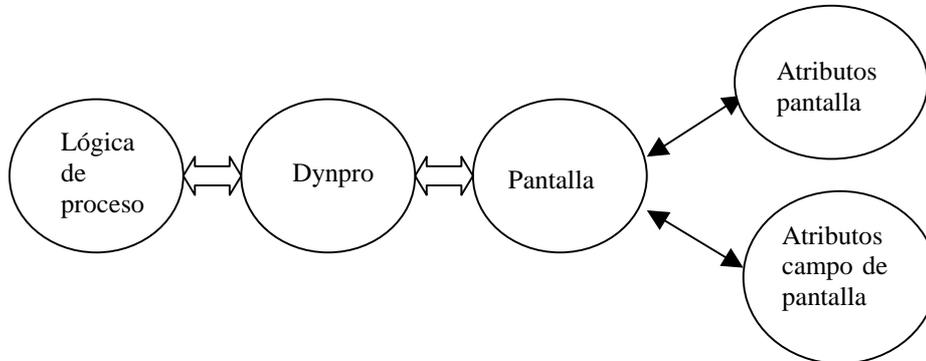


Manual Básico de Modul Pool

MODUL-POOL	2
SCREEN-PAINTER	3
ATRIBUTOS DE UNA DYNPRO	4
LISTA ELEMENTOS DE UNA DYNPRO	5
LÓGICA DE PROCESO DE UNA DYNPRO	6
PBO.....	7
PAI.....	8
EDITOR DISPOSICIÓN DE UNA DYNPRO	8
ÁREA DE LOS CONTROLES	9
ÁREA DE MENÚ	10
Modo TEST	11
ATRIBUTOS	12
CAMPOS DICTIONARY/PROGRAMA.....	15
VENTANA LISTA ELEMENTOS	17
CONTROLES	18
Ejemplo 1	18
Ejemplo 2:.....	21
Ejemplo 3:.....	27
Radiobutton:.....	41
MENU PAINTER.....	43
BARRA DE MENÚS	48
TECLAS DE FUNCIÓN	50
TECLAS DE FUNCIÓN ESTANDAR	50
COMO CREAR UNA TECLA DE FUNCIÓN	51
MANTENIMIENTO DE UNA TECLA DE FUNCIÓN	53
BARRA DE PULSADORES	53

MODUL-POOL

Un *modulpool* nos permite mantener los dynpros de una transacción (un *modulpool* solo se puede ejecutar a través de una transacción). Un *dynpro* está compuesto, en primer lugar, de una pantalla y una lógica de proceso de esa pantalla. En la pantalla definimos los atributos de la propia pantalla, así como los atributos de cada campo. En la Lógica de proceso definimos los módulos que se ejecutarán antes de que el sistema presente la pantalla, y los procesos que se ejecutarán después. Los objetos parciales de un *dynpro* serían:



Un ejemplo de *modulpool* sería:

Consulta Circuitos Workflow de Autorizaciones

Consulta Iniciar Administradores Sistema Ayuda

Ejecutar Segu Firmante Descripción Impr. Relación Ver Documento

Selección: T Circuito: 20 Fecha de Inicio: 15.05.2000 a 15.06.2000

Estado: Fecha de Intervención:

Sociedad: 0430 Importe del documento:

Nodo/CeCo: Número de documento:

Soc.	Ci.	Workflow	Documento	S	Importe	F.Inicio	F.Interv.
0430	20	716104	1400002173	C	10.000	01.06.2000	
0430	20	716113	1500000580	C	10.000	01.06.2000	
0430	20	717314	1400002181	C	0	09.06.2000	
0430	20	717315	1400002181	C	0	09.06.2000	
0430	20	717316	1400002181	C	0	09.06.2000	

Han sido encontrados 5 registros

GED (2) (020) emadsapi

Screen Painter

Menú Painter

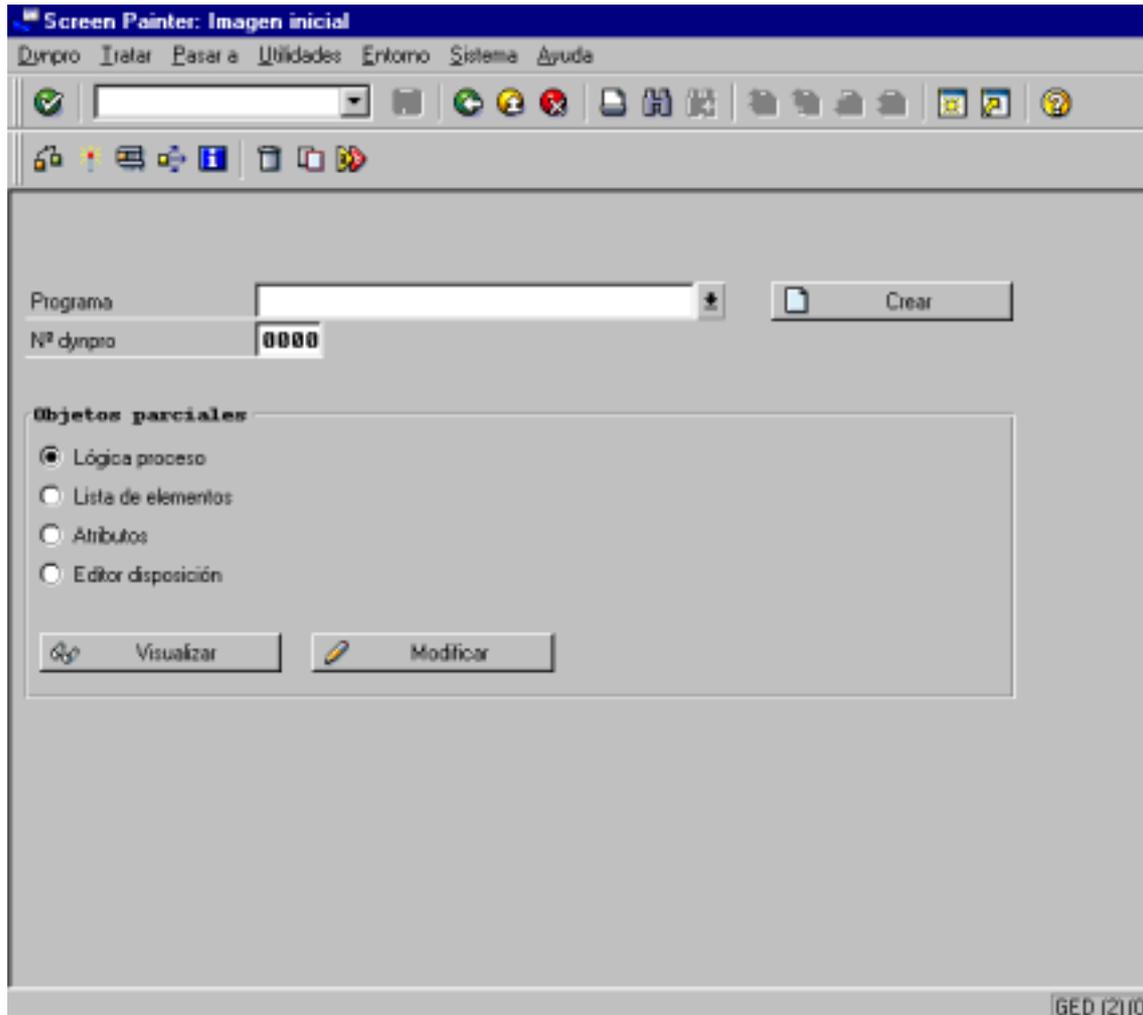
Un *modulpool* esta formado por una *Screen Painter* y un *Menu painter*.

Para la creación de un *modulpool* tendríamos que seguir los siguientes pasos:

1. Crear un programa de tipo *Modulpool*, que sería el programa marco.
2. Ir a la *Screen Painter* para crear el dynpro.
3. Definir el *Menú painter*.
4. Crear la transacción para que llame al *modulpool*.

SCREEN-PAINTER

Para crear las *dynpros* y asociarlas al programa lo haremos a través del *Screen painter* (transacción SE51) desde el *SAP Easy Acces* sería: *Menú SAP* → *Herramientas* → *Workbench ABAP* → *Interfase del usuario* → *SE51* → *Screen Painter*. La pantalla que nos tiene que aparecer es la siguiente:



El significado de los campos es el siguiente:

- Programa → En este campo introduciremos el programa que hayamos creado con anterioridad. El programa debe existir obligatoriamente.
- Dynpro → Numero de dynpro o pantalla que vamos a modificar, visualizar o crear.
- En los objetos parciales tenemos lo siguiente:
 1. Lógica de proceso → Es la parte donde se codifica las sentencias para el control de la dynpro. La lógica se separa en PBO (Process Before Output, proceso que se ejecuta antes de mostrar la dynpro) y en PAI (Process After Input, proceso que se ejecuta después de la selección de algún objeto en pantalla o cuando se pulse determinadas teclas).
 2. Lista Elementos → Nos muestra una lista de los campos y sus atributos que se utilizan en la dynpro.
 3. Atributos → Nos muestra los atributos de la dynpro.

4. Editor Disposición → Es donde se crearía la pantalla “in situ”. Hay dos opciones para la creación en modo texto o en modo gráfico.

Para crear una dynpro tenemos que poner el nombre y dynpro. Es recomendable que la numeración de la dynpro sea 9xxx y que haya una separación de 10 entre dynpro y el nombre del programa que sea *SAPMZxw*, donde: *x* -> es el módulo y *w* es una pequeña descripción del programa.

Cuando pulsamos el botón de *Modificar*, *Visualizar* o *Crear* nos saldrá una pantalla con 3 pestañas que son:

- Atributos → Son los atributos de la dynpro. Pestaña que sale por defecto cuando creamos una dynpro.
- Lista. Elem → Muestra los campos y atributos que se utilizan en la dynpro, y donde podremos modificar ciertos valores de los campos. Se suele utilizar para consultar los valores de los campos, para modificar sus atributos se utiliza la pantalla de Dispositivos.
- Lóg. Proceso → Es donde escribiremos los códigos necesarios para el control de la dynpro.

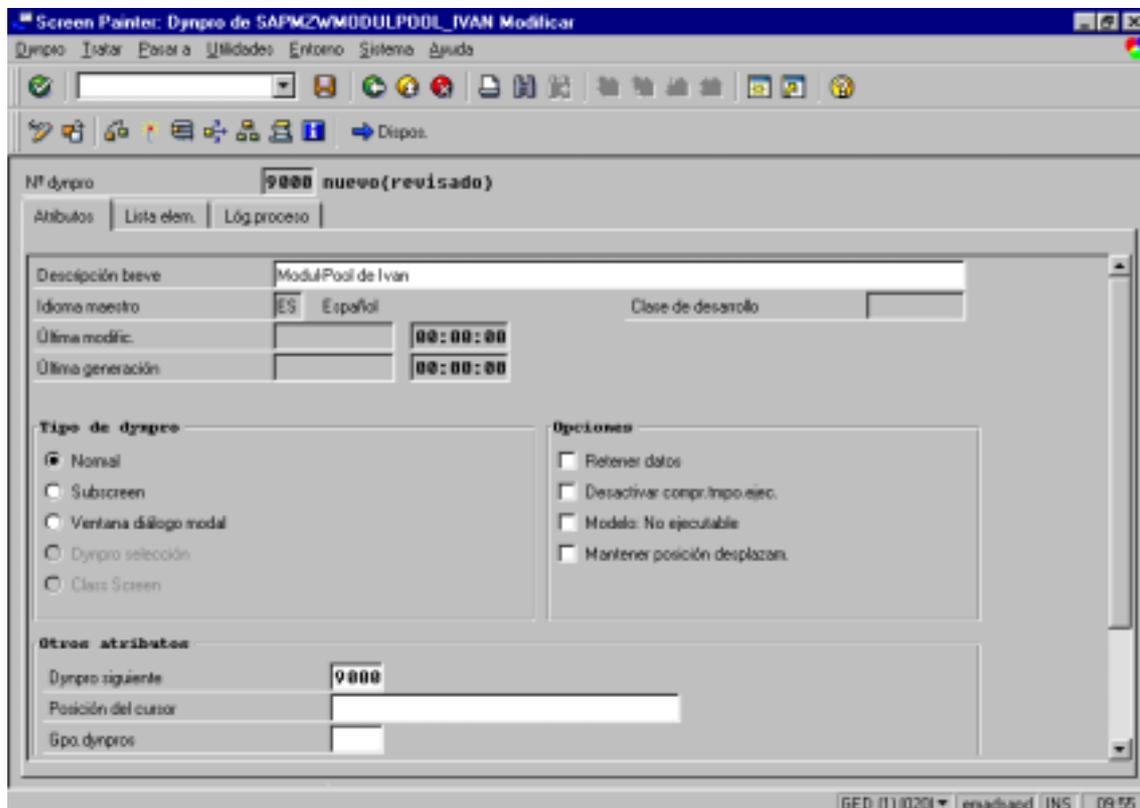
Aparte hay un botón en la parte superior de la pantalla que es el siguiente:



Este botón a que SAP lo llama: *Pasar a dispositivo* en realidad es el editor gráfico es la parte donde se ponen los campos en la dynpro y configuramos los atributos de los que salgan en pantalla (es más cómodo hacerlo desde aquí).

ATRIBUTOS DE UNA DYNPRO

En la pantalla de atributos podemos definir que aspecto tenga nuestra dynpro. La pantalla de atributos es la siguiente:



El campo *Descripción breve* es obligatorio.

Manual Básico de Modul Pool

Los tipos de dynpro son los siguientes:

- Normal → Es una dynpro normal y es la opción que SAP pone por defecto.
- Subscreen → Es una dynpro que esta dentro de una dynpro.
- Ventana diálogo modal → Es una ventana de diálogo.

Las opciones son las siguientes:

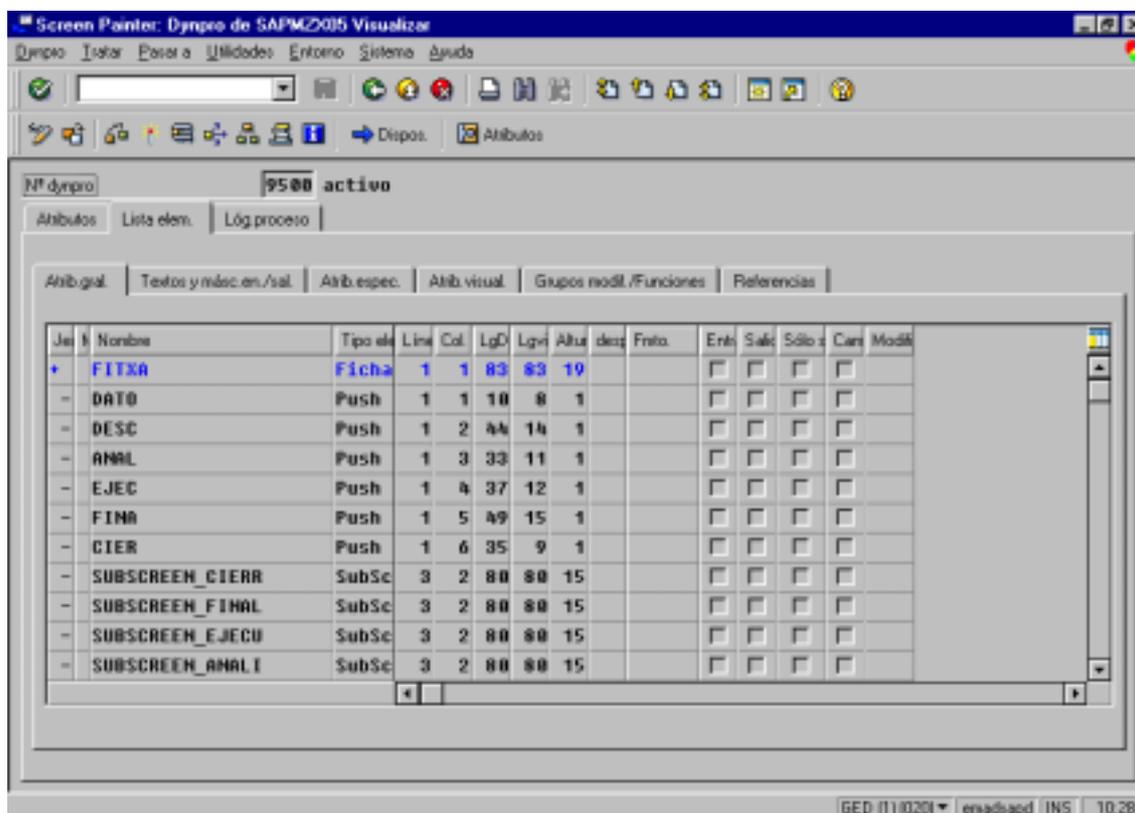
- Retener datos → Activando este checkbox conseguimos que cuando accedamos de nuevo a la dynpro conservaremos los datos que hayamos introducido. Inclusive si accedemos desde otro modo.
- Desactivar compr. Tmpo. Ejec. → Si no se activa comprime la dynpro en tiempo de ejecución. La compresión es eliminar líneas invisibles o marcos que estén al final de la pantalla, etc...
- Modelo: No ejecutable → Uso interno de SAP.
- Mantener posición desplazam. → Se utiliza para dynpros largos y guarda la posición donde estábamos al realizar alguna acción.

En otros atributos tenemos:

- Dynpro siguiente → Se utiliza para cuando se pulse la tecla ENTER que vaya a la dynpro indicada. Si ponemos el valor '0' saldrá del programa inmediatamente.
- Posición del cursor → Aquí pondremos el nombre del campo donde queremos que aparezca el cursor cuando aparezca la pantalla.
- Gpo. Dynpros → Este permite asignar la dynpro que estamos tratando a un grupo de dynpros (que se actualiza en la tabla TFWAT). Con ello podemos hacer modificaciones uniformes a un grupo de dynpros.
- Líneas/Columnas -> Indica cuantas líneas y columnas están ocupadas y cual es el tamaño actual de la dynpro. Estos últimos valores se pueden modificar.
- Menú contxt. -> Indicamos que menú contextual podemos asignar a una dynpro.

LISTA ELEMENTOS DE UNA DYNPRO

Esta es la pantalla con más opción que hay en la *Screen painter*. Un ejemplo sería la siguiente pantalla:



En esta pantalla tenemos una pestaña con opciones varias sobre los campos de la dynpro. En cada pestaña hay un *Table control* que nos permite modificar aspectos varios de los campos.

Para no profundizar demasiado haré una explicación básica de las pestañas principales. Las opciones de cada pestaña se pueden consultar en la ayuda o pulsando F1 o sobre el atributo de un campo.

- Atrib. Gral. → En esta pestaña podremos ver y modificar el aspecto general de los campos como en que posición esta, su longitud, si es un campo de entrada o solo salida, tipo de campo, etc...
- Textos y másc. en/sal. → Desde aquí podemos ver y modificar el aspecto externo (lo que se ve en pantalla) del campo. Como el texto de salida, el icono que tiene, etc...
- Atrib. Espec. → Desde aquí vemos los atributos especiales del campo. Como puede ser: que sea campo obligatorio o no, el valor se guarde en campo de memoria, permitir la entrada de mayúsculas y minúsculas, etc...
- Atrib. Visual. → En esta pestaña vemos los atributos visuales del campo. Como que sea visible o invisible, permita ceros a la izquierda, etc...
- Grupos modif./Funciones -> Desde aquí vemos a que grupos pertenecen el campo, que grupo de función tiene, etc...
- Referencias -> Desde aquí vemos el campo donde se utiliza, por ejemplo, podemos que campos tiene una *Subscreen*.

LÓGICA DE PROCESO DE UNA DYNPRO

La lógica de proceso es, personalmente, la parte más importante de una dynpro. Ya que es el lugar donde se controlan todo lo que pasa en una dynpro (que campos se visualizan, que valores tendrán los campos, que tecla se ha pulsado, etc...). Un ejemplo de lógica de proceso sería:

```
process before output.
  module status_9100.
  module preparar_fecha.
  loop at t cursor reg1.
    module carga_linea.
  endloop.

process after input.
  module exit_command at exit-command.
  module ok_code_1_9100.

  chain.
    field: zuconx-sociedad,
        zuconx-subcirccto.
    module validar_sociedad.
  endchain.
  chain.
    field zuconx-sociedad module preparar_lectura_sociedad.
  endchain.
  chain.
    field: zuconx-nodo.
  endchain.
```

La lógica de proceso se divide en dos partes que son el PBO y el PAI, que serán explicada más adelante. Estas dos partes se dividen gracias a la instrucción *PROCESS*, la sintaxis de dicha instrucción es la siguiente:

Manual Básico de Modul Pool

PROCESS BEFORE OUTPUT.

El proceso PBO se suele utilizar para iniciar los campos de la pantalla, posicionar el cursor, mostrar o esconder campo o cambiar los atributos de los campos dinámicamente.

