - Desarrollo de módulos de funciones para eventos

## Relacionar un Evento a un Workflow


- ¿Quiénes reaccionan a los eventos?
  - Un **Workflow**
  - Un **paso** de tipo “**esperar por evento**”
  - Una **Tarea**
- Cuando se lanza un evento, este puede tener **uno o mas receptores**.
- Se pueden usar eventos para:
  - **Lanzar** un workflow o una tarea
  - **Parar** un workflow, una tarea o un paso de tipo “esperar por evento”.
  - **Forzar al workflow a cambiar algo** sobre si mismo.
- A su vez un workflow puede en si mismo lanzar eventos utilizando el paso de tipo “**creador de evento**”.

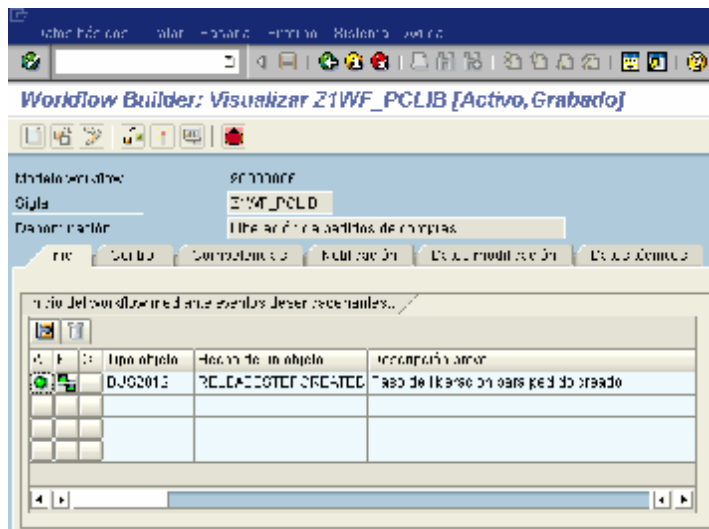
# Relacionar un Evento a un Workflow

- Unir el Evento al Workflow
  - Para establecer el inicio automático de un workflow a partir de un evento debemos indicárselo en la configuración del workflow en el Workflow Builder (SWDD).
  - Una vez posicionados el en workflow que deseamos iniciar con un evento, debemos pasar a la cabecera del workflow.

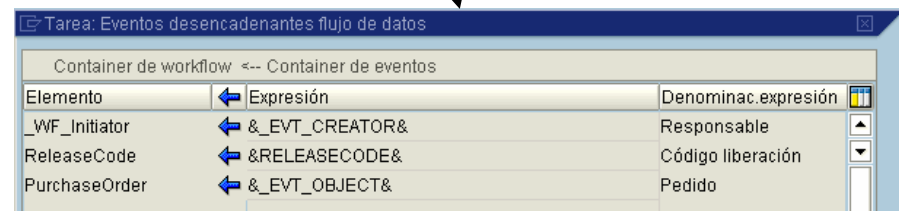


# Relacionar un Evento a un Workflow

- Unir el Evento al Workflow
  - En la cabecera indicaremos que tipo de objeto y evento lanzaran el workflow
  - Al crear la relación automáticamente aparecerá un binding (  ) que pasará datos desde el contenedor del evento al del workflow.
  - Podremos modificar el binding para agregar los parámetros que deseemos.