PROCESS AFTER INPUT.

El proceso PAI se suele utilizar para validar las entradas realizadas por el usuario, procesar la posición del cursor o procesar las funciones activadas.

PROCESS ON HELP-REQUEST.

Este evento se procesa si el usuario pulsa la tecla F1 estando el cursor sobre algún campo de la pantalla. Si el evento se define el campo en cuestión (sentencia FIELD), la ayuda que aparece será la programa en la transacción.

PROCESS ON HELP-REQUEST.

Este evento se procesa si el usuario pulsa la tecla F4 estando el cursor sobre algún campo de la pantalla. Si el evento se define el campo en cuestión (sentencia FIELD), la ayuda que aparece será la programa en la transacción.

Ahora explicaremos los aspectos generales de la lógica de proceso.

En la lógica de proceso solo se pueden poner una serie de sentencias determinadas y determinadas variantes de dichas sentencias que son:

- LOOP (se utiliza en el PBO como en el PAI).
- FIELD (solo se puede utilizar en el PAI).
- CHAIN (solo se puede utilizar en el PAI).
- MODULE (se utiliza en el PBO como en el PAI).
- CALL SUBSCREEN (se utiliza en el PBO como en el PAI).

Para solucionar el problema de limitación de sentencias se utiliza la sentencia *module* (que es el equivalente a la sentencia *Perform* pero no tiene la posibilidad de pasar parámetros.). La sentencia *module* llama a un módulo donde tendremos las sentencias que nosotros queramos.

También hay ciertos controles donde es obligado escribir ciertas sentencias para su funcionamiento que son:

- STEP-LOOP.
- TABLE-CONTROL
- SUBSCREEN.

PBO

PBO significa *Process Before Output* y donde se ponen las sentencias que queremos que se ejecuten antes de visualizar la pantalla.

Cuando creamos el PBO, SAP pone el siguiente módulo: *MODULE STATUS_9000* (los últimos 4 dígitos corresponden siempre al número de dynpro). Aunque en un principio dicho módulo esta asteriscado.

Este módulo se utiliza para cargar el *status*, o sea, el *Menu painter* y el título de la ventana.

Podemos crear una *Screen painter* sin *Menu painter* pero en la *Screen painter* tenemos que poner un botón que nos permita salir de la dynpro.

Si vamos al *MODULE STATUS_xxxx* veremos que nos sale el siguiente código:

```
module status_9500 output.  
* SET PF-STATUS 'xxxxxxxx'.  
* SET TITLEBAR 'xxx'.  
endmodule.      "
```

La primera sentencia (*SET PF-STATUS*) es la que llama al *Menu painter* y la segunda (*SET TITLEBAR*) es la que llama al título de la dynpro. Como crear un *Menu painter* y un título serán vistos más adelante.

En el PBO es donde se inicializan los controles *Step loop* y *Table control* con los valores que queramos.

PAI

PAI significa *Process After Input* este proceso se ejecuta cuando pulsamos algún botón (tanto del *Menu painter* o del *Screen painter*), pulsemos ENTER o alguna tecla de función (teclas F1, SHIFT+F2, etc...). Como en el PBO, SAP crea el módulo *module user_command_9500* (los últimos 4 dígitos siempre corresponden al número de dynpro) para el control de las teclas o botones pulsado. Este módulo inicialmente esta asteriscado.

Para saber la tecla o botón se ha pulsado tenemos 2 formas de hacerlo:

1. A través de la variable del sistema SY-UCOMM. Esta variable almacena la última tecla o botón que se ha pulsado.
2. Siempre que se crea una dynpro hay un campo que solo se puede ver en la *Lista Elementos* y que es de tipo *OK*. Para poder activar este campo primero tenemos que declarar un variable en el programa marco que sea un *CHAR* de 20 posiciones y después poner el nombre de la variable en dicho campo. Ejemplo:

Je	N	Nombre	Tipo ele	Líne	Col.	LgD	Lgvi	Altur	desp	Fmto.	Entri	Salic	Sólo s	Cam	Modifi
		OK-CODE	OK	0	0	20	20	1	<input type="checkbox"/>	OK				<input type="checkbox"/>	

El nombre de la variable suele ser OK-CODE, aunque también puede ser: OK_CODE, OKCODE, etc...

No hay ningún problema en que se utilicen las dos a la vez, aunque no se suele hacer para no complicar las cosas.

Hay dos instrucciones que sólo se utilizan en el PAI que se utiliza para controlar los campos de entrada, que son: CHAIN y FIELD

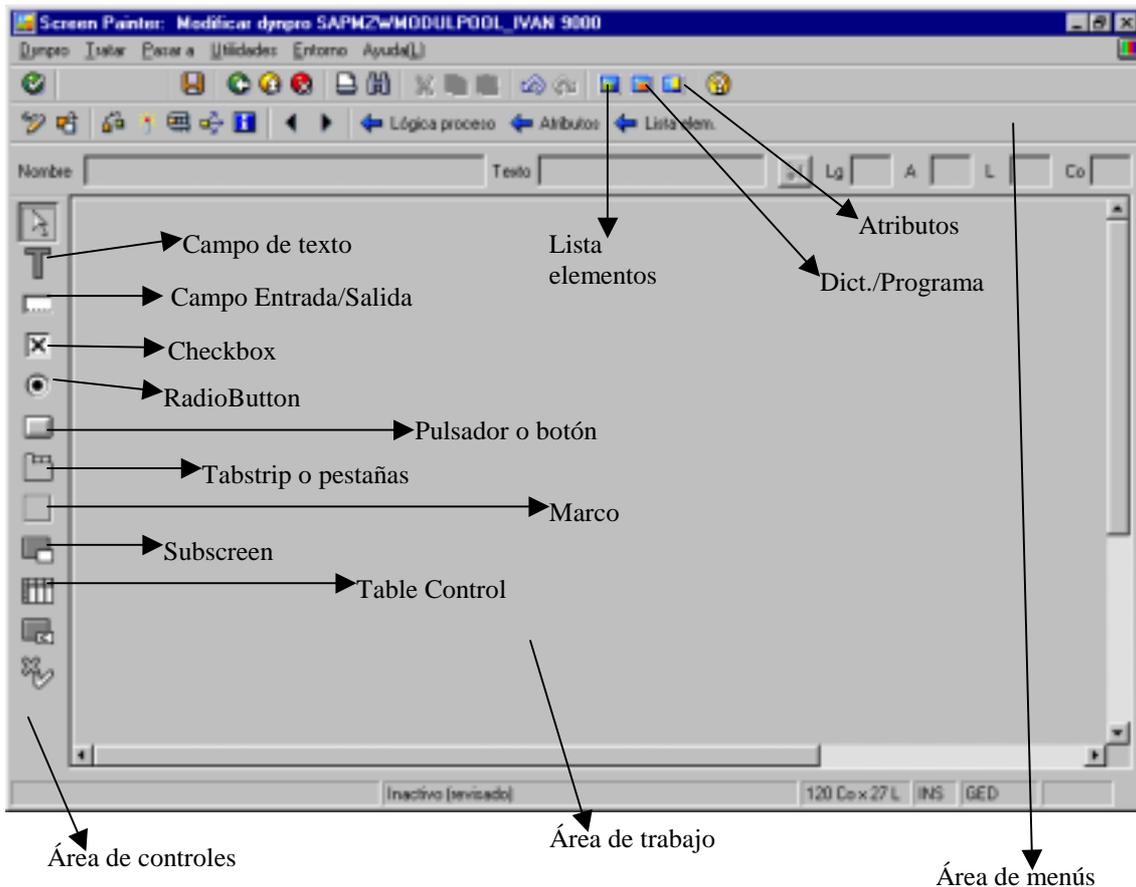
- FIELD → Instrucción que ejecuta un determinado cuando cambia el valor de un campo.
- CHAIN → Es igual que el CHAIN pero este agrupa campos en pantalla. Es decir, podemos agrupar 5 campos en pantalla y cuando algunos de ellos cambie de valor entonces ejecutar un determinado módulo.

EDITOR DISPOSICIÓN DE UNA DYNPRO

El editor de disposición o editor gráfico, como he dicho antes, es donde se colocan los campos en pantalla y es donde se configuran el aspecto de los campos. Para acceder a ella siempre dispondremos del siguiente botón:



Cuando pulsemos al botón nos saldrá la siguiente pantalla:



Como vemos hay tres áreas diferenciadas que son:

- Área de los controles → Es donde seleccionaremos el objeto que insertaremos en el área de trabajo.
- Área de trabajo → Es donde se colocan los objetos seleccionados en el área de controles. En esta área es donde podremos modificar los atributos de cualquier objeto que pongamos.
- Área de menús → Es donde están los botones para generar, activar, cambiar de dynpro, ir a los atributos de dynpro, etc... Esta *área de menú* se verá más adelante.

ÁREA DE LOS CONTROLES

Los objetos que podemos insertar son los siguientes:

- Campo de texto → Con este objeto podemos escribir cualquier texto para que se visualice permanentemente porque su valor no varía en tiempo de ejecución.
- Campo Entrada/salida → Como en el *Campo de texto* podemos insertar cualquier texto, pero con la diferencia que con este objeto podemos introducir cualquier valor en tiempo de ejecución. Gracias a una propiedad en el campo podemos convertirlo en un *listbox*.
- Checkbox → Campo de longitud 1 donde podemos activar / desactivar.
- Radiobutton → Es un conjunto de campos de entrada de longitud 1 sobre el que se puede activar o desactivar uno y sólo uno de ellos.
- Pulsador o botón → Control que cuando se pulsa ejecuta las instrucciones que tengan asociadas.
- Tabstrip o pestañas → Control que permite separar la información en pestañas, un ejemplo, es la pantalla que nos ha aparecido al crear la dynpro. En un *Tabstrip* solo se pueden poner *Subscreen*.
- Marco → Es un recuadro que sirve para una separación visual de los datos.
- Subscreen → Control donde se visualizarán los datos que queramos de una forma independiente a la dynpro principal. La llamada a la dynpro que aparece en la *Subscreen* creada se realiza en el PBO a través de la sentencia *Call subscreen*.

Hay que recordar los siguiente puntos:

1. Es bueno ponerle un nombre sencillo y claro. Para saber rápidamente cual es la función del control.
2. Algunos controles como los *Radio buttons*, *Checkbox*, *Campo Entrada/salida* es necesario declarar un variable con el mismo nombre y tipo. Aunque para hacer esto es recomendable utilizar el botón *Dict./Programa*.
3. Y como no, una vez creado la dynpro, la lógica de proceso, etc... Habrá que activar la dynpro.
4. Si queremos modificar el valor de cualquier control en tiempo de ejecución, el control ha de estar de declarado en el programa marco o en alguna *include* del programa marco cuando vayamos ha modificar el valor del control en tiempo de ejecución.

ÁREA DE MENÚ

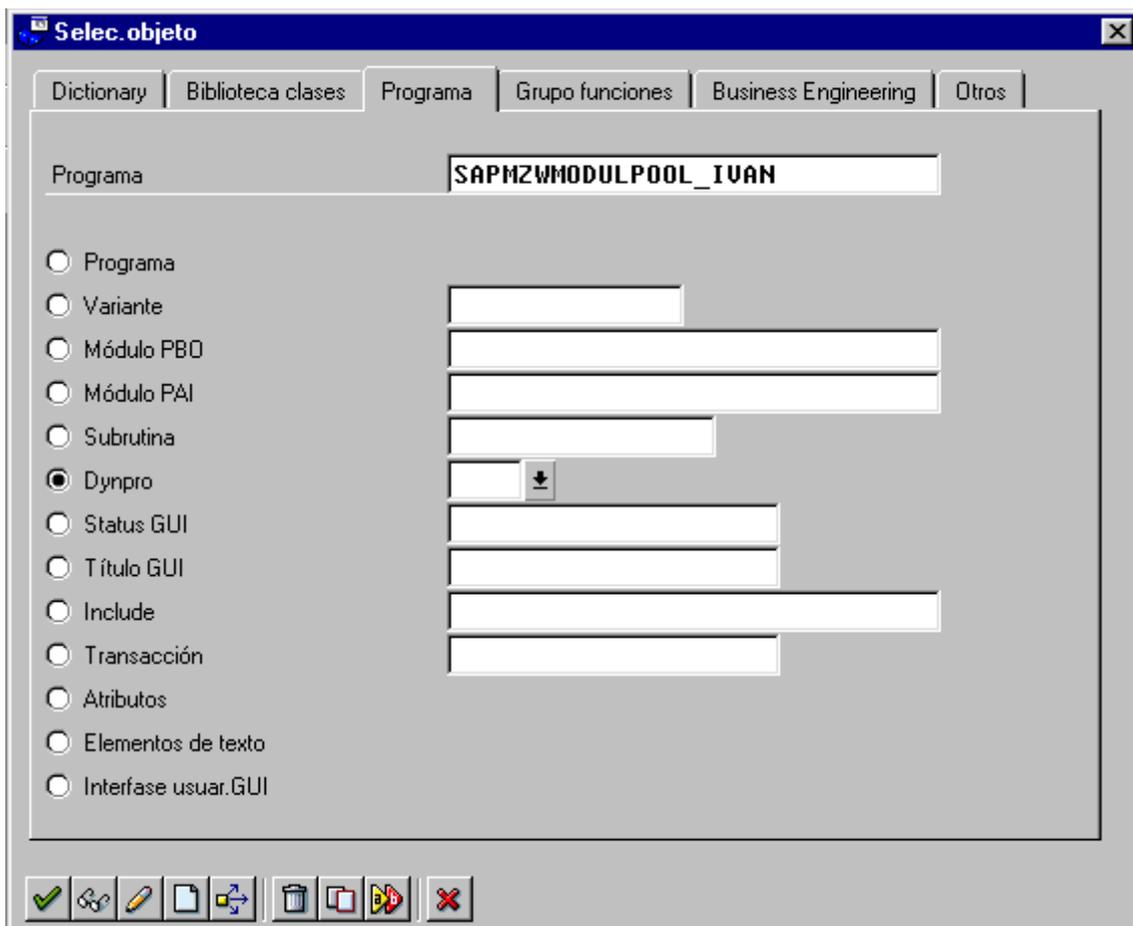
En el Área de menú es un *Menu painter* (se explicara más adelante su funcionamiento) con los botones necesarios para llevar el control del *Editor de disposición*. Disponemos de los siguientes botones.



→ Este botón sirve para intercambiar entre el modo de visualización y modificación.



→ Si pulsamos este botón podremos cambiar de objeto. Es decir, cambiar de dynpro, programa, etc... La pantalla que nos saldría sería la siguiente:



Como vemos se puede cambiar a cualquier objeto del sistema.



→ Botón que verifica si la dynpro y su lógica de proceso son correctas.



→ Con este botón activamos la dynpro. Es importante generarla porque sino SAP ejecuta la última vez que se activo la dynpro y los cambios efectuados a posteriori no tendrán efecto sino se activa.



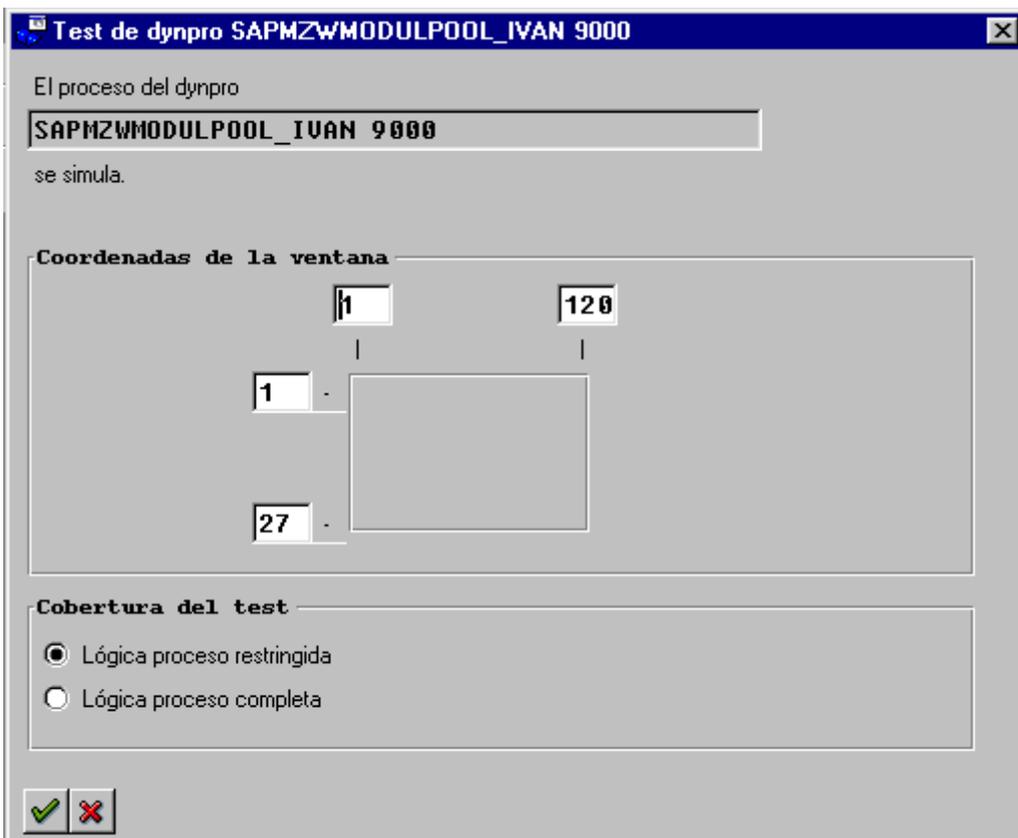
→ Con este haremos un test al dynpro. En el test solo veremos como quedan los campos en pantalla, si hemos puesto lógica de proceso se ejecutara de manera restringida o completa, según indiquemos. SAP pone un *Menu painter* por defecto. Veremos como funciona en el apartado *Modo TEST*



→ Se utiliza para saber en que programa, Mod. Interact., transacciones se utiliza el programa y dynpro que estamos tratando.

Modo TEST

El modo TEST es una utilidad que nos permite ver como queda una *dynpro*, si le damos a ejecutar veremos que aparece la siguiente ventana:



En esta pantalla tenemos dos zonas que son:

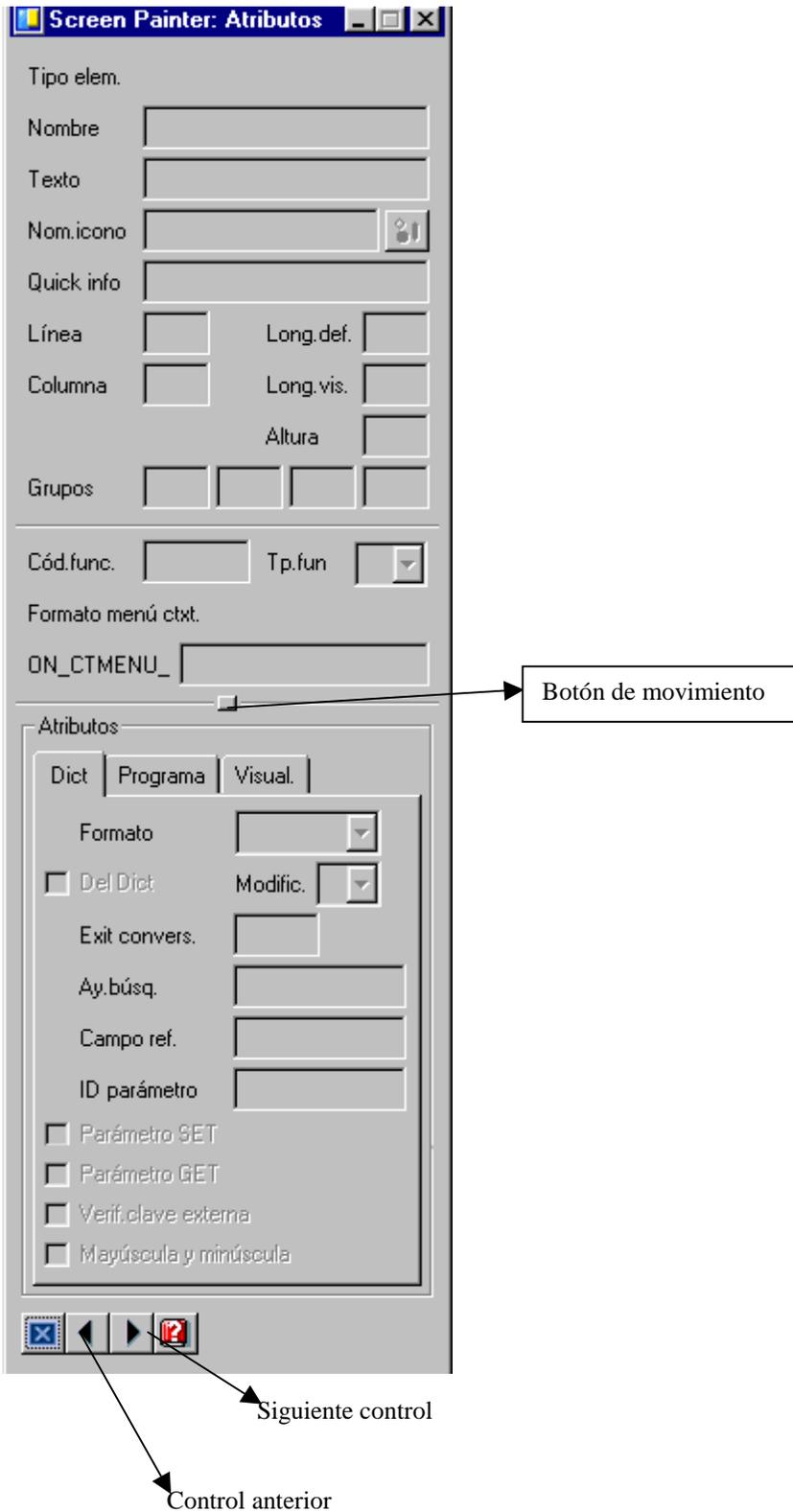
- Coordenadas de la ventana → Indicaremos la anchura y altura con se visualizara la dynpro.
- Cobertura del test → La lógica de proceso se ejecutara de dos maneras diferentes:
 1. Lógica proceso restringida → La dynpro se ejecuta, ejecutando las instrucciones que no van a ocasionar error.
 2. Lógica proceso completa → Todas las sentencias de la lógica de proceso se ejecutarán.

ATRIBUTOS

En cualquier campo que tengamos en el área de trabajo podremos ver y modificar sus atributos pulsando F2 o sobre el siguiente icono:



La pantalla que nos saldrá será la siguiente:



Dependiendo del tipo campo habrá algunas opciones en los atributos que estén desactivadas o ocultas. Para mover esta ventana con libertad es recomendable utilizar que utilizar el botón de movimiento, el que se indica en la pantalla anterior.

El significado de los campos es el siguiente:

- Nombre → Es el nombre del campo, que pongamos en pantalla. En el único control que poner el nombre es opcional (SAP ya pone uno) es en el *campo de texto*. No puede haber ningún *Nombre* repetido en la dynpro.
- Texto → Es el texto de salida de un campo.
- Nom. Icono → Ha algunos controles (excepto a: *campo de entrada*, *Subscreen* y *Table control*) podemos ponerle un icono que aparece a la izquierda del texto que pongamos. En el control *Tabstrip* no podemos ponerle un icono, pero sí a cada una de las pestañas.
- Quick. Info. → Cuando en un control o campo ponemos un icono, cuando estemos en tiempo de ejecución y el cursor este sobre ese control aparecerá un “bocadillo” con el texto que hayamos puesto en este campo. SAP pone un texto al poner un icono pero lo podemos cambiar.
- Línea → En que línea aparece el control. Podemos cambiar el valor y cuando pulsemos veremos en que posición se nos queda.
- Col → Lo mismo que en línea, pero aquí en columna esta el campo.
- Desplazal. → Si esta activo permite desplazarse por el contenido del campo. Es decir si el campo mide 10 caracteres y el valor que contiene mide 20 caracteres solo veremos los 10 primeros, pero con este *Checkbox* activo podemos navegar por los 10 caracteres restantes. Si no esta activo solo veríamos los 10 primeros sin la posibilidad de ver los 10 últimos. Cuando activamos este campo el campo *VisLg* se activa.
- Dropdown → Permite convertir un *campo de entrada / salida* en una lista despegable o un cuadro combinado. Si vale SPACE la salida es de un *Campo de entrada / salida* normal. Si vale ‘L’ la visualización sé efectúa como una lista despegable y no se puede hacer entradas directas. Si vale ‘C’ la visualización sé efectúa como un cuadro combinado y es posible hacer entradas directas, esta opción no esta soportada por SAP.
- Vislg → Este campo indica cuantos caracteres se podremos ver cuando el *Checkbox Desplazal*. Este activo. Siguiendo el ejemplo anterior, si este campo tiene el valor 15, solo podremos navegar por los 15 primeros caracteres, pero no lo podremos hacer con los otros 5.
- Def.long. → Indica la longitud del campo.
- Alt. → Es la altura del campo o control.
- Grupos → Podemos englobar un control o varios en un grupo para poder hacer modificaciones globales en control que pertenezcan a un grupo determinado. Un control puede pertenecer a cuatro grupo diferentes. El nombre del grupo es de 4 caracteres.
- Cód func. → Aparece en determinados controles (como en los botones o pulsadores) y es básico ponerlo para saber que control se ha pulsado. Ya que cuando pulsemos un botón el valor que contendrá la variable SY-UCOMM o el campo de tipo *OK* (visto anteriormente) será lo que valga *Cód.Func.* y no el nombre del control. Como en el campo *Nombre* no puede haber ningún *Cód. Func.* repetido.
- ON_CTMENU_ → Es el menú contextual.

Abajo tenemos *Atributos* que tiene 3 pestañas: *Dict*, *Programa* y *Visual*. En la pestaña *Dict* tenemos los siguientes valores:

- Formato → Indica el formato de un campo. Si es un campo del *Data Dictionary* el formato lo pone SAP (y no se puede modificar). Si es un campo nuestro tendremos que indicarle un formato, aunque por defecto es de tipo CHAR.
- Del Dict. → Sólo válido para los campo de diccionario. Si lo activamos el campo toma las mascara de edición y el texto de salida de la referencia que tenga el campo en el *Data Dictionary*.
- Modific. → Solo se activa cuando el *Checkbox Del Dict.* esta activo y sirve para escoger el tipo de mascara de edición o el texto de salida.
- Exit convers. → Si queremos realizar una conversión diferente a la que hace SAP tendremos que poner en este campo la rutina que queremos que se ejecute para realizarla. Este campo solo está disponible cuando el *Checkbox Del Dict.* esta desactivado.

- ID parámetro → Los campos del *Data Dictionary* cogerán el valor ID (este valor se utilizara para pasar los parámetros SPA/GPA) ahí indicado. Los que no lo sean se podrá poner el ID que queramos (hasta 20 posiciones). Este campo esta activo cuando el *Checkbox Del Dict.* esta desactivado.
- Parámetro Set → El *Checkbox Del Dict.* esta desactivado. Al activar el parámetro SET, el sistema almacenará el ID del parámetro correspondiente al campo el valor ahí indicado. Para ello es necesario indicar el ID del parámetro correspondiente.
- Parámetro Get → Al activar el parámetro GET no se visualizará el campo con valor inicial sino que con el valor definido previamente en la memoria SAP. Para ello es necesario indicar el ID de parámetro correspondiente.
- Mayúscula y Minúscula → Al activar este atributo se permitirá hacer entradas en mayúsculas y minúsculas, no se hará ninguna conversión a letras mayúsculas. Este atributo o solamente se debería aplicar en campos *Char*.

En la pestaña *Programa* tenemos las siguientes opciones:

- Campo Entrada → Si se ha activado el atributo 'Entrada' con 'X' se podrá indicar datos en este campo durante la ejecución de la transacción. Si sólo se ha marcado 'Salida', no se podrá indicar ningún valor. Es necesario marcar uno de los dos atributos.
- Campo Salida -> Si se ha marcado 'Salida', el sistema visualizará los contenidos de campo durante la ejecución de la transacción. Es necesario marcar por lo menos uno de los atributos.
- Sólo cpo. Salida → Este valor sólo es oportuno para campos con máscara de edición. Un campo con máscara de edición se distingue de un campo de texto aun durante el tiempo de ejecución en su representación cuando ha sido definido como campo de salida y no de entrada. Al activar el indicador con máscara de edición para salida no será posible diferenciar su representación durante el tiempo de ejecución de la de un campo de texto. De esta forma es posible insertar variables en textos aparentemente fijos.
- Campo obligatorio → Al activar este atributo, el campo se convierte en una campo de entrada obligatoria. La máscara de edición tendrá un signo “?” en la primera posición si el campo no está en un loop.
- Pul entr. Posibles → Sólo se puede asignar este atributo a *Campos de entrada / salida* (máscaras de edición). De esta forma, se puede controlar de qué modo se soporta una Ayuda para entradas para el *Campo de entrada / salida* en el tiempo de ejecución.
- Alin. Derecha → Al activar este atributo se visualizarán los campos alineados a la derecha. Este atributo sólo se puede utilizar en formatos numéricos para campos de entrada, o en campos de salida.
- Entrada * → Al activar este atributo se podrá indicar un '*' en la primera posición del campo.
- Sin anulación → Al activar este atributo no es posible eliminar las entradas hechas en este campo con el signo de anulación '!'.
• Sin masc. edición → Al activar este atributo no se utilizará ninguna máscara de edición para este campo. De esta forma se podrán indicar todos los caracteres deseados, también los caracteres especiales '!', '?', '_' que generalmente se utilizan para funciones especiales. Este atributo sólo se podrá utilizar en campos *Char*, especialmente para casos donde se deben poder indicar y visualizar todos los caracteres especiales.

En la pestaña *Visual* tenemos las siguientes opciones:

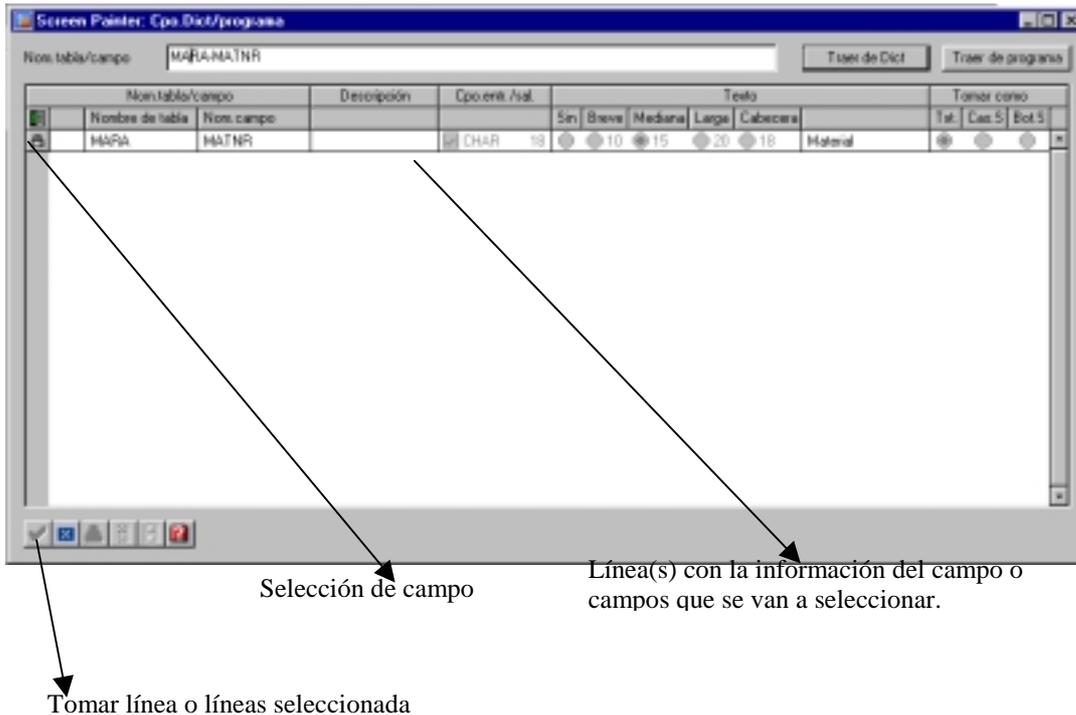
- Tp. Letra equidist → Por lo general, se dará salida a los campos de texto (palabras clave) y a los *Campos de entrada / salida* (máscaras de edición) en tipos de espaciado proporcional durante el tiempo de ejecución. Si éste atributo está seleccionado, durante el tiempo de ejecución se utilizará, por el contrario, el tipo de letra equidistante.
- Visible → Al activar este atributo el campo será visualizado de forma brillante.
- Invisible → Al activar este atributo no serán visibles los valores entrados y visualizados en este campo.
- De 2 dimensiones → Este indicador sólo es oportuno para campos con formato de edición (campos de entrada y de salida). Los formatos de edición generalmente aparecen de forma tridimensional durante el tiempo de ejecución. Al activar este indicador la representación será bidimensional.
- Sensible al doble clic → Este atributo sólo tiene sentido para campos de texto y *Campos de entrada / salida*. Cuando el atributo está fijado, durante el tiempo de ejecución se visualizará que un doble clic lanza una acción (punto de activación).

CAMPOS DICTIONARY/PROGRAMA

Esta es una utilidad que facilita muchas cosas a la hora de insertar campos en la dynpro. Para que nos salga tenemos que pulsar F6 o el siguiente botón:

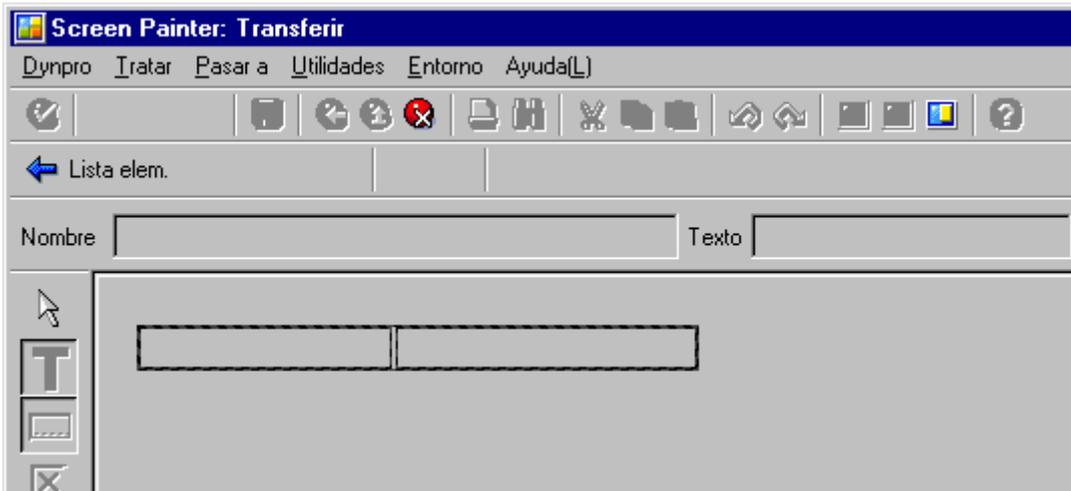


Con esto podemos insertar en la dynpro un campo de una tabla interna, una variable o un campo de diccionario. En los dos primeros tipos es necesario que estén declarados en el programa marco y dicho programa debe estar activado. En caso de un campo de diccionario SAP nos pone el texto de salida de manera automática. La pantalla es la siguiente:



La pantalla que sale es la siguiente. El funcionamiento es sencillo:

1. En nom. Tabla/campo → Pondremos el nombre de tabla (interna o de diccionario) o campo (tabla interna / diccionario o variable)
2. Seguidamente pulsaremos sobre el botón *Traer de Dict.* (si es algún campo o tabla que sea del diccionario) o en *Traer de programa* (en caso que sea una tabla interna o variable) Si le damos *ENTER* asume que le hemos dado al botón de *Traer de Dict.*
3. Si todo es correcto entonces SAP insertará una línea con los datos que ha encontrado. Digo correcto porque los datos han de estar activos y han de existir tanto en el diccionario o en el programa.
4. Para seleccionar la línea o líneas tenemos que pulsar sobre el botón de *Selección de campo* (como se indica en la figura) y después pulsar el botón de *Tomar líneas*. Si aparece el candado (como en este caso) significa que el campo ya está en el dynpro y no podemos seleccionarlo.
5. Una vez seleccionado moveremos el campo con el cursor hasta la posición deseada y haremos *click* para que se quede fijo en la posición que hayamos deseado.



Como vemos los recuadro es del campo MARA-MATNR, el de la izquierda es el texto de salida (que es un *campo de texto*) y la derecha es el *campo de entrada / salida*. Mientras el recuadro este así, lo podremos mover libremente hasta que hagamos un *click* con el ratón, que entonces se posicionara en la pantalla. Y quedaría de la siguiente forma:



Así sería su aspecto final.

Para insertar los campo de una tabla interna o tabla de diccionario, en este caso lo haré de una tabla interna, se haría:

Primero declarar una tabla en el programa marco y generar el programa marco. Segundo ir a la pantalla de *Campo Dict./Prog.* y poner el nombre de la tabla declarada (en mi caso la "T" que tiene la estructura de la tabla USR21) y pulsar sobre *Traer Programa* y el resultado sería este:

No.	Non tabla/campo		Descripción	Cpo. ent./sal.	Texto					Tonal		
	Nombre de tabla	Non campo			Sin	Breve	Mediana	Larga	Cabecera	Tit.	Cac.	
	T	MANDT		<input checked="" type="checkbox"/> CHAR 3	●	●	●	●	●		●	●
	T	BNAME		<input checked="" type="checkbox"/> CHAR 12	●	●	●	●	●		●	●
	T	PERSNUMBER		<input checked="" type="checkbox"/> CHAR 10	●	●	●	●	●		●	●
	T	ADORNUMBER		<input checked="" type="checkbox"/> CHAR 10	●	●	●	●	●		●	●
	T	KOSTL		<input checked="" type="checkbox"/> CHAR 8	●	●	●	●	●		●	●
	T	START_MENU		<input checked="" type="checkbox"/> CHAR 30	●	●	●	●	●		●	●

Botón de selección campos.

Si queremos seleccionar todos los campos es recomendable pulsar sobre el botón que indico con el nombre *Selección Campos*. (Una vez seleccionados todos, si queremos hacer el proceso inverso deberemos pulsar de nuevo sobre el campo de *selección campos*).

Tercero y último es tomar los campos y posicionarlos. Si lo hacéis, podréis comprobar que los campos no tienen *texto de salida*. Esto es debido a que los *textos de salida* solo están disponibles para los campos de diccionario.

VENTANA LISTA ELEMENTOS

Si pulsamos F7 o el botón:



Nos saldrá una ventana con todos los campos que se utilizan en la dynpro y con todos sus atributos (vistos en el apartado anterior de *Atributos*). Desde esta ventana podemos modificar los atributos de cualquier campo y lo que destaca es su aspecto visual. La pantalla que nos saldría es la siguiente:

The screenshot shows a window titled 'Screen Painter: Lista de elementos'. At the top, there are four colored buttons: blue for 'Atributos genet.', red for 'Atributos Dict', yellow for 'Atrib. progr.', and green for 'Atrib. visual'. Below these is a table with columns for field names and various attribute types. The table contains several rows of data, including 'USR21-BNAME', 'USR21-PERSNUMBE', 'USR21-ADDRNUMBE', 'USR21-KOSTL', 'USR21-START_MEN', and 'USR21-START_MEN'. Arrows point from labels to specific elements in the interface:

- Botón de selección (points to the blue button)
- Tipo de control (points to the 'Nombre' column header)
- Ir a Atributo generales (points to the blue button)
- Ir a Atributos de diccionario (points to the red button)
- Ir a Atributos de programa (points to the yellow button)
- Ir a Atributos Visuales (points to the green button)
- Atributo anterior (points to the left arrow button at the bottom right)
- Atributo siguiente (points to the right arrow button at the bottom right)

Como los datos se presentan de una forma muy visual y clara. La navegación se ha mejorado respecto a versiones anteriores gracias a los botones de ir directamente al tipo de atributo y los botones de ir al siguiente atributo. Además en la parte de la izquierda hay un icono que nos indica el tipo de control.

CONTROLES

En este apartado veremos de una forma simple el uso de los diferentes controles disponibles en SAP. En vez de hacer una explicación teórica de cada control que haría su comprensión más difícil, haré unos ejemplos sencillos para explicar los controles, en cada ejemplo combinaré varios controles. En estos ejemplos no haremos un *Menu painter* que serán explicados más adelante para no complicarlos. Una vez sepamos como crear *Menu painter* podemos ponerlos en los ejemplos siguientes.