Los parametros `_WF_Initiator` y `_EVT_Object` se generan automáticamente

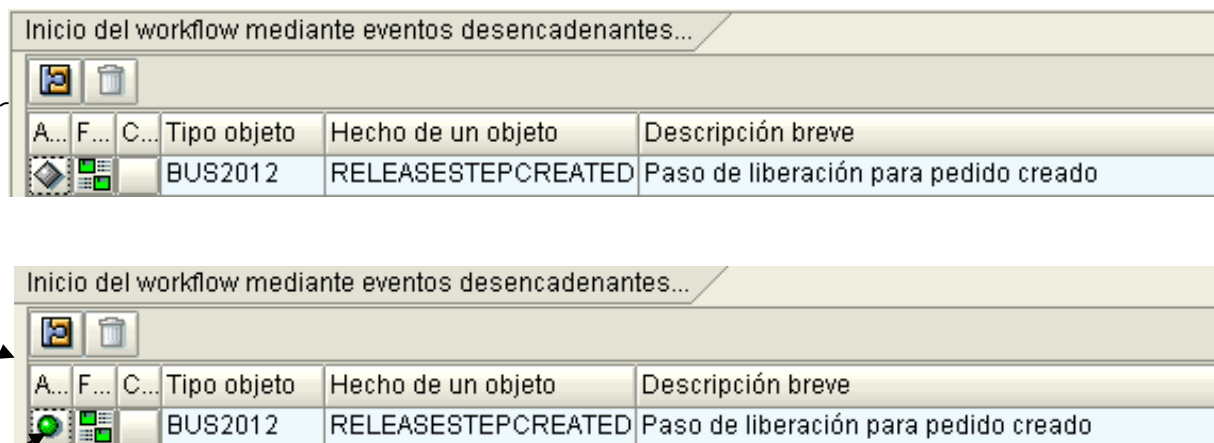


The screenshot shows the 'Tarea: Eventos desencadenantes flujo de datos' dialog box. It contains a table with columns for 'Elemento', 'Expresión', and 'Denominac. expresión'. The table lists three parameters: '\_WF\_Initiator' with expression '&\_EVT\_CREATOR&' and name 'Responsable', 'ReleaseCode' with expression '&RELEASECODE&' and name 'Código liberación', and 'PurchaseOrder' with expression '&\_EVT\_OBJECT&' and name 'Pedido'.

| Elemento      | Expresión      | Denominac. expresión |
|---------------|----------------|----------------------|
| _WF_Initiator | &_EVT_CREATOR& | Responsable          |
| ReleaseCode   | &RELEASECODE&  | Código liberación    |
| PurchaseOrder | &_EVT_OBJECT&  | Pedido               |

# Relacionar un Evento a un Workflow

- Unir el Evento al Workflow
  - Finalmente deberemos “activar” el binding entre el workflow y el evento. Esta activación en la jerga de workflow se denomina “event linkage”
  - Tener en cuenta que esta actividad genera una orden de transporte de customizing.



Presionar aquí...



# Relacionar un Evento a un Workflow

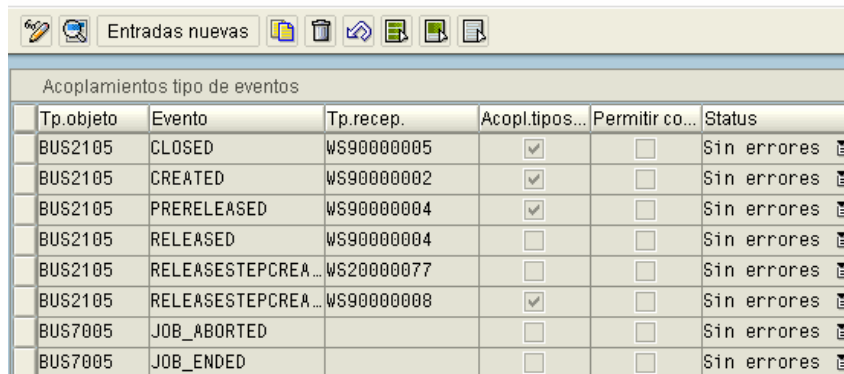
- Event Linkage
  - Otra forma de activar el linkage entre el evento y el workflow es a través del siguiente menú



# Relacionar un Evento a un Workflow

- Event Linkage
  - En esta transacción nos aparecerá una lista con todas las posibles relaciones entre un workflow y un evento, a su vez podremos ver si el linkage que los vincula esta o no activado.

## Modificar vista "Acoplamiento tipo de eventos": Resumen



| Entradas nuevas | Acoplamiento tipo de eventos |            |                                     |                          |             |
|-----------------|------------------------------|------------|-------------------------------------|--------------------------|-------------|
| Tp.objeto       | Evento                       | Tp.recep.  | Acopl.tipos...                      | Permitir co...           | Status      |
| BUS2105         | CLOSED                       | WS90000005 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Sin errores |
| BUS2105         | CREATED                      | WS90000002 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Sin errores |
| BUS2105         | PRERELEASED                  | WS90000004 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Sin errores |
| BUS2105         | RELEASED                     | WS90000004 | <input type="checkbox"/>            | <input type="checkbox"/> | Sin errores |
| BUS2105         | RELEASESTEPCREA...           | WS20000077 | <input type="checkbox"/>            | <input type="checkbox"/> | Sin errores |
| BUS2105         | RELEASESTEPCREA...           | WS90000008 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | Sin errores |
| BUS7005         | JOB_ABORTED                  |            | <input type="checkbox"/>            | <input type="checkbox"/> | Sin errores |
| BUS7005         | JOB_ENDED                    |            | <input type="checkbox"/>            | <input type="checkbox"/> | Sin errores |

- Haciendo doble-click sobre un linkage pasaremos a ver sus detalles

# Relacionar un Evento a un Workflow

- Event Linkage

- En esta pantalla principalmente podremos activar o desactivar el linkage, presionando sobre la opción “Acopl. Tipos activado”.
- Luego podremos configurar opciones acerca de cómo el sistema reaccionará ante un error, pudiendo optar entre:
  - Desactivar el evento
  - Marcar el acoplamiento (linkage) como erróneo
  - No modificar el acoplamiento

*Modificar vista Acoplamientos tipo de eventos: Detalle*

Entradas nuevas

Tipo objeto: BUS2105  
Evento: CREATED  
Tp.receptor: WS90000002

Acoplamientos tipo de eventos

|  |                         |
|--|-------------------------|
| Progr.rec.MF   | SWW_WI_CREATE_VIA_EVENT |
| Función verif.   |                         |
| Tp.progr.rec.MF  |                         |
| Destino  |                         |
| <input checked="" type="checkbox"/> Acopl.tipos activado |                         |
| <input type="checkbox"/> Permitir cola de eventos        |                         |
| Comportam.al dar mensaje error                           | Param.previa sistema    |
| Status pr.rec.eventos                                    | Sin errores             |

# Relacionar un Evento a un Workflow

- Event Linkage

- También deberemos indicarle cual es el programa receptor del evento. Por defecto para los workflows el programa es:
  - **SWW\_WI\_CREATE\_VIA\_EVENT**
- Podremos indicar una función de verificación la cual se utiliza para validar el linkage (si falla la función no se inicia el workflow)
- Por último podremos facilitar una función para determinación de receptores, es decir que en tiempo de ejecución (observando parámetros del contenedor del evento) podremos determinar el workflow receptor dinámicamente.

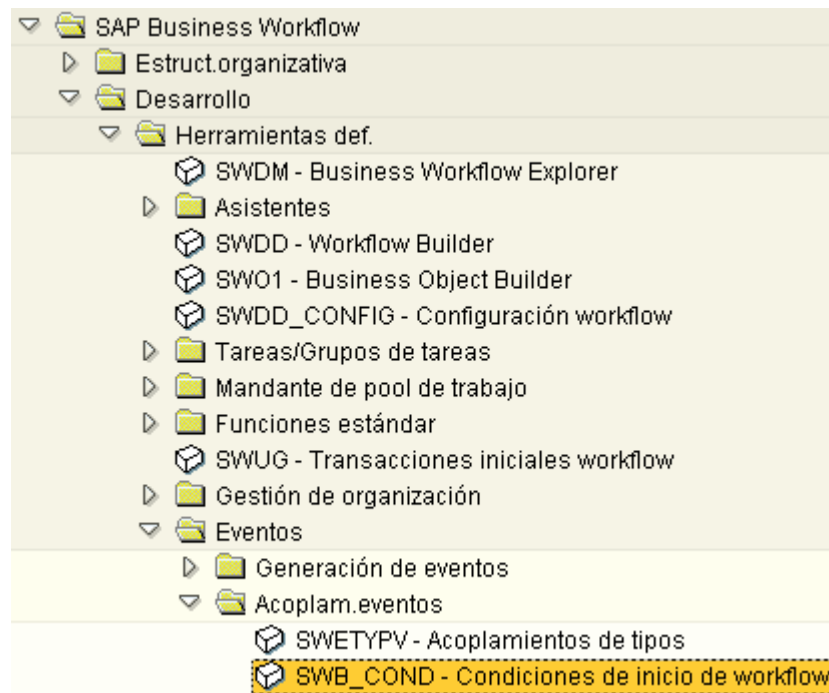
| Acoplamiento tipo de eventos |                         |
|------------------------------|-------------------------|
| Progr.rec.MF                 | SWW_WI_CREATE_VIA_EVENT |
| Función verif.               |                         |
| Tp.progr.rec.MF              |                         |

# Relacionar un Evento a un Workflow

- Condiciones de Inicio
  - Anteriormente vimos como puede condicionarse el inicio de un workflow utilizando una función de verificación en el linkage (luego veremos como programarla),
  - Sin embargo esto puede ser un poco tedioso, sobre todo por que hay que desarrollar ABAP para lograrlo (aunque es la manera mas flexible de lograrlo).
  - SAP provee una manera mucho mas fácil (no tan flexible pero bastante completa) de limitar el inicio de un workflow al dispararse un evento y esto es a través de las condiciones de inicio.