Ejemplo 1

En el primer ejemplo haré una combinación de *Campos de texto*, *Campos entrada / salida* y *Botón*. Lo que voy hacer es poner un campo de entrada donde se pida el usuario y cuando se pulse un botón entonces mostrará el nombre completo del usuario. Los pasos a seguir son los siguientes:

Primero tenemos que declarar las tablas necesarias para la búsqueda en el programa marco. Las tablas son: USR21 y ADRP. El programa ha de quedar así:



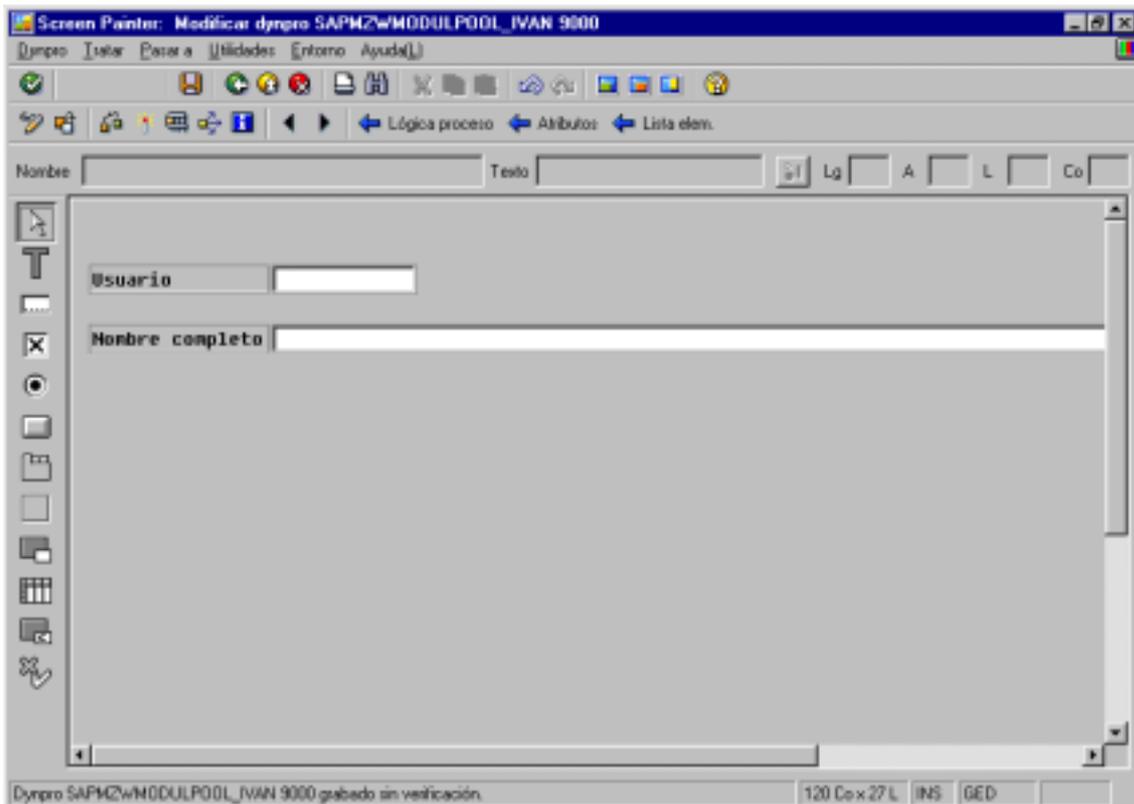
```
Modulpool      SAPMZWMODULPOOL_IVAN      activo
|-----|
|  program sapmzwmodulpool_ivan .
|
|  tables: usr21, adrp.
|
```

Una vez introducido las sentencias generaremos el *modulpool*.

Una vez tengamos el programa generado y activo iremos a colocar los controles. Tendremos que poner los siguientes controles:

- Campo de texto que ponga: *Usuario*
- Campo entrada para poder introducir el código de usuario.
- Campo de texto que ponga: *Nombre completo*.
- Campo de salida para que muestra el nombre completo.
- Un botón que busque el nombre completo del usuario introducido.
- Otro botón que nos permita salir de la dynpro (recordar que no hay *Menu painter*).

Para poner los 4 primeros controles es mejor ir a la ventana de *Campos Dict./Prog* y nos traemos del diccionario los siguientes campo: USR21-BNAME y ADRP-NAME_TEXT. El resultado sería el siguiente:



Queremos que el usuario no toque, o mejor dicho, no pueda modificar el nombre completo. Para ello haremos un *click* sobre el campo ADRP-NAME_TEXT y pulsaremos F2 para que nos salga la ventana de atributos del campo (es más rápido ir por teclado que no ir a al botón de *Atributos*) e iremos a la pestaña *Programa* y el *Checkbox Campo de entrada* los desactivaremos. Y veremos que el campo tendrá el siguiente aspecto:



Ahora el campo solo será de salida y no de entrada / salida como antes. Ahora solo nos falta poner los 2 botones.

Primero haremos el botón de *Salir*, para ellos insertamos el control *Botón* y le pondremos los siguiente atributos:

- Texto -> Salir de la dynpro
- Nombre -> SALIR
- Cód. Func -> SALIR

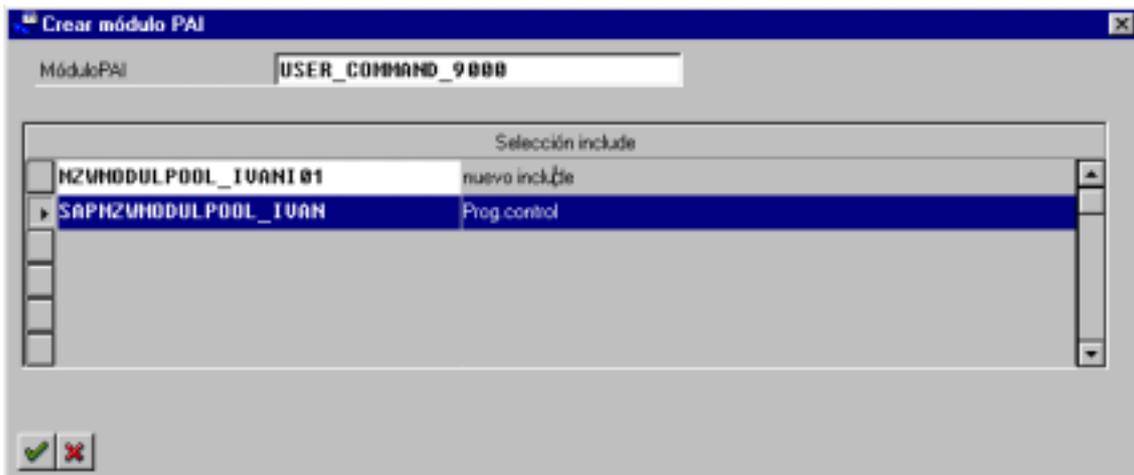
El campo *cód. Func* ha de ser rellenado obligatoriamente para controlar en el PAI que botón se ha pulsado.

El segundo botón será el de búsqueda y tendrá los siguientes atributos:

- Texto -> Buscar usuario.
- Nombre -> BUSCAR
- Cód. Func -> BUSCAR

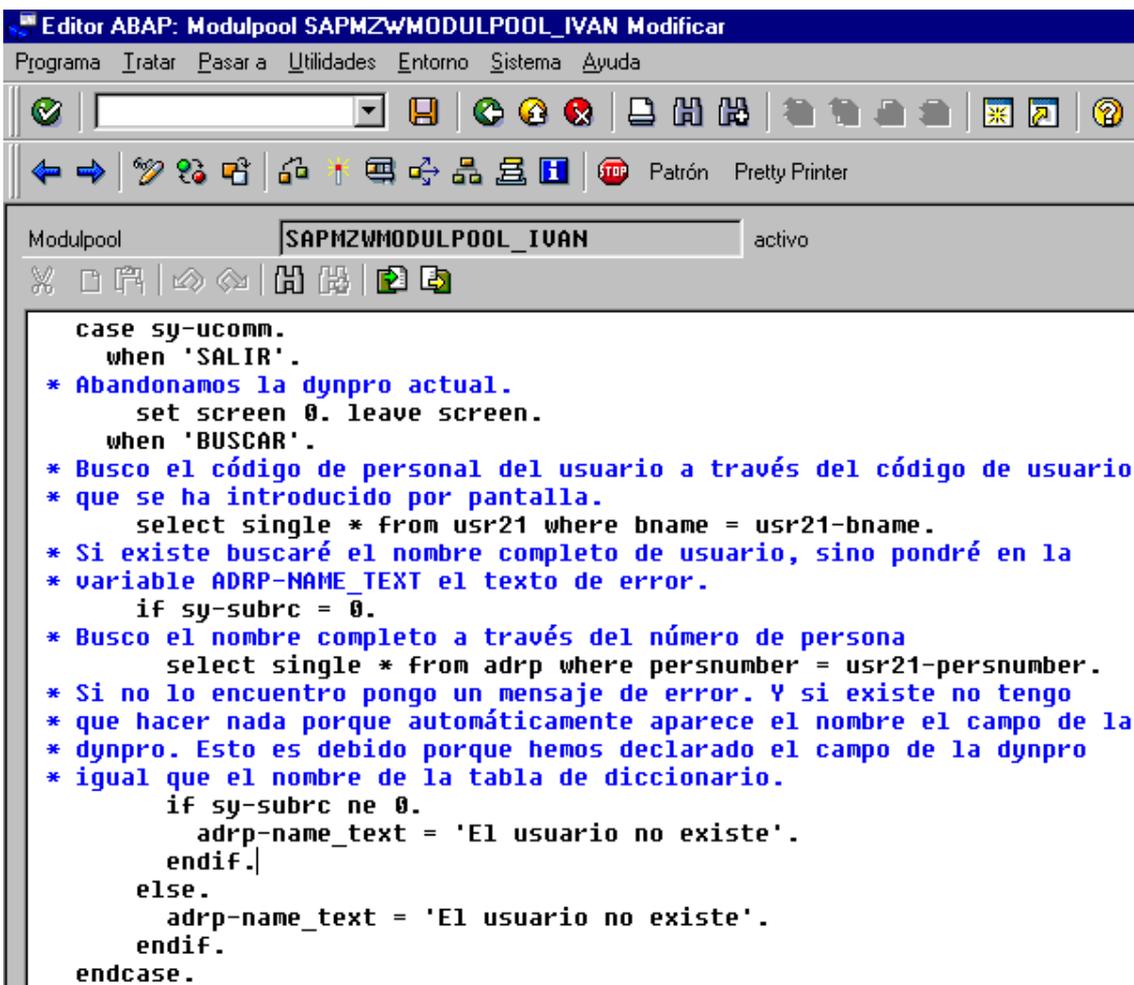
Una vez tengamos los controles puestos lo que haremos, y por este orden es: Grabar, Verificar y Activar. Una vez echo esto lo que tenemos que hacer es poner el código necesario para que los controles “se comuniquen”.

Como hemos visto en el apartado de *Lógica de Proceso*, SAP nos pone dos módulos (asteriscado inicialmente) uno para el control del PBO y otro para el del PAI. Nosotros desasteriscaremos el módulo del PAI *module user_command_9000* y luego pulsaremos *doble-click* para crearlo en el programa marco, es decir el siguiente pantalla:



Tenemos que seleccionar el *Prog. Control*, aunque para programas complejos es bueno separar los módulos del PBO y del PAI en diferente *Includes*.

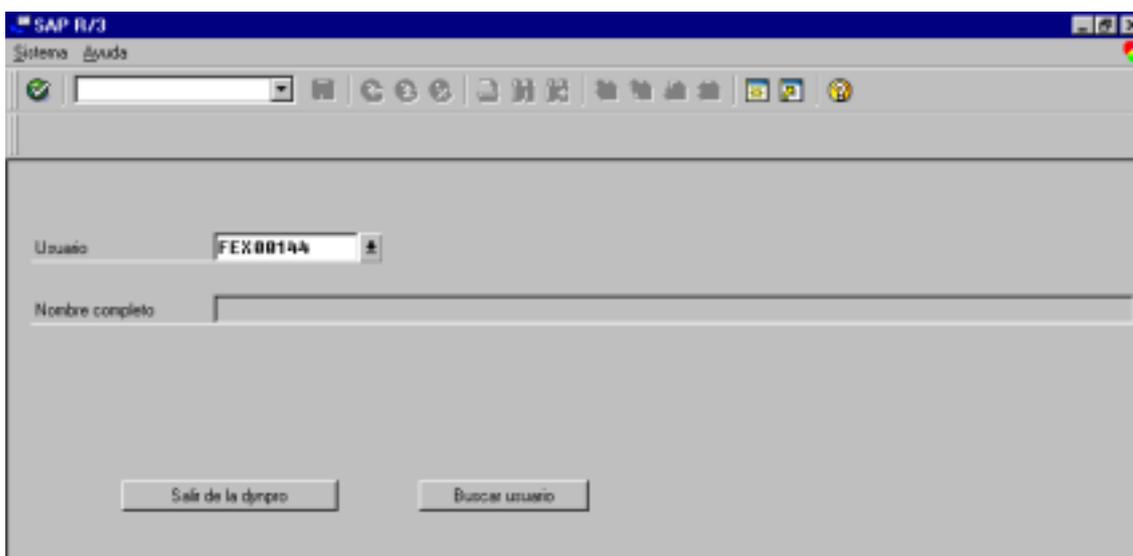
Una vez creado el módulo tenemos que pensar como vamos a controlar los botones o teclas que se pulsen. Para no complicarlo lo haremos con la variable del sistema SY-UCOMM. El código fuente del módulo sería el siguiente:



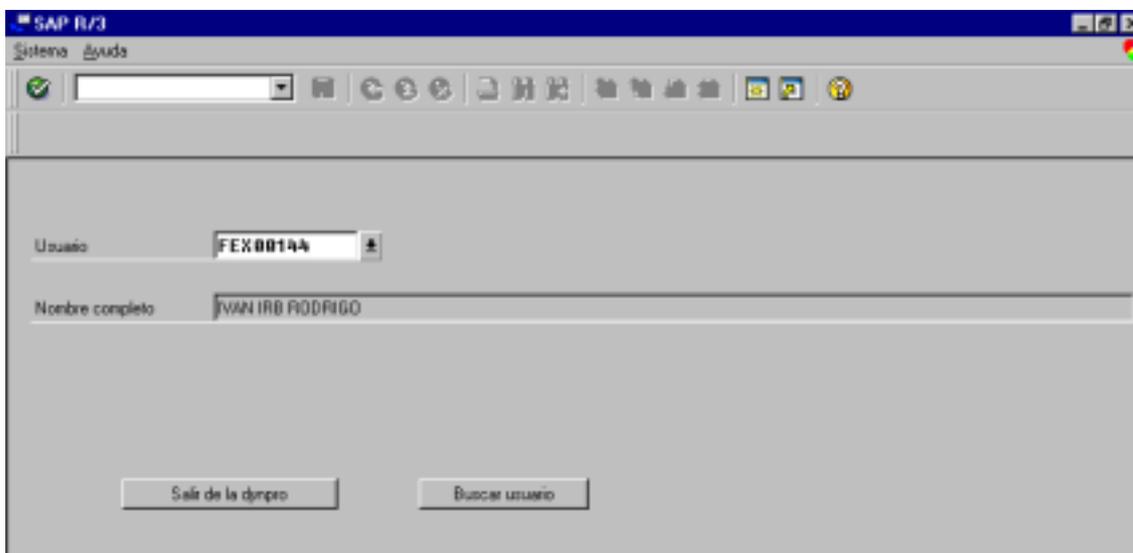
Después tenemos que grabar, verificar y Activar el *Modulpool*. Cuando volvamos atrás la lógica de proceso tiene que quedar de la siguiente manera:

```
PROCESS BEFORE OUTPUT.  
* MODULE STATUS_9000.  
*  
PROCESS AFTER INPUT.  
MODULE USER_COMMAND_9000.
```

No olvidados de activar la *Lógica de proceso* para que SAP tome los cambios que hayamos efectuados. Por último tendremos que crear la transacción (SE93) y asociarla a nuestro *Modulpool*. La pantalla principal nos saldrá de la siguiente forma:



Cuando pongamos el nombre de usuario y le demos a buscar nos saldrá esto:



Como vemos nos ha salido el nombre del usuario FEX00144, que justamente soy yo.

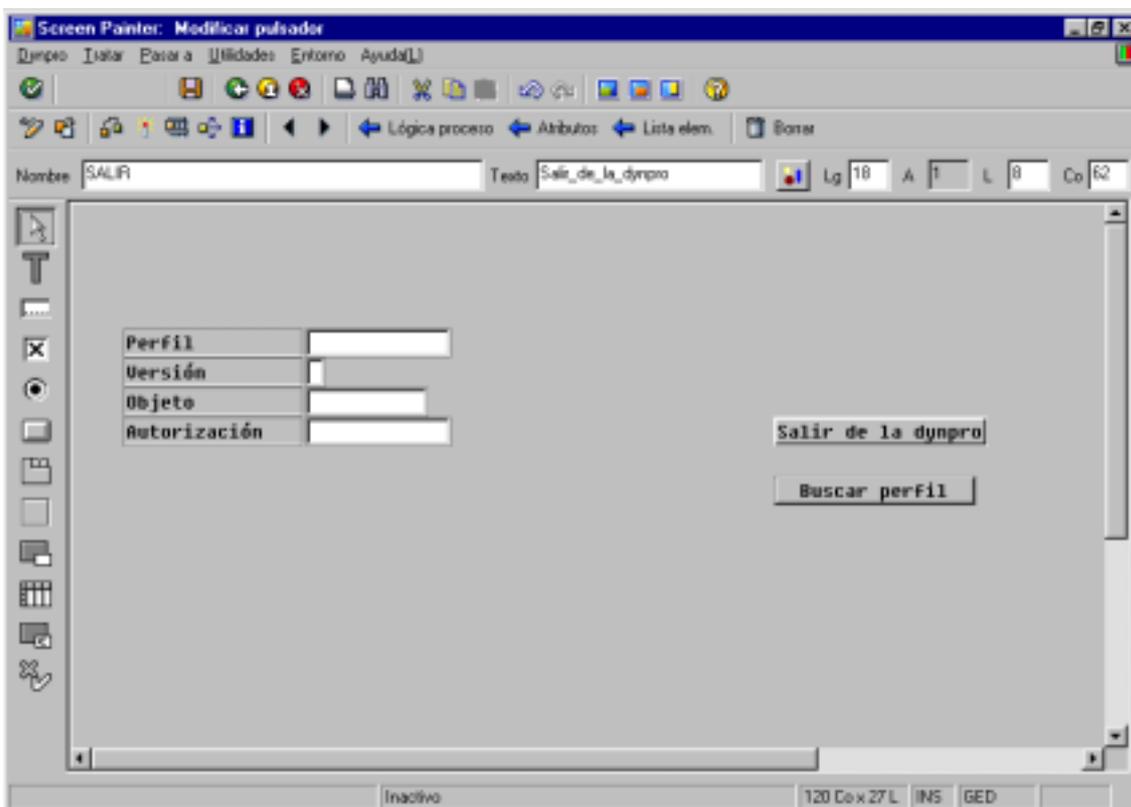
Ejemplo 2:

En este ejemplo crearemos un *Step loop*. Un *Step loop* es un *loop* de campos. Un *Step loop* lo podemos crear de un solo campo como de varios. Como ejemplo haremos que de un perfil nos muestre los objeto y autorizaciones que tenga.

Manual Básico de Modul Pool

Primero haremos dos botones, uno que nos permita salir de la dynpro y el segundo que es el que nos hará la consulta.