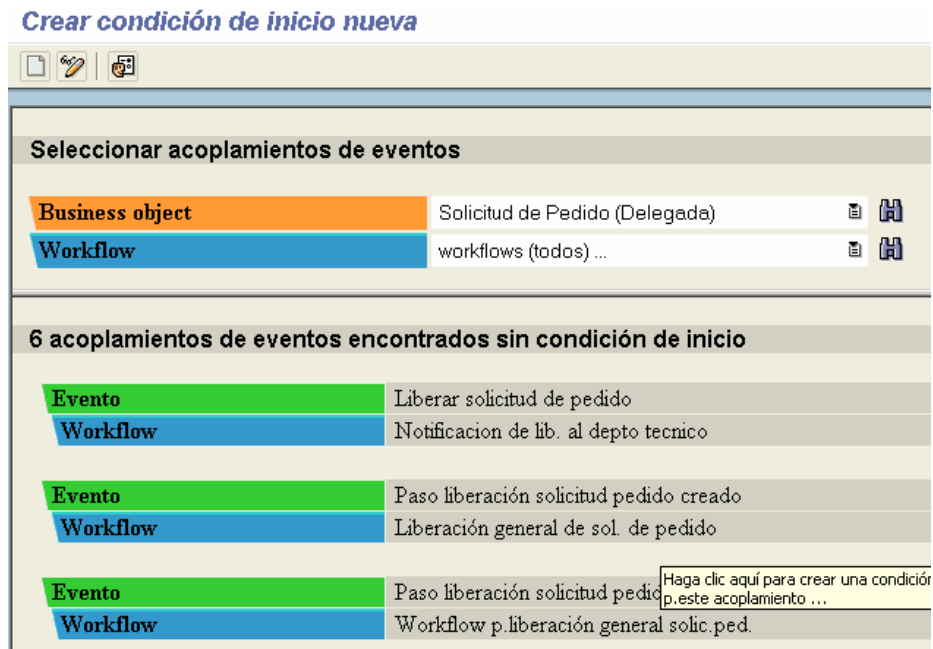
# Relacionar un Evento a un Workflow

- Condiciones de Inicio
  - Para configurar condiciones de inicio seguir la siguiente opción de menú