Lo segundo es poner todos campos de la tabla *UST10S* (recordar que la tabla ha de estar declarada en el programa marco y el programa ha de estar activo) en la dynpro, la pantalla quedaría así:

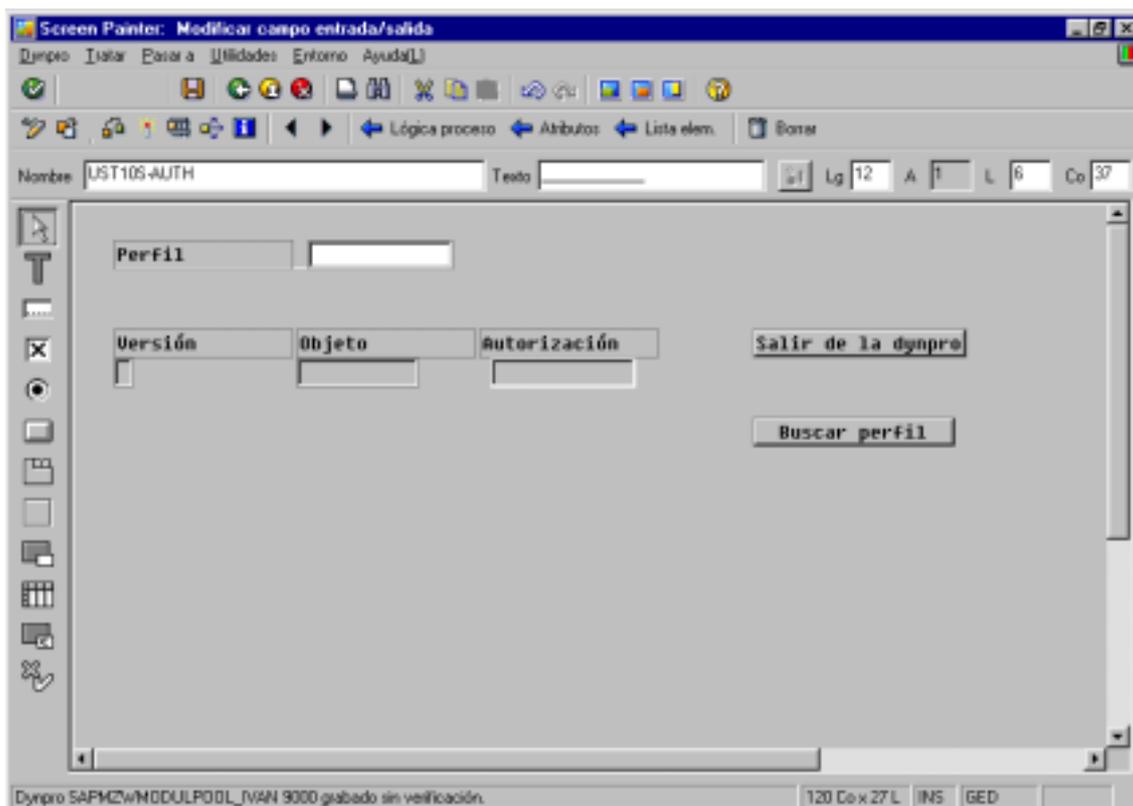


Cuando cogemos un campo de una tabla a través del *Campos Dict./Prog* veremos que SAP pone el mismo nombre tanto al *Campo de Salida* (que es el texto del campo) como al *Campo de Entrada* (el valor del campo en sí). Este pequeño problema no pasa nada si no utilizamos *Step loop* ni *Table control*, pero cuando lo hacemos tendremos que cambiar el nombre de los campos de salida de los campos debido a que SAP se hace un lío al ver que dos controles tienen el mismo campo y uno de ellos está en un *Step loop*. En nuestro caso tenemos que modificar los *Campos de Salida* de los campos:

- UST10S-AKTPS
- UST10S-OBJCT
- UST10S-AUTH

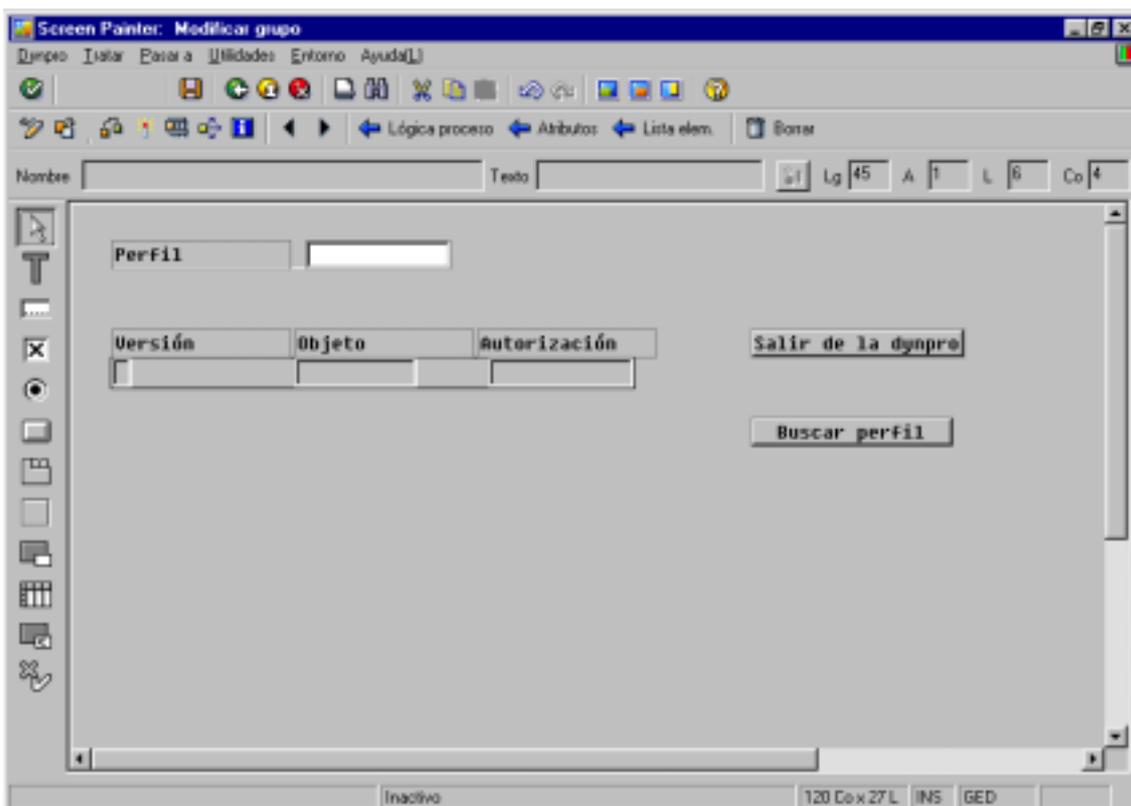
Yo suelo hacer lo siguiente: Si el campo se llama *UST10S-AKTPS* le quito el nombre de la tabla quedándose en *AKTPS*.

Una vez cambiado los nombres pondremos en orden los campos, o sea, como en esta pantalla:

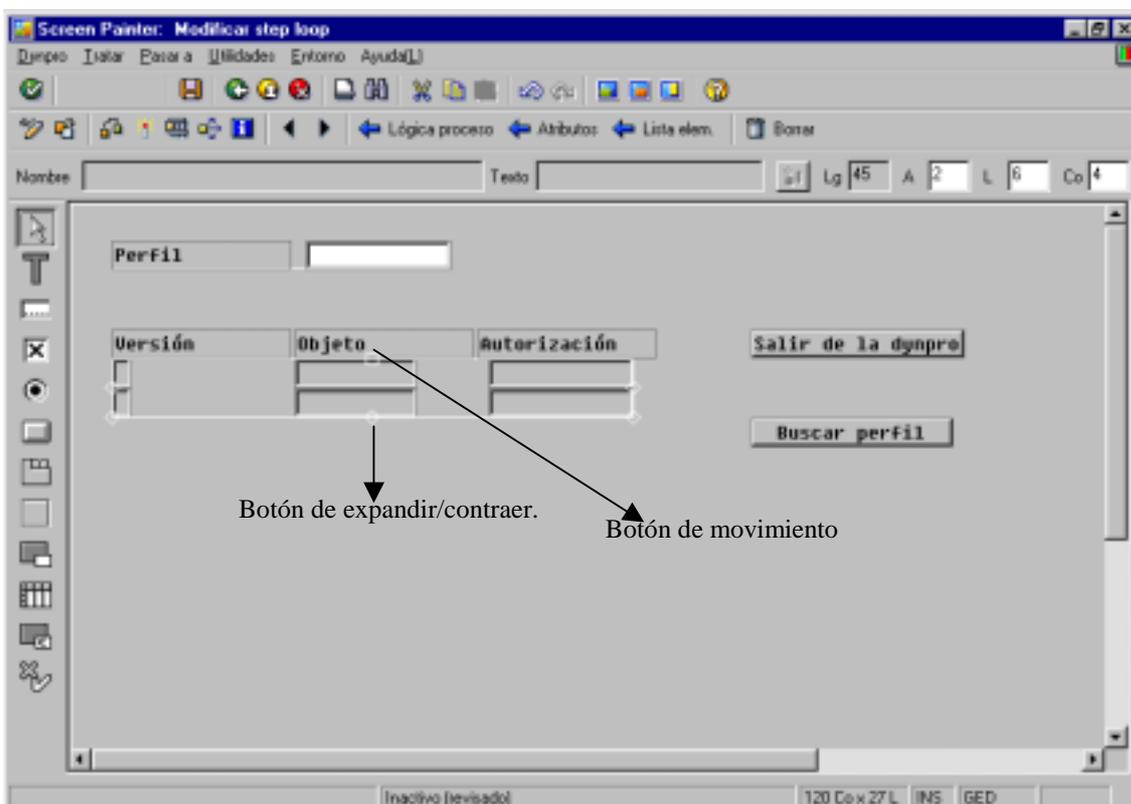


Como veréis en esta pantalla he hecho que los campos sean solo de salida (se ha visto en el ejemplo anterior como se hacía).

Ahora toca crear el *Step loop* de los tres campos, primero tenemos que englobar los campos. Para ello haremos un clic en la esquina superior izquierda del *Campo de entrada Versión* y moveremos el cursor, veréis que se hace un recuadro, hasta que el recuadro cubra los campos de entrada (NO hay que englobar a los texto de salida de los campos), o sea, hasta el esquina inferior derecha del *Campo de Entrada Autorización*. El resultado sería este:



Como vemos en la imagen el recuadro ha englobado los tres *Campo de Entrada*. Ahora iremos al menú: *Tratar* → *Agrupación* → *Step loop* → *Definir* entonces veremos que nos aparece la siguiente imagen:



Veremos que los campo englobados se nos han desglosado en dos. Los rombos que aparecen sirven para expandir o contraer el número de filas que queremos para el *Step loop* y el cuadro de la parte superior sirve para mover el *Step loop* en la posición que queremos.

Manual Básico de Modul Pool

Un *Step loop* es como mínimo de dos filas, aunque si intentáis reducirlo a uno veréis que SAP no lo permite. Los *Step loop* pueden ser de dos tipos:

- Fijo → Si un *Step loop* tiene cuatro filas y los datos que vamos a visualizar ocupan cinco filas el resultado es que solo se verían las cuatro primeras filas. Si hay menos datos a visualizar que filas tiene *Step loop*, el *Step loop* no se adaptara a los datos. Valor por defecto que pone SAP cuando creamos un *Step loop*.
- Variable → Este tipo de *Step loop* es como un acordeón, es decir, se expande o se contrae dependiendo de los datos que le pasemos. Si el *Step loop* es de dos filas y los datos van a ocupar cuatro, el *Step loop* se expande hasta que quepan dichos datos.

SAP solo permite tener un *Step loop* variable en una dynpro, pero no hay límites para los *Step loop* fijos.

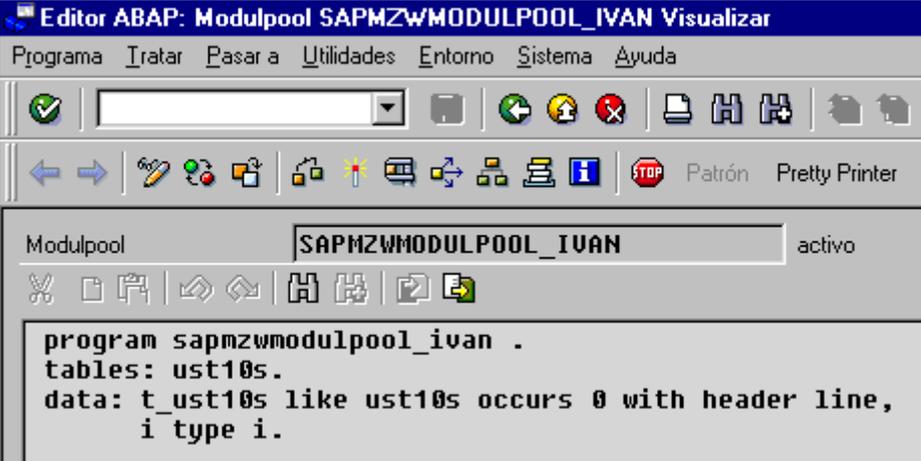
Para pasar de un *Step loop* fijo a uno variable tenemos que ir al menú *Tratar* → *Agrupación* → *Step loop* → *Variable*. Y para pasar de uno variable a fijo tenemos que ir al menú *Tratar* → *Agrupación* → *Step loop* → *Fijo*. Si queremos deshacer un *Step loop* tendremos que ir al menú *Tratar* → *Agrupación* → *Step loop* → *Desglosar*.

Recordar que para operaciones con un *Step loop* tenemos que seleccionarlo con anterioridad. Para ello solo tenemos que pulsar sobre el cuadro o los rombos del *Step loop*.

Una vez hayamos creado solo grabaremos la dynpro, porque si los verificamos o generamos nos dará error porque aún nos falta poner unas sentencias para que todo funcione correctamente.

Cuando creamos un *Step loop* tenemos que poner en la *Lógica de proceso* dos *loops*. El primero se coloca en el PBO y es para pasar los datos de una tabla interna a los campos del *Step loop* de la dynpro, y el segundo se coloca en el PAI y es para controlar que campos se han modificado.

Las variables necesarias para que el *Step loop* funcione serían estas:



```
program sapmzwm modulpool_ivan .
tables: ust10s.
data: t_ust10s like ust10s occurs 0 with header line,
      i type i.
```

La tabla interna *t_ust10s* la utilizaremos guardar el resultado de buscar los objetos y autorizaciones de un determinado perfil. Y la variable “*i*” es una variable para controlar los registros que pasamos al *Step loop* (la variable ha de ser de tipo entero).

La lógica de proceso quedaría de la siguiente forma:

```

|
process before output.
* MODULE STATUS_9000.
* Leemos la tabla interna T_UST10S y pasamos los datos de
* dicha tabla a la tabla UST10S a través del modulo
* ASIGNAR_DATOS
  loop at t_ust10s cursor i.
    module asignar_datos.
  endloop.
*
process after input.
* LOOP que solo sirve para controlar que registros han
* cambiado
  loop at t_ust10s.
  endloop.
* Módulo para controlar las teclas de la dynpro.
  module user_command_9000.

```

En el PBO se pone el *loop* que sirve para pasar los datos de la tabla interna *t_ust10s* a los campos del *Step loop* a través del módulo *asignar_datos* (recordar que en la lógica de proceso solo se pueden poner determinadas sentencia y determinadas variantes de dichas sentencias.). La variante *cursor* "i" del *loop* se utiliza para controlar que registro sé esta insertando en el *Step loop*. El contenido del módulo *asignar_datos* sería el siguiente:

```

module |asignar_datos output.
ust10s = t_ust10s.
endmodule.                " ASIGNAR_DATOS  OUTPUT

```

Vemos que es una asignación simple debido a que la tabla *t_ust10s* tiene la misma estructura que la tabla *ust10s*.

Hay que destacar que el módulo que se utiliza para poner un *Menu painter* y un título a la dynpro están asteriscados.

En el PAI lo primero que se ejecutaría es el *loop* de la tabla interna. Este *loop* solo se utiliza para saber que registro del *Step loop* se ha modificado. Aunque el *loop* no lo utilizemos es obligatorio ponerlo, ya que SAP obliga a tener dos *loop* por cada *Step loop* que tengamos declarado en una dynpro, uno el PBO y otro en el PAI.

El código del módulo *user_command_9000* sería el siguiente:

```

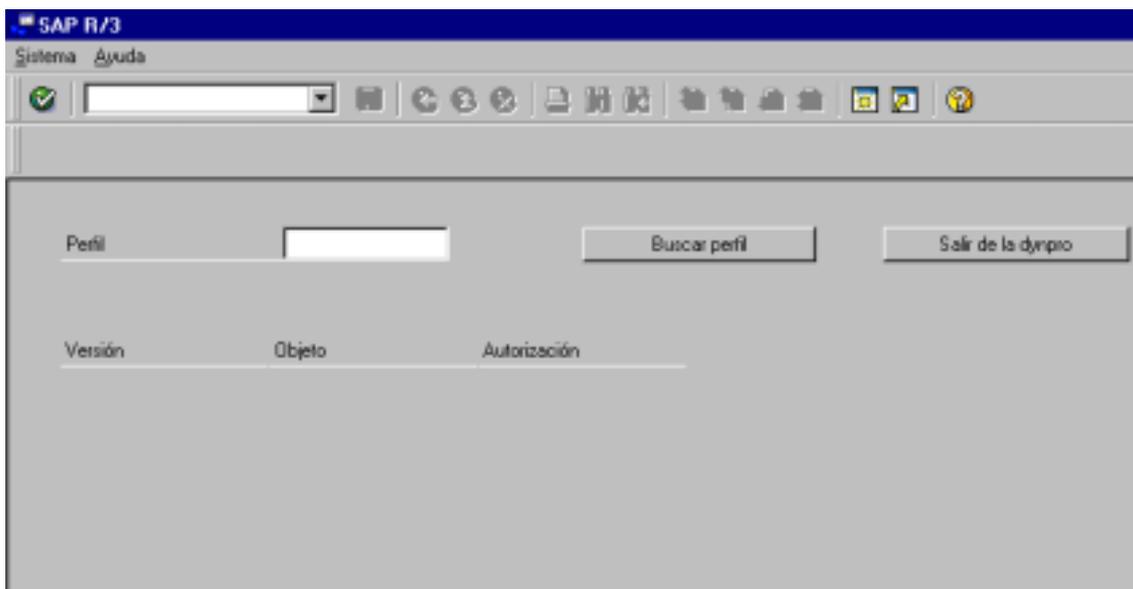
module user_command_9000 input.
  case sy-ucomm.
    when 'SALIR'.
* Abandonamos la dynpro actual.
    set screen 0. leave screen.
    when 'BUSCAR'.
* Primero vacio la tabla interna T_UST10S.
    refresh: t_ust10s. clear: t_ust10s.
* Busco en la tabla UST10S el perfil del campo de la dynpro UST10S-PROFN
* y el resultado de la búsqueda lo guardo en la tabla interna T_UST10S
    select * from ust10s into table t_ust10s
      where profn = ust10s-profn.
  endcase.
endmodule.                " USER_COMMAND_9000  INPUT

```

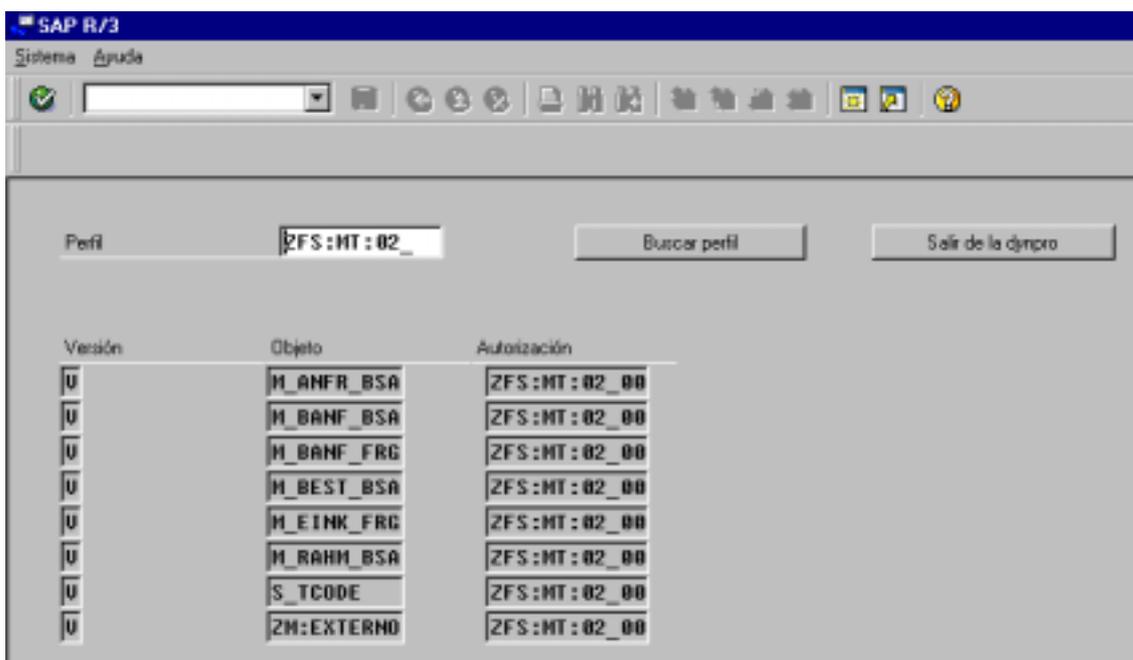
Como vemos el código de búsqueda es sencillo y no tiene ningún misterio.

Cuando hayamos escrito el código tanto en los módulos (que están el programa marco) como en la lógica proceso procederemos a grabar, verificar y activar tanto la dynpro como el *Modul pool* del programa marco. Como ninguno de los dos estará activado, SAP nos permite generar los dos a la vez.

Una vez generado solo tenemos que probarlo, para ello, ejecutaremos la transacción que hemos creado para el ejemplo anterior y la pantalla que nos tiene que salir es la siguiente:



Bueno, os saldrá igual exceptuando los botones que los he cambiado de sitio en el último momento. Para probarlo solo tenemos que introducir un perfil (los perfiles simples están en la tabla *ust10s*) y dar al botón *Buscar Perfil*. El resultado sería este:



Como vemos queda muy bonito, podemos insertar cualquier perfil y veremos que el *Step loop* al ser variable se expande y contrae igual que un acordeón.

Ejemplo 3:

En este ejemplo vamos a combinar un *Tabstrip* con *Subscreen*. Con ello haremos mostrar los perfiles de un determinado usuario.

Un *Tabstrip* es un control que esta formado por pestañas, en el entorno Windows y no Windows (MAC, Linux) es muy común verlo. Un ejemplo de un *Tabstrip* sería el siguiente:

Como vemos un *Tabstrip* esta formado por pestañas, puede tener tantas pestañas como queramos. Los *Tabstrips* presentan un problema es que no pueden tener controles como: *Campos de Entrada*, *Checkbox*, etc... en su interior. El único control que se puede poner es una *Subscreen*.

Subscreen es un control que llama a una dynpro creada en el programa de control. Por lo tanto una *Subscreen* puede contener cualquier control, incluso, poner otro *Tabstrip*.

El único problema es que dentro de una *Subscreen* no podemos llamar a otra *Subscreen*, salvo, que lo hagamos a través del control *Tabstrip*.

Nuestro *Tabstrip* estará formado por dos pestañas, que son:

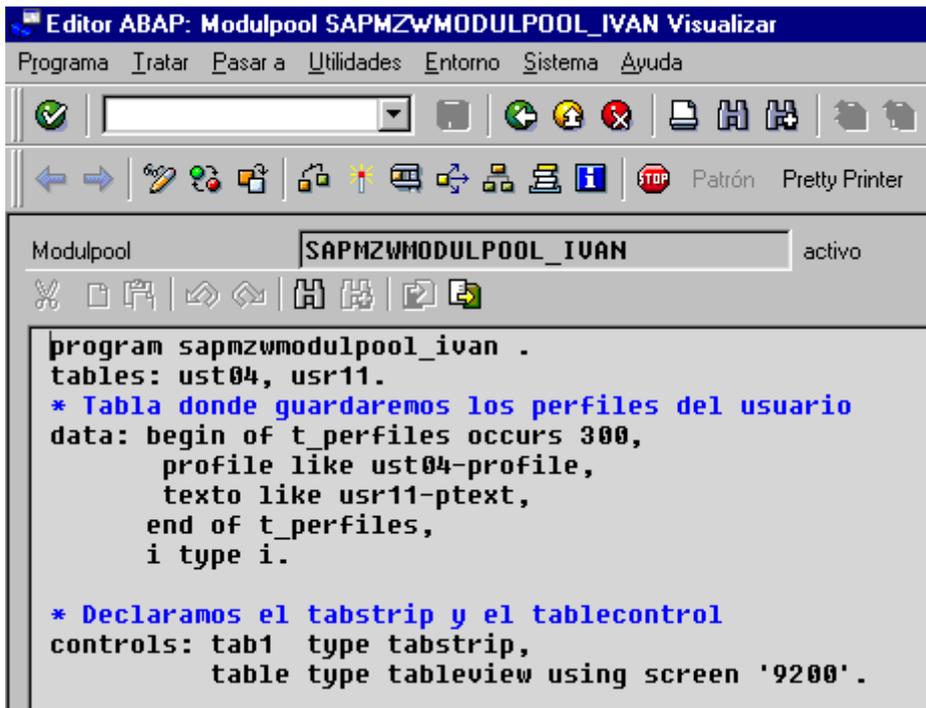
- En una habrá un *Campo de entrada* donde introduciremos el código de usuario y un botón para realizar la búsqueda de los perfiles de los usuarios.
- En la segunda pestaña haremos un *Table control* donde mostraremos los perfiles del usuario introducido en la primera pestaña.

Lo mejor es hacer las dynpros que serán llamadas desde la *Subscreen* y luego hacer la dynpro con el *Tabstrip* y las *Subscreen*. Para ello haremos las dynpros haremos en el siguiente orden:

- Dynpro 9100 → Será aquella que introduzcamos el código de usuario y cuando pulsemos el botón *buscar* nos ira a la pestaña donde nos saldrán los perfiles del usuario introducido.
- Dynpro 9200 → Es donde se mostraran los perfiles del usuario a través del *Table control*.
- Dynpro 9000 → Es la pantalla principal que contendrá el botón de salida y el *Tabstrip* que contendrá las dynpros anteriores.

Nota: Recordar que las dynpros 9100 y 9200 tiene que ser de tipo “*Subscreen*”, este parámetro se modifica en la pantalla de “*Atributos*” de una dynpro.

Pero antes de crear las dynpros vamos a declarar las variables que necesitamos de antemano, que son las siguientes:



Las tablas que utilizaremos son las siguientes:

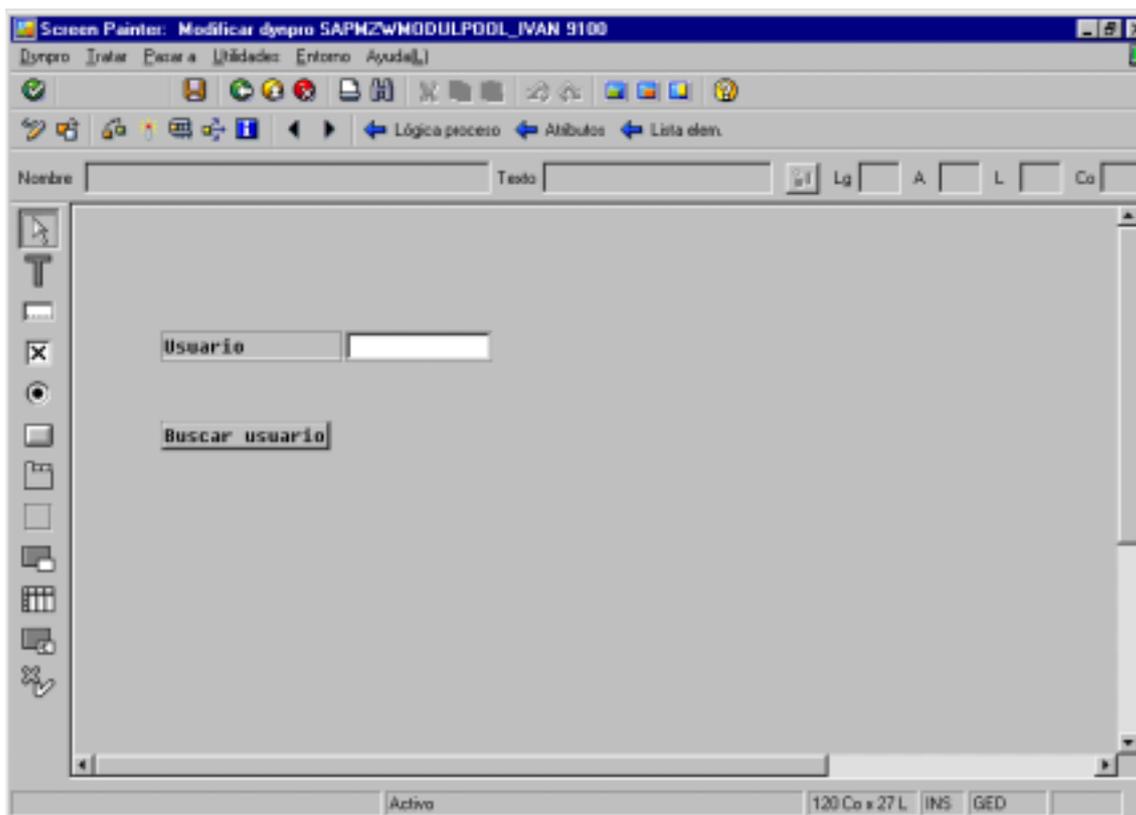
- UST04 → Tablas con los perfiles del usuario.
- USR11 → Tabla con la descripción de los perfiles.

La tabla interna *t_perfiles* es donde guardaremos los perfiles con su descripción. La variable *i* es para controlar los datos que insertamos en el *Table control*. Y en *controls* es donde declararemos los controles que utilizaremos que son:

- Tab1 -> Variable del *Tabstrip*.
- Table -> Es el *Table control* y tenemos que indicar en que dynpro utilizaremos dicho control.

Una vez declaradas las variables vamos al diseño de las pantalla:

La dynpro 9100 tendrá el siguiente aspecto:



En los *Atributos* de la dynpro tendremos que indicar que será de tipo *Subscreen*, si no lo indicamos se producirá un error en tiempo de ejecución cuando la *Subscreen* del programa principal llame a la dynpro.

El *Campo de entrada/Salida* y el *Campo de texto* donde se introducirá el usuario serán cogidos del campo *ust04-bname* (A través del *Cpo. Dict./Prog.*). Y el botón de *Buscar Usuario* tendrá el código de función *buscar*.

La lógica de proceso será el siguiente

```
process before output.  
* MODULE STATUS_9100.  
process after input.  
* Módulo para controlar las teclas de la dynpro.  
  module user_command_9100.
```

El código del módulo *module user_command_9100* es el siguiente:

```

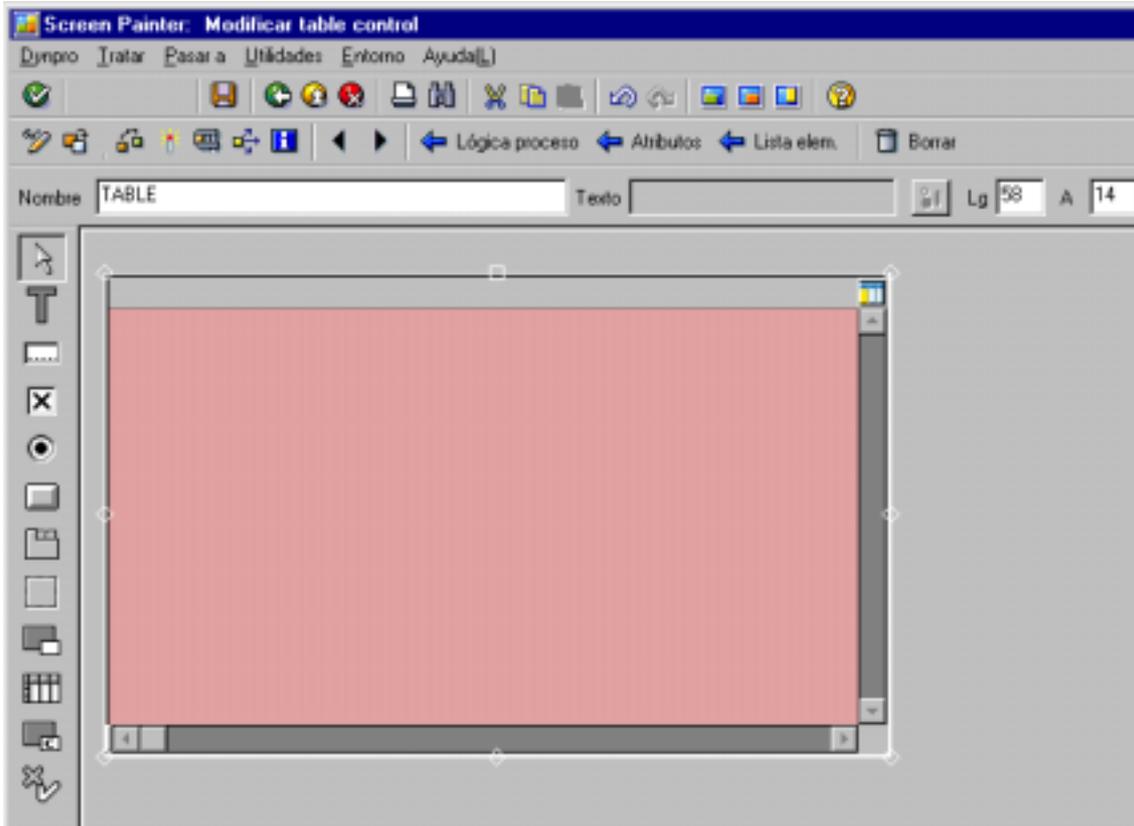
case sy-ucomm.
  when 'BUSCAR'.
* Vacio la tabla donde se guardan los perfiles del usuario
  clear: t_perfiles. refresh: t_perfiles.
* Busco los perfiles del usuario en la tabla UST04
  select * from ust04 where bname = ust04-bname.
         t_perfiles-profile = ust04-profile.
* Busco la descripción del perfil, indico que el perfil ha de estar
* activo y el lenguaje del perfil ha ser igual al del sistema.
  select single * from usr11 where profn = ust04-profile and
                                langu = sy-langu and
                                aktps = 'A'.

  if sy-subrc = 0.
    t_perfiles-texto = usr11-ptext.
  else.
    t_perfiles-texto = 'No hay descripción'.
  endif.
  append t_perfiles.
endselect.
* Una vez buscado los perfiles del usuario, activo la pestaña donde
* se visualizaran los perfiles.
  tab1-activetab = 'PERF'.
endcase.
endmodule.                                " USER_COMMAND_9100 INPUT

```

La dynpro 9200 será la que tenga el *Table control*, vamos como se haría paso a paso:

Primero es declarar el control en la zona de variables en el programa marco o en el include que tengamos para las variables, este paso ya lo hemos hecho anteriormente. Lo segundo es poner el control en la dynpro, que es este:



En los *Atributos* de la dynpro tendremos que indicar que será de tipo *Subscreen*, si no lo indicamos se producirá un error en tiempo de ejecución cuando la *Subscreen* del programa principal llame a la dynpro.

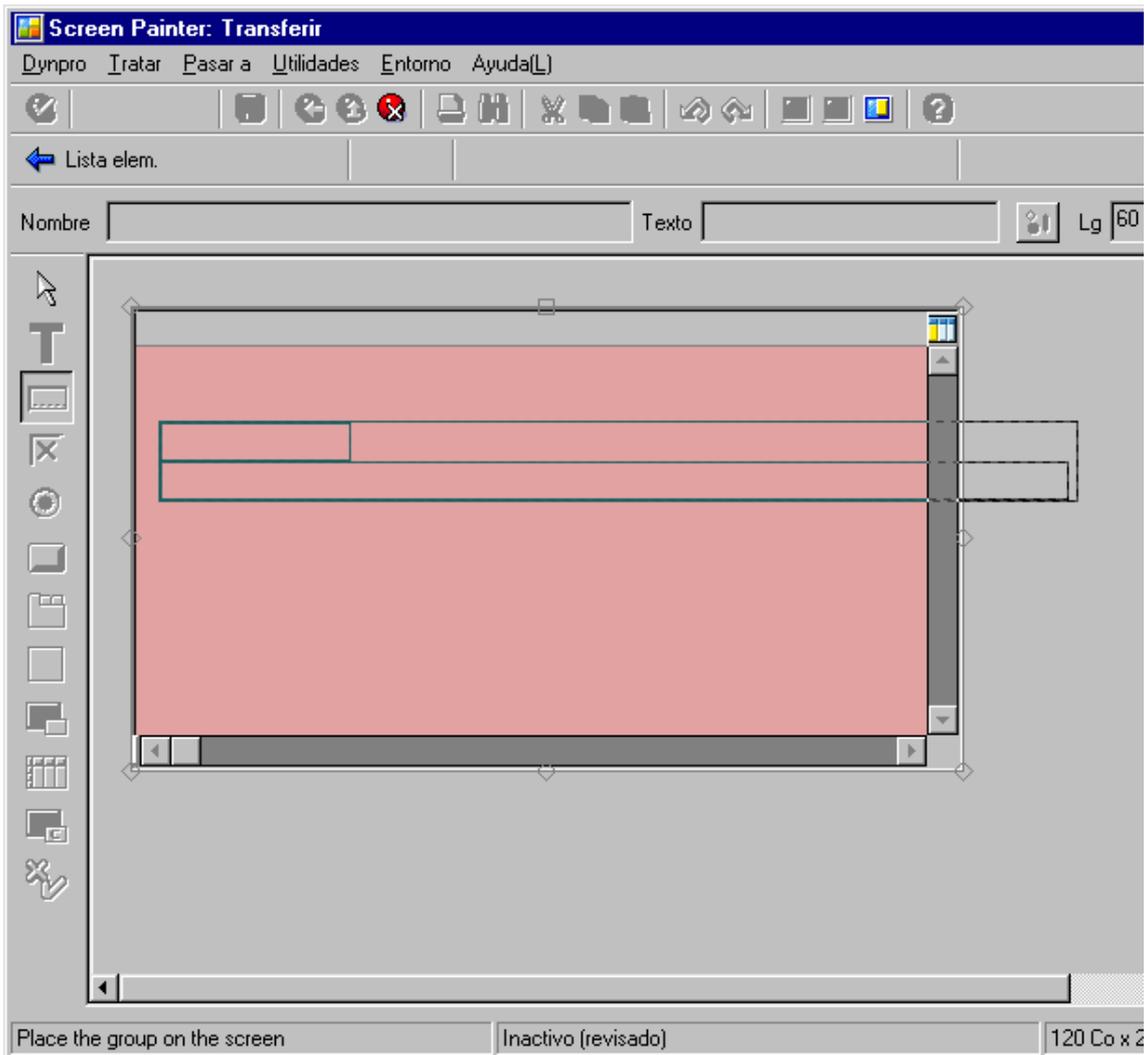
Antes de nada es ponerle el nombre al *Table control*, recordar que ha de ser el mismo nombre que el declarado en el programa marco. En nuestro caso el nombre es *Table*.

Los atributos específicos de este control son los siguientes:

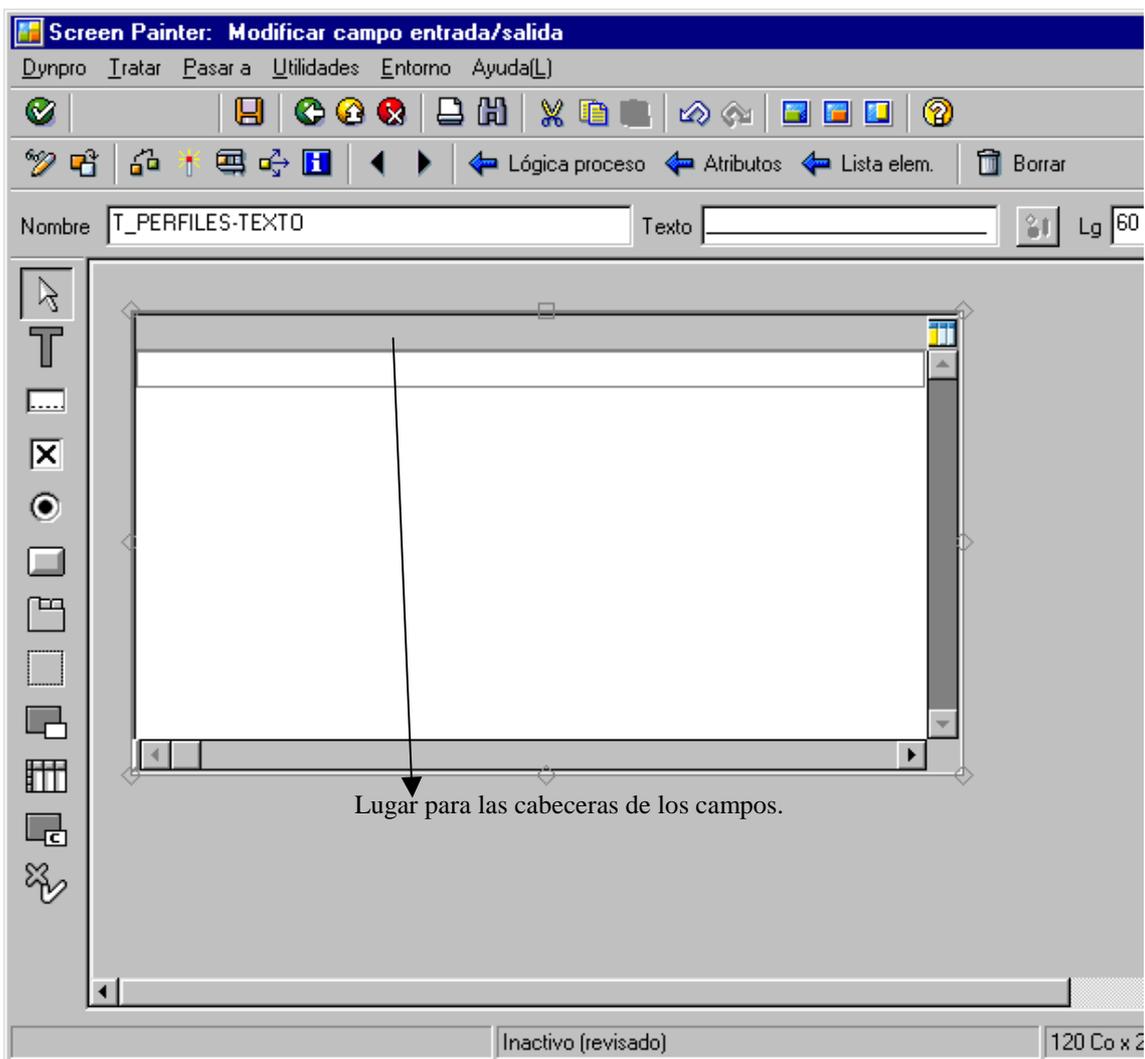
- C/cabcol → Si lo activamos nos permite poner cabecera a las columnas que insertemos.
- Config → Si esta activo nos saldrá un icono en la parte superior izquierda donde podremos modificar el aspecto del *Table control* en tiempo de ejecución.
- C/título → Si lo activamos podremos poner un título al *Table control*. Para poner el título SAP nos dirá que tenemos que poner un control de tipo *Campo de texto* en la zona que SAP nos reserva para el título del *Table control*.
- Resizing, vertical → Activo podremos modificar la separación vertical del *Table control*
- Resizing, horizontal → Lo mismo que antes pero para la horizontal.
- Líneas sep, vertical → Activado veremos las líneas de separación verticales del *Table control*.
- Líneas sep, horizontal → Activado veremos las líneas de separación horizontales del *Table control*.
- Marc. Línea, Sin → Activo no podremos seleccionar ninguna línea vertical del *Table control*.
- Marc. Línea, simple → Activo solo podremos seleccionar una línea del *Table control*.
- Marc. Línea, múltiple → Activo podremos seleccionar una o varias líneas del *Table control*.
- Marc. Columna, Sin → Activo no podremos seleccionar ninguna columna vertical del *Table control*.
- Marc. Columna, simple → Activo solo podremos seleccionar una columna del *Table control*.
- Marc. Columna, múltiple → Activo podremos seleccionar una o varias columna del *Table control*.
- C. Marc. Col → Si lo activamos saldrá en la parte izquierda del *Table control* un recuadro donde se utilizara para seleccionar las líneas (dependiendo del valor de *Marc. Línea*) del *Table Control*. Si lo activamos el cuadro de texto que tenemos a la derecha se activara y tendremos que poner el nombre de una variable (ha de ser un *char* de una posición) que será la encargada de controlar que líneas están activas y cuales no lo están. Y por supuesto la variable que pongamos ha de estar declarada en el programa marco o en la include del programa marco donde tengamos las variables.