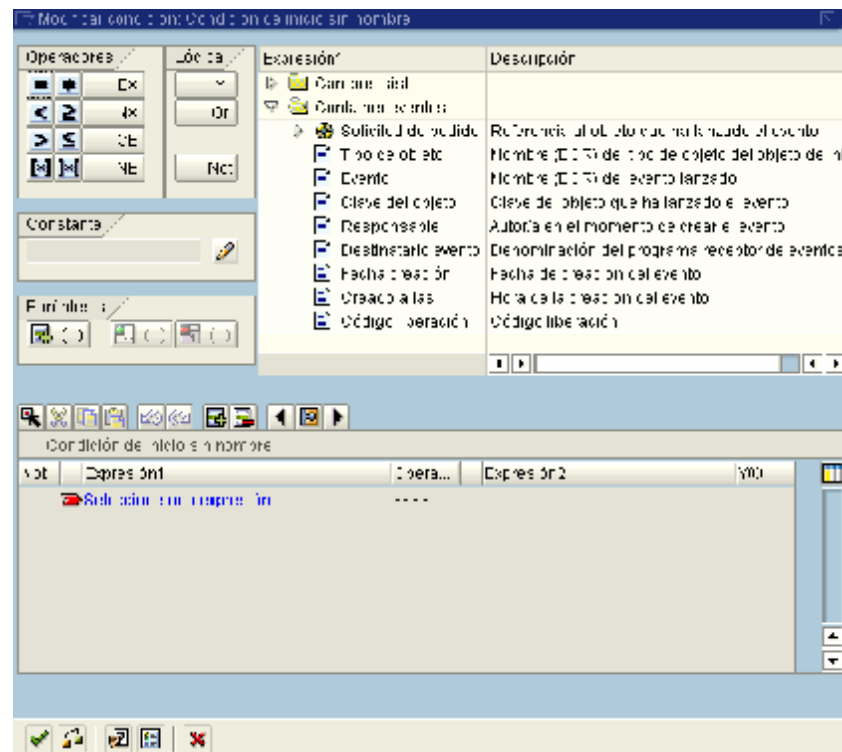
# Relacionar un Evento a un Workflow

- Condiciones de Inicio
  - Para crear la condición seleccionamos el tipo de objeto (en el ejemplo es la solicitud de pedido)
  - Aparecerán todos los eventos acoplados con workflow y seleccionamos uno.



# Relacionar un Evento a un Workflow

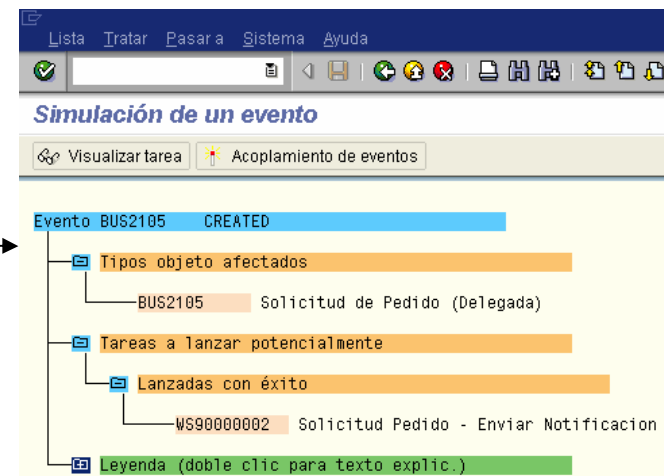
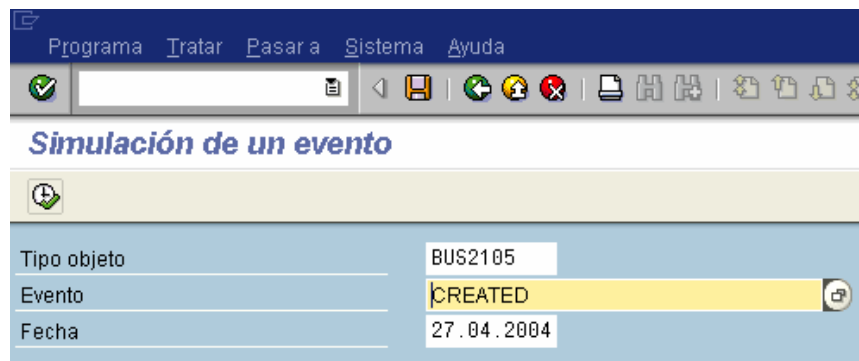
- Condiciones de Inicio
  - Utilizando las variables del contenedor del evento, creamos las condiciones lógicas que deseemos para que se cumpla o no el lanzamiento del workflow.





# Relacionar un Evento a un Workflow

- Finalizado la configuración del linkage de eventos y workflow, y la correspondiente condición de inicio, podremos verificar el evento.
- Para verificar los eventos podremos usar la transacción para simular eventos (**SWU0**) o para crearlos (**SWUE**).