Después de esta pequeña explicación de los atributos vamos a colocar los campos en el *Table control*, tarea muy sencilla como veremos.

Los campos los cogeremos de la tabla interna *t_perfiles* (a través del *Campo Dict./Prog.*), recordar que tenemos que activar el programa para que nos salga nuestra tabla interna en el *Campo Dict./Prog.* Cuando los tomemos veremos que con el cursor podemos mover los campos seleccionados, lo que haremos será moverlo hasta dentro del *Table Control* como en la siguiente imagen:



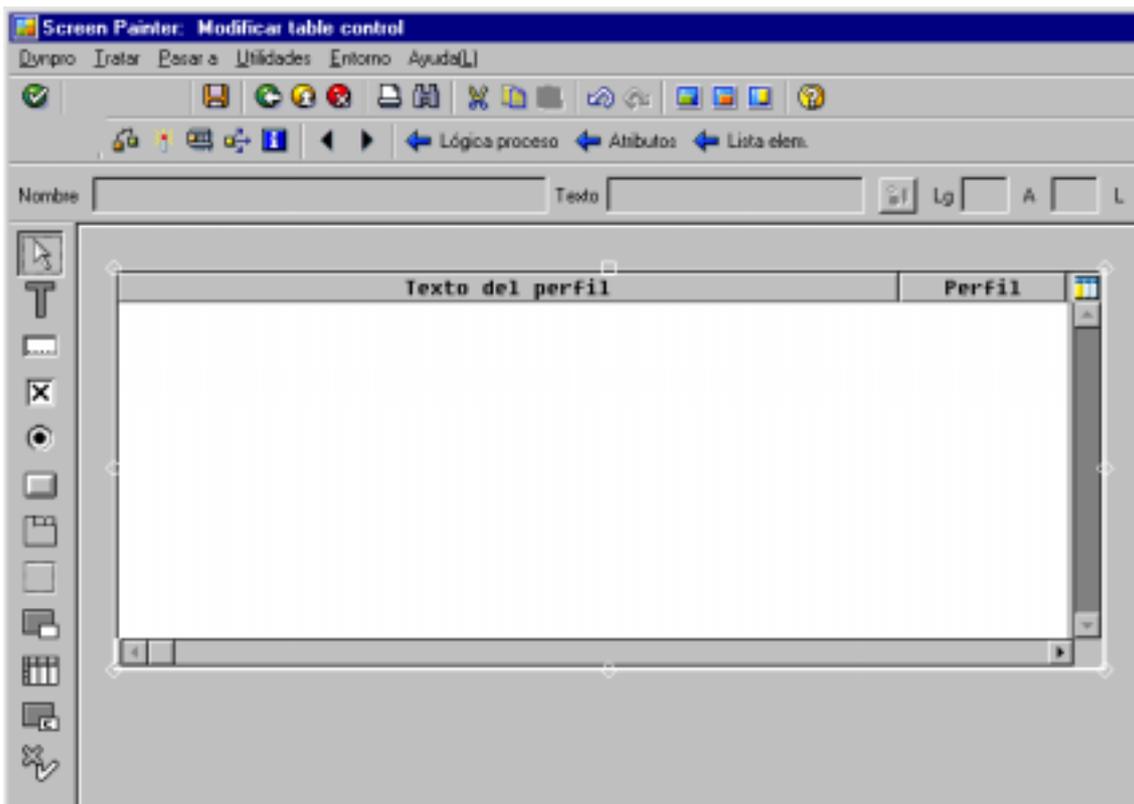
Una vez lo tengamos en la posición haremos un click con el ratón y automáticamente se nos colocara en el *Table Control*, quedaría de esta forma:



A simple vista no vemos que campo es el del perfil ni que campo es la descripción. Ahora pondremos las cabeceras de los campos.

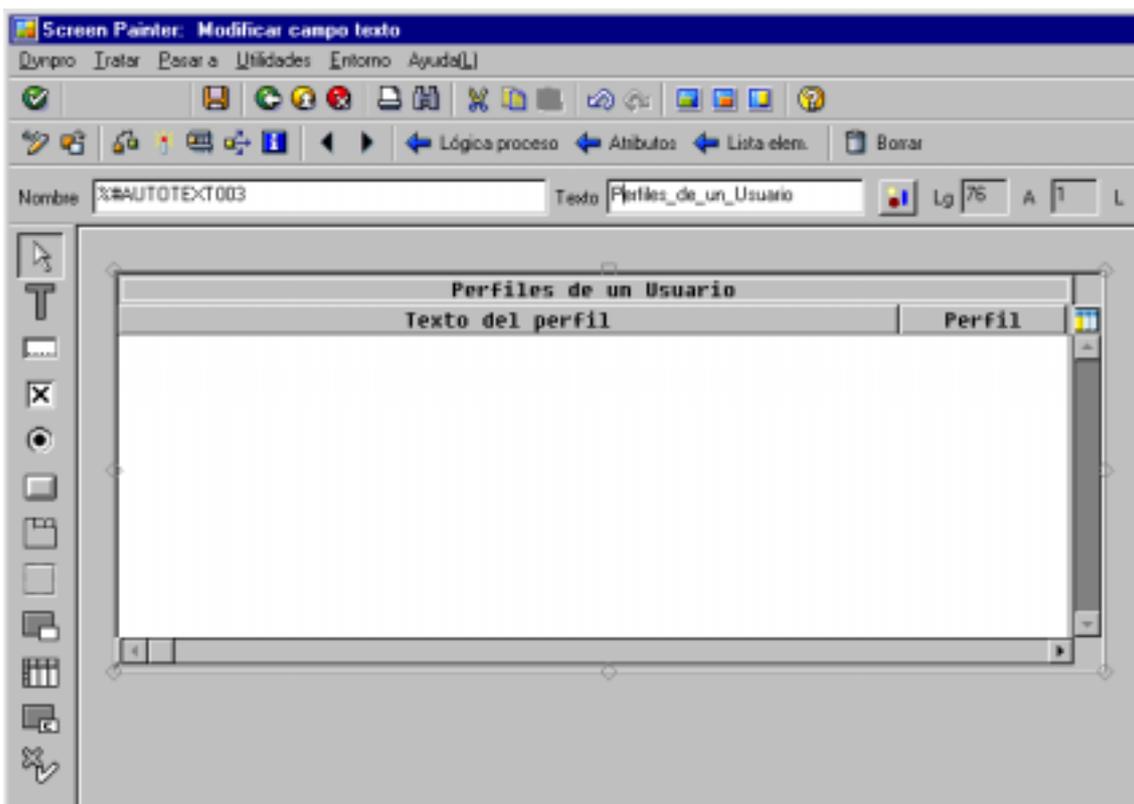
Nota: Si insertamos campos de un tabla de diccionario automáticamente SAP nos pone la cabecera del campo, esto lo hace siempre y cuando el campo "C/cabcol" este activado.

Para insertar la cabecera tenemos que coger un control de tipo *Campo de texto* e insertarlo en el área de cabeceras, veremos que dicho control se expande igualando la longitud del campo que tenga debajo. La pantalla tiene que quedar de esta forma:



El tratamiento del *Campo de Texto* veremos que es el mismo que si estuviera fuera del *Table Control*. A los dos controles le pondremos una descripción, el nombre lo pone SAP de forma automática. Como podréis observar lo correcta sería que saliese primero el perfil y después la descripción. En este caso no sale porque cuando insertamos un grupo de campos (de tabla interna) y nos lo coloca ordenado por campo. Para hacerlo bien habría que colocarlo de uno en uno.

Ya solo nos falta ponerle un título al *Table control* para ello iremos a los atributos del *Table Control* y activaremos el *Checkbox C/título*, veremos que nos aparece un espacio encima de las cabeceras de columna y es en ese lugar donde pondremos otro *Campo de Texto* con su descripción, la pantalla quedaría así:



Nota: Las cabeceras de columna y el título de un “Table Control” las podemos modificar en tiempo de ejecución. Para ello tenemos que declarar una variable con el mismo nombre del control “Campo de texto” y hacerle una asignación en cualquier parte del programa.

Una vez realizado el diseño solo nos faltaría poner el código en la *Lógica de proceso* para el control del *Table Control*. El código sería el siguiente:

```

process before output.
* MODULE STATUS_9200.
* Recorro la tabla donde estan los perfiles y su texto para que sean
* insertados en el table control.
  loop at t_perfiles with control table cursor i.

  endloop.

process after input.
  loop at t_perfiles.
  endloop.

* MODULE USER_COMMAND_9200.

```

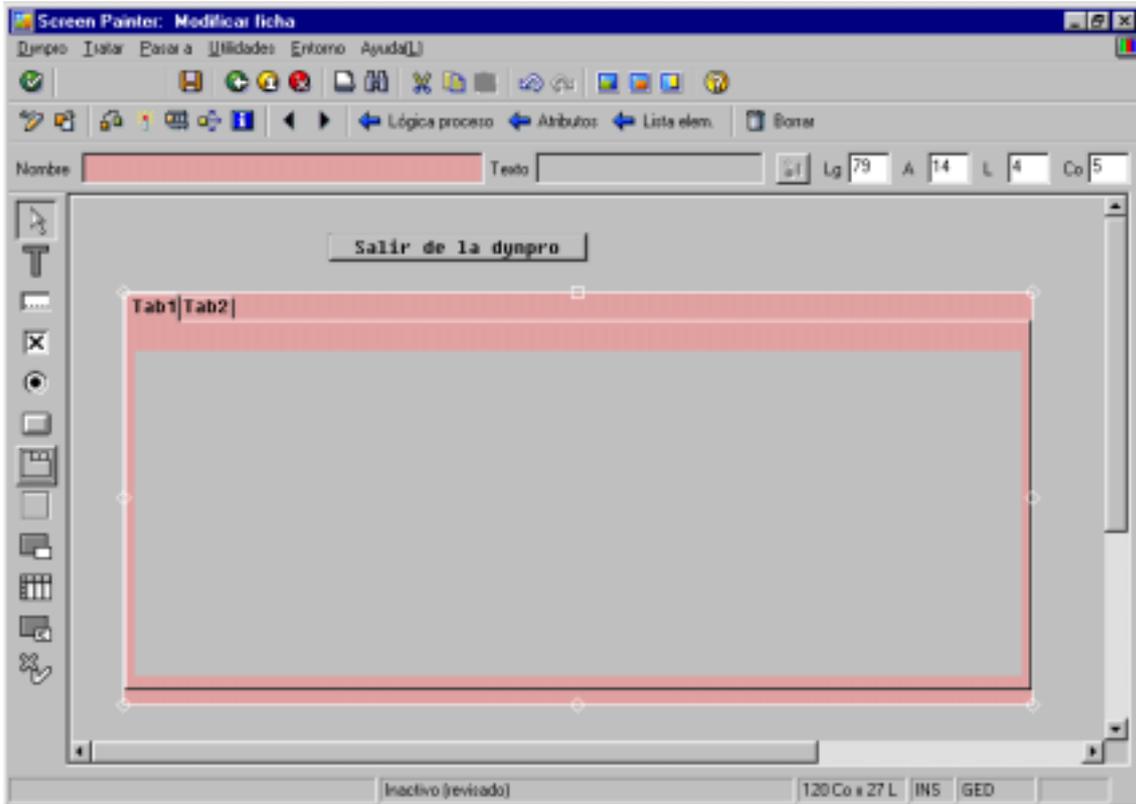
Como vemos el código es muy sencillo. El primer *loop* (el que esta en el PBO) es para pasar los datos de la tabla interna al *Table control* (igual que el *Step loop*).

Si os acordáis del ejemplo anterior dentro del *Loop* había un *module* que hacía la asignación de datos al *Step loop*. En nuestro caso como el *Table control* esta formado por los mismo campos de la tabla interna *t_perfiles*, SAP hace la asignación automáticamente y no hay que hacer ninguna asignación.

El segundo *Loop* es el que controla que campos del *Table control* se han modificado, que en nuestro caso no haremos ningún tipo de control. En esta *Lógica de proceso* no controlaremos que tecla se pulsa.

Por último nos tocara crear la dynpro principal, o sea, la que tiene el *Tabstrip*. Esta dynpro tendrá el número 9000 y el atributo de pantalla será el de *Ventana Normal*.

Primero pondremos el botón de salir de la dynpro que tendrá el código de función *Salir*. Después colocaremos el *Tabstrip* y la pantalla quedará de esta forma:



Lo primero es ponerle un nombre al *Tabstrip*, que será *Tab1*. En la pantalla de *Atributos* el campo nuevo que tenemos es donde pone *Lengüeta*, este campo indica cuantas pestañas tiene el *Tabstrip*. Este valor lo podemos modificar y poner las pestañas que queramos

Después de ponerle el nombre y haberle echado una ojeada a los atributos es la hora de tocar las pestañas. Para ello haremos un click donde pone *Tab1* y veremos que aparece un recuadro como este:



Una vez seleccionado pulsaremos sobre el botón de *Atributos* o F2 y veremos los atributos de la pestaña:

Tipo elem. Pulsador

Nombre [Redacted]

Texto Tab1

Nom.icono [Redacted]

Quick info [Redacted]

Línea 1 Long.def. 4

Columna 1 Long.vis. 4

Altura 1

Grupos [Redacted] [Redacted] [Redacted] [Redacted]

Cód.func. [Redacted] Tp.fun [Redacted]

Formato menú cxt.

ON_CTMENU_ [Redacted]

Atributos

Dict Programa Visual

Formato CHAR

Del Dict Modific. [Redacted]

Exit convers. [Redacted]

Ay.búsq. [Redacted]

Campo ref. [Redacted]

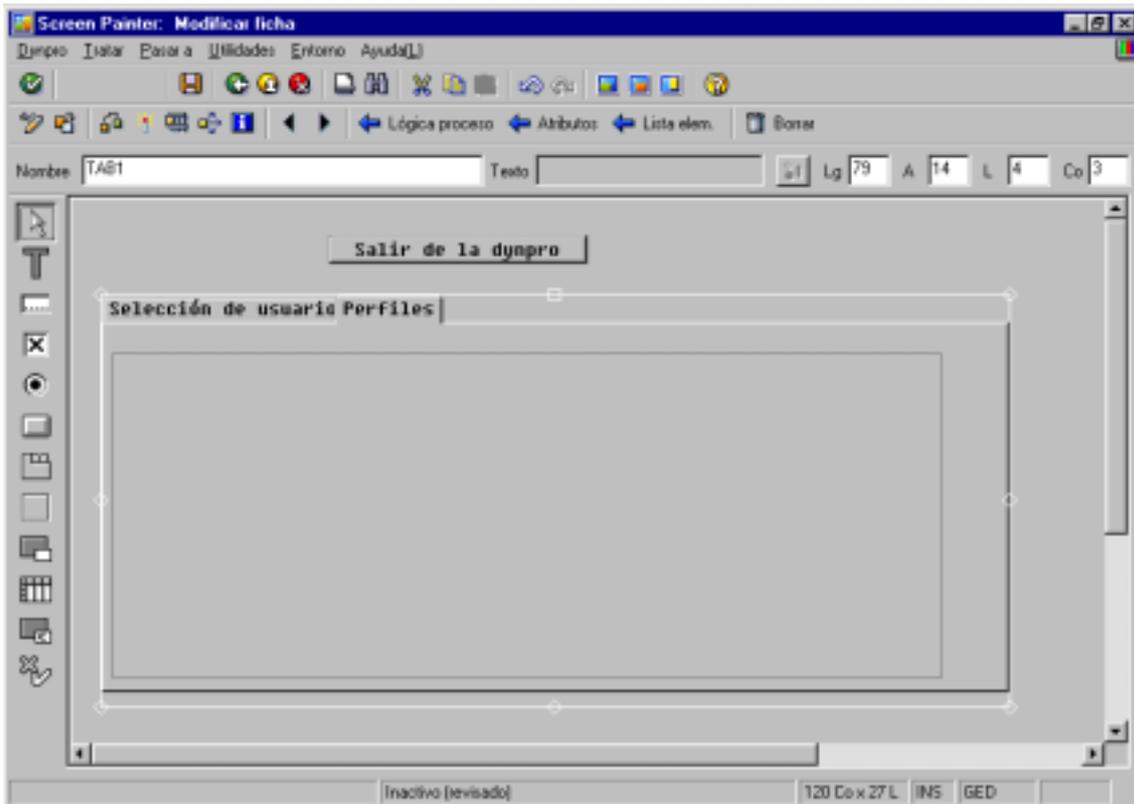
ID parámetro [Redacted]

Los campos que aparecen en ese color rojizo son obligatorios. Como vemos los atributos son los de siempre y ya han sido explicados con anterioridad. Pero veamos que tendremos que poner en los campos obligatorios:

- Nombre → Es el nombre de la pestaña y podemos ponerle el nombre que queramos. Este campo no es obligatorio aunque su color le indique, si no ponemos nada SAP le pone un nombre.
- Texto → Es donde ponemos la descripción a la pestaña.
- Cód. Func. → Es donde pondremos el código que más tarde utilizaremos para el control de la pestaña. En la primera que será la de usuarios le pondremos el código: *USU*. La segunda pestaña que será la de perfiles tendrá el código: *PERF*.
- Campo ref. → En este lugar es donde pondremos el nombre de la *Subscreen*. Para la pestaña de usuarios la *Subscreen* será: *Subscreen_usuario* y en la pestaña de perfiles le pondremos: *Subscreen_perfiles*

Una vez rellenados los campos obligatorios y la descripción veremos que en la pestaña sigue apareciendo ese color rojizo, esto se debe a que no hemos puesto la *Subscreen*. Colocar una *Subscreen* en un *Tabstrip* es muy sencillo. Solo tenemos que seleccionar el control *Subscreen*, con el botón izquierdo del ratón seleccionar en que zona queremos que este el *Subscreen* y cuando soltemos el botón izquierdo SAP cogerá automáticamente el nombre de la *Subscreen*. ¿Cómo coge el nombre? Muy sencillo, SAP lo coge del campo *Campo ref.* de los atributos de la pestaña donde estamos colocando la *Subscreen*, como he explicado antes.

La pantalla nos quedaría de esta forma:



Como vemos el aspecto de las *Subscreen* es parecido al de los marcos y en sus propiedades tendremos casi todas las propiedades que en el *Tabstrip* (exceptuando el campo de *Lengüeta* que no la tiene).

Por último tenemos que teclear el código para la gestión del *Tabstrip*. La lógica de proceso sería la siguiente:

process before output.

```
* MODULE STATUS_9000.
*
* Llamamos a las dynpros 9100 y 9200 del programa actual (sy-repid)
* y se visualizaran en la subscreen correspondientes.
* En esta parte se llamaran a las PBO de las dynpro 9100 y 9200
  call subscreen: subscreen_usuario including sy-repid '9100',
                 subscreen_perfiles including sy-repid '9200'.
```

process after input.