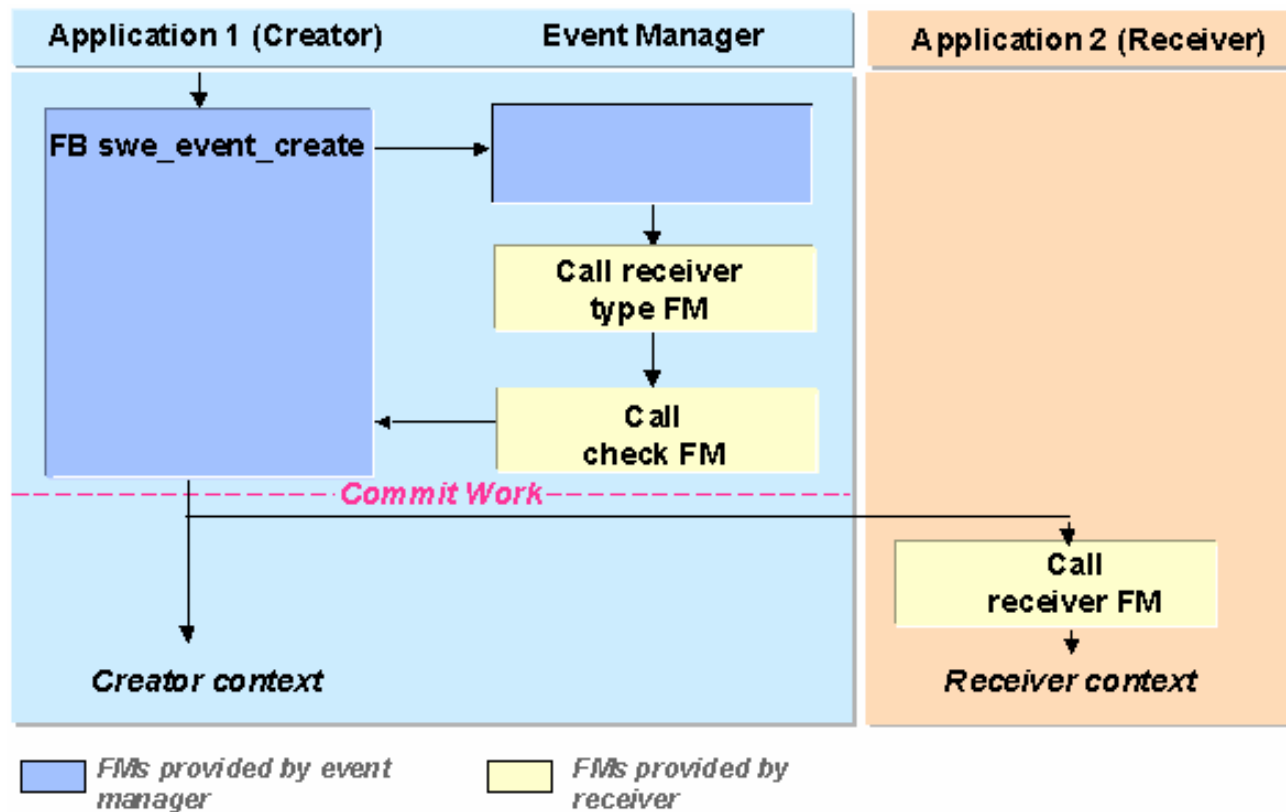


# Contenidos

- Definición de Eventos
- Lanzando eventos desde las aplicaciones SAP
  - Lanzando eventos con “change documents”
  - Lanzando eventos con mensajes
  - Lanzando eventos con cambios de status
  - Otros tipos de lanzamiento de eventos
- Relacionar un evento a un workflow
  - Definición en el Workflow Builder
  - Event Linkage
  - Condiciones de Inicio
- Desarrollos en Eventos
  - Llamar un evento de un programa
  - Desarrollo de módulos de funciones para eventos

# Desarrollos en Eventos

- Arquitectura del Sistema Eventos en SAP



# Desarrollos en Eventos

- Desarrollo de un programa que lance un evento
  - El programa que desee disparar un evento deberá utilizar el modulo de funciones `SWE_EVENT_CREATE` que forma parte de la amplia API de funciones para workflow.
  - La estructura lógica del programa debería ser la siguiente:
    - Llenar el **contenedor de eventos** con los parámetros necesarios
    - Componer la **clave del objeto** que debe instanciarse para llamar al evento
    - Llamar la función **SWE\_EVENT\_CREATE**
    - Controlar las excepciones
    - Disparar el evento con **COMMIT\_WORK** explícito

```
include <CNTN01>.

data: begin of asset_key,
      company_code like anla-bukrs,
      asset_no     like anla-anln1,
      sub_number   like anla-anln2,
end of asset_key.
data: object_key   like sweinstcou-objkey.
      swc_container like evt_container.

* Write parameters into event container
swc_set_element evt_container 'flag_equi_aendern' 'X'.

* Compose object key
asset_key-company_code = '0001'.
asset_key-asset_no     = '000000123456'.
asset_key-sub_number   = '0100'.
object_key = asset_key.

* Trigger the event
call function 'SWE_EVENT_CREATE'
  exporting
    objtype = 'BUS1022'
    objkey  = object_key
    event   = 'changed'
  tables
    event_container = evt_container
  exceptions
    others = 01.

if sy-subrc ne 0.
  " do your own error handling
endif.

* start tRFC processing
commit work.
```

# Desarrollos en Eventos

- Tener en cuenta los objetos delegados al ejecutar el evento

| Object type | Event   | Receiver type |
|-------------|---------|---------------|
| BUS1001     | Created | WS00005711    |
| Z_BUS1001   | Created | WS97000313    |

```

include <CONTN01>.
data: object key      like sweinstcou-objkey.
      swc_container   like evt_container.

object_key = '000012345678'.

* trigger the event
call function 'SWC EVENT CREATE'
  exporting
    objtype = 'BUS1001'
    objkey  = object_key
    event   = 'Created'
  tables
    event_container = evt_container
    exceptions      = 01.

* start IREFC processing
commit work.
  
```

| Object type | Event   | Receiver type |
|-------------|---------|---------------|
| BUS1001     | Created | WS00005711    |
| Z_BUS1001   | Created | WS97000313    |

```

include <CONTN01>.
data: object key      like sweinstcou-objkey.
      swc_container   like evt_container.

object_key = '000012345678'.

* trigger the event
call function 'SWC EVENT CREATE'
  exporting
    objtype = 'Z_BUS1001'
    objkey  = object_key
    event   = 'Created'
  tables
    event_container = evt_container
    exceptions      = 01.

* start IREFC processing
commit work.
  
```

# Desarrollos en Eventos

- Desarrollo del modulo de **función para verificar el evento**

- El modulo de función debe tener la siguiente **estructura**

- La lógica del modulo de funciones debe ser la siguiente

- Leer los parámetros del contendor de eventos
- Verificar lo que se desee verificar
- En caso de errores **lanzar cualquier excepción**

```
function check_fm
  importing objtype like swetypecou-objtype
             objkey  like sweinstcou-objkey
             event   like swetypecou-event
             rectype like swetypecou-rectype
  tables     event_container structure swcont
  exceptions others
```

```
function check object type idocappl.
*****
* importing objtype like swetypecou-objtype
*           objkey  like sweinstcou-objkey
*           event   like swetypecou-event
*           rectype like swetypecou-rectype
* tables     event_container structure swcont
* exceptions check_failed
*****

include <CNTN01>.

constants:
  c_otype_appl like swetypecou-objtype value 'IDOCAPPL',
  c_evt_otype  like swcont-element value '_EVT_OBJTYPE'.

data: initiating_otype like swetypecou-objtype.

* Implementation of check_fm

* Read originating objtype from event container
swc_get_element event_container
               c_evt_otype initiating_otype .

* check if initiating objtype is desired one
if initiating_otype <> c_otype_appl.
  raise check_failed.
endif.

endfunction.
```

# Desarrollos en Eventos

- Desarrollo del modulo de **función para determinar receptores**

- El modulo de función debe tener la siguiente **estructura**

- La lógica del modulo de funciones debe ser la siguiente

- Leer los parámetros del contendor de eventos
- Determinar el workflow que debe llamarse utilizando los datos obtenidos
- En caso de errores **lanzar cualquier excepción, sino terminar la función.**

```
function rectypeget_fm
    importing objtype like swetypecou-objtype
            objkey    like sweinstcou-objkey
            event like swetypecou-event
            generic_rectype
                    like swetypecou-rectype
    exporting rectype like swetypecou-rectype
    tables      event_container structure swcont
    exceptions others
```

```
function cwf event rec dokst get.
*****
* importing objtype           like swetypecou-objtype
* importing objkey           like sweinstcou-objkey
* importing event            like swetypecou-event
* importing generic_rectype  like swetypecou-rectype
* exporting rectype          like swetypecou-rectype
* tables      event_container structure swcont
* exceptions no_rectype
*****
include <CNTN01>.

data: l_dokst like draw-dokst.
data: begin of drawkey,
      dokar like draw-dokar,
      doknr like draw-doknr,
      dokvr like draw-dokvr,
      doktl like draw-doktl,
      end of drawkey.

* Implementation of receiver type fm
* Read document status from event container
swc_get_element event_container
'DOCUMENTSTATUS' l_dokst.
drawkey = objkey.

* read receiver type from table tdws
select single * from tdws
      where dokar = drawkey-dokar and
            dokst = l_dokst.
If sy-subscr ne 0.
  Raise no_rectype.
Else.
  Rectype = tdws-otype.
  Rectype+2 = tdws-objid.
Endif.
```