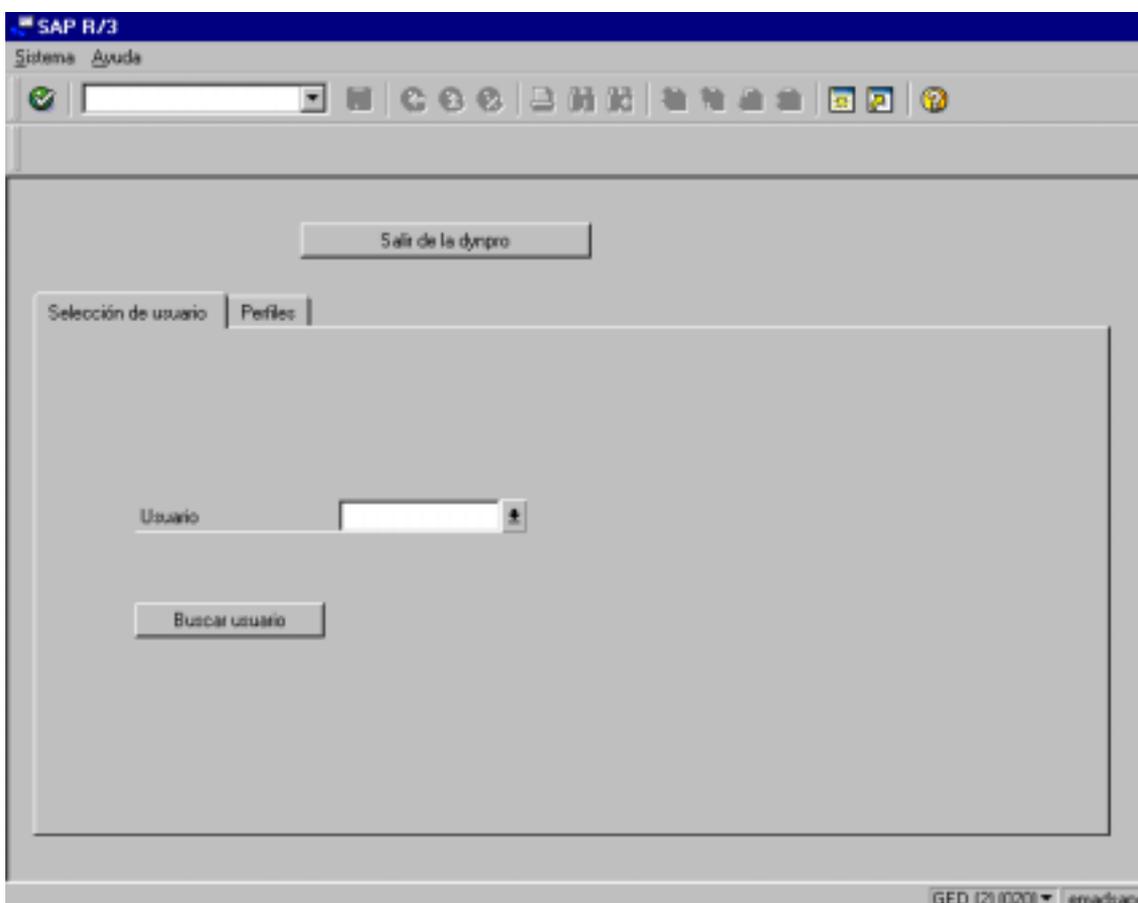
```
* En esta parte se ejecutaran las PAI de las dynpros indicadas
* anteriormente.
  call subscreen: subscreen_usuario,
                 subscreen_perfiles.
* Control del teclado de la dynpro principal.
  module user_command_9000.
```

Y el módulo *User_command_9000* tendría el siguiente código:

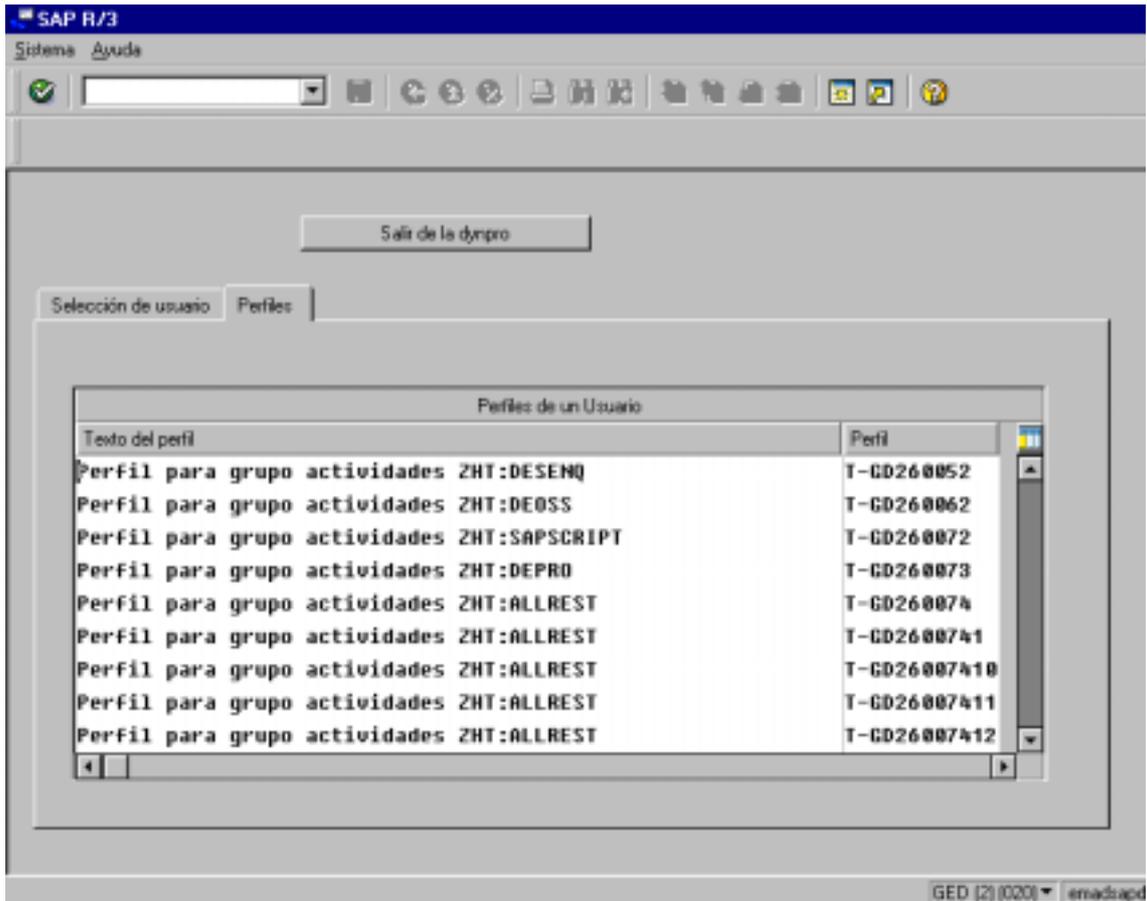
```

module user_command_9000 input.
  case sy-ucomm.
    when 'SALIR'.
* Salgo de la dynpro.
      set screen 0. leave screen.
* Cuando se le de a la pestaña de Usuarios tenemos que activarla
* a mano porque SAP no lo hace
      when 'USU'.
        tab1-activetab = 'USU'.
* Con la pestaña de perfiles lo mismo.
      when 'PERF'.
        tab1-activetab = 'PERF'.
      endcase.
  endmodule.
" USER_COMMAND_9000 INPUT
  
```

Una vez tengamos todo echo tenemos que activar y genera tanto el programa marco como las dynpros. Cuando ejecutemos la transacción que tenemos asociada al *modulpool* nos saldrá la siguiente pantalla:

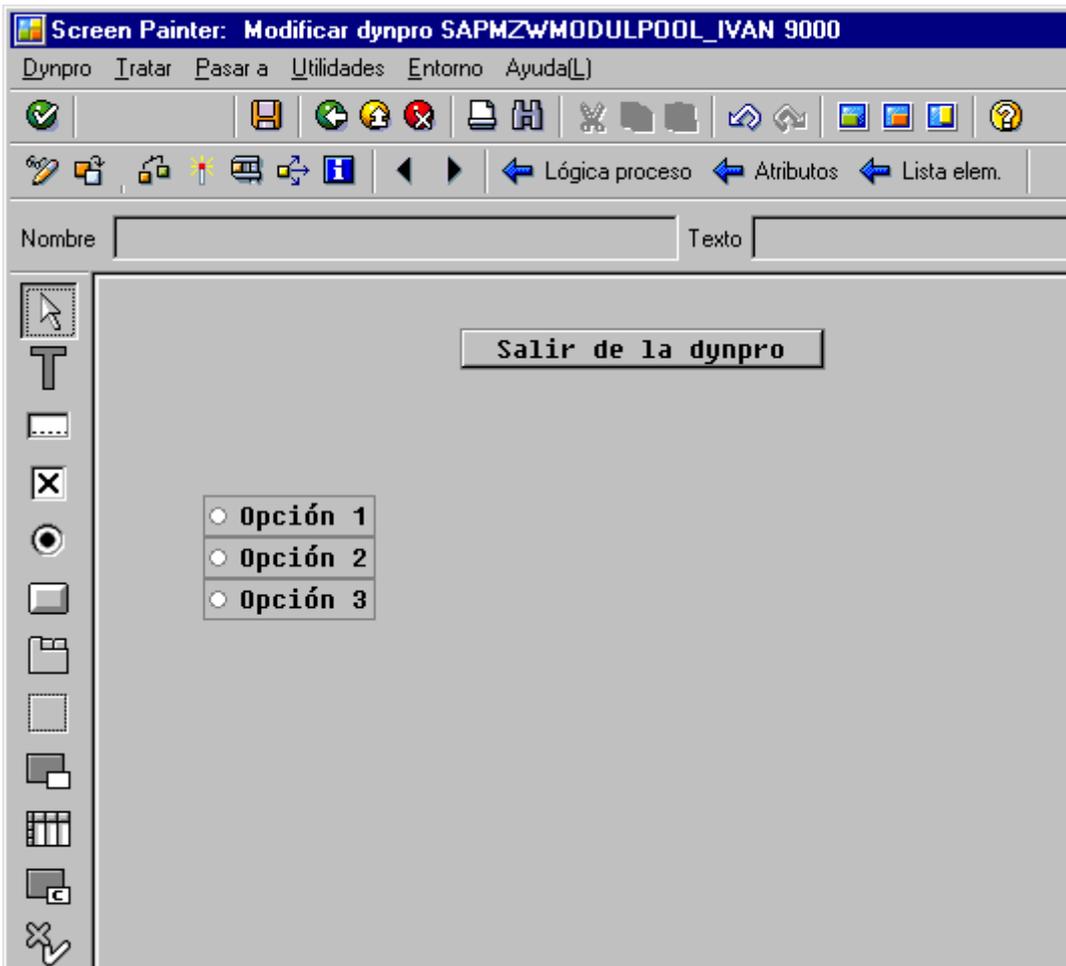


Cuando pongamos un usuario y le demos al botón de buscar nos saldrá lo siguiente:



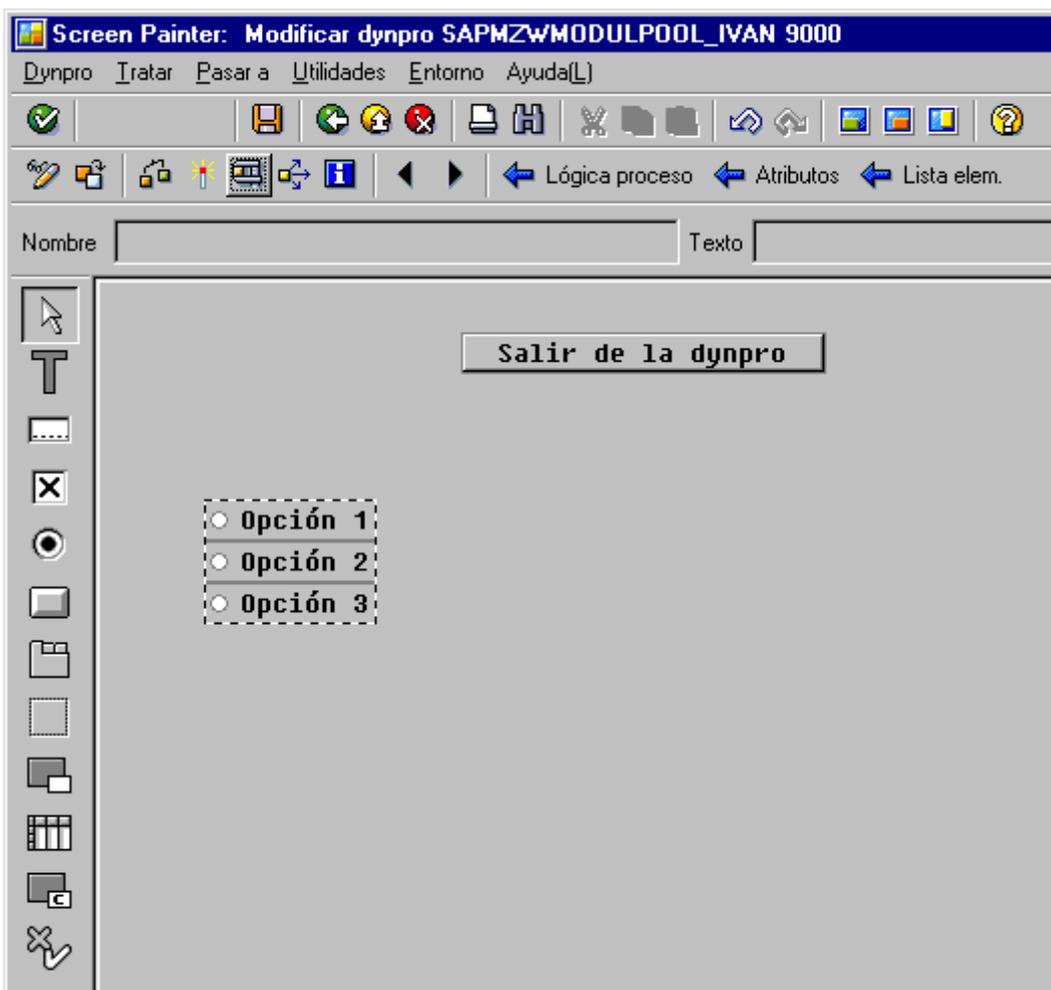
Radiobutton:

Para los *Radiobuttons* no voy hacer un ejemplo sino explicar como funcionan mediante unos pantallazos. Podemos ver un ejemplo de estos en la *Screen painter* (transacción SE51), fijaros en el *marco* que pone *Objetos parciales*, veréis que solo se puede seleccionar uno de los cuatro *Radiobutton* disponibles. Para ver como funcionan crearemos una dynpro, o como yo he hecho, que he aprovechado la dynpro 9000. Lo que haremos es colocar tres *Radiobuttons*, como en esta pantalla:



A los *Radiobuttons* hay que ponerles un nombre para saber quien es quien. Y para controlar los valores de los *Radiobutton* en tiempo de ejecución tendremos que declarar una variable del mismo nombre y que sea de tipo *char* de una posición. Si hacemos un *test* de la dynpro veremos que los tres *Radiobuttons* están activos y no podremos deseleccionar ninguno de los tres. Para hacer que los tres funcionen a la vez, tenemos que agruparlos.

Para agruparlos tenemos que seleccionar los tres *Radiobuttons*, para ello seleccionaremos los *Radiobuttons* e iremos al menú *Tratar* → *Agrupación* → *Gp. Botones sel.* → *Definir* y veremos que los *radiobuttons* quedan de esta forma:



Si hacemos un TEST de la dynpro veremos que si seleccionamos uno se deselecciona el que esta seleccionado.

Podemos hacer tantas agrupaciones como queramos y las agrupaciones pueden ser de tantos *radiobuttons* como queramos.

Si queremos desagrupar tenemos que ir por el menú *Tratar* → *Agrupación* → *Gp. Botones sel.* → *Definir*

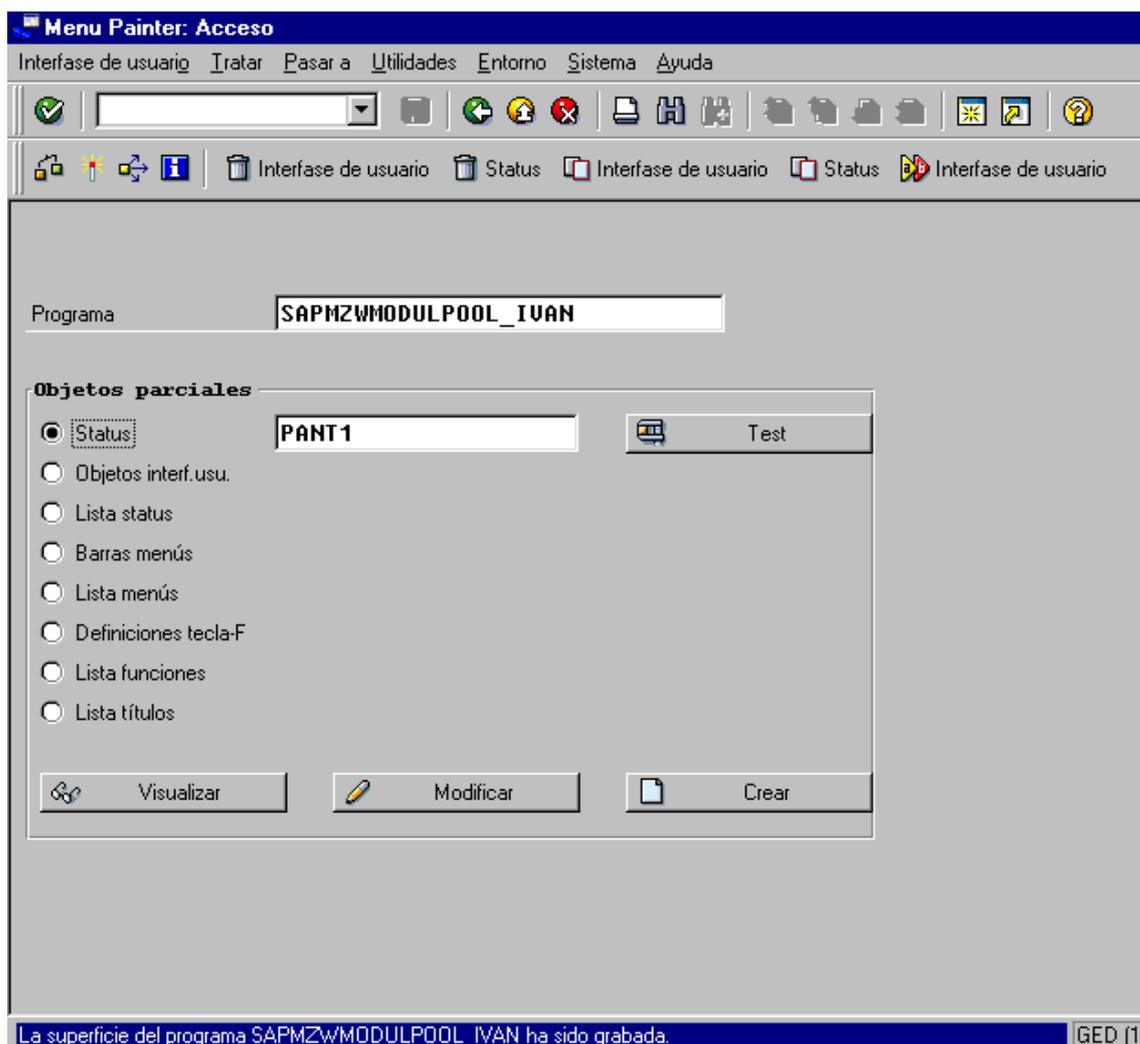
Con estos ejemplos sabremos utilizar de forma básica cualquier control de SAP.

MENU PAINTER Y BARRA DE TÍTULO

Un *Menu painter* sería los menús y botones que aparecen en la parte superior de una dynpro. La *Barra de título* es el título que le ponemos a la dynpro.

En un *Modul pool* el *Menu painter* lo llamamos desde el PBO y a través de la sentencia *SET PF-STATUS*. Para crear un *Menu painter* se puede hacer de dos formas diferentes que son:

La primera es través del *Menu painter* que es la transacción SE41, desde el *SAP Easy Acces* sería: *Menú SAP* → *Herramientas* → *Workbench ABAP* → *Interfase del usuario* → *SE51* → *Screen Painter*. Cuya pantalla es la siguiente:



Esta es la pantalla principal y el significado de los diferentes campos es el siguiente:

- Programa → Nombre del programa donde vamos a crear el *status* o *Menu painter*.
- Status → Nombre del *Menu painter* que vamos a crear, modificar, visualizar o borrar.
- Objetos interf usu. → Esta opción nos permite navegar por los diferentes componentes del *Menu painter*.
- Lista status → Muestra los *status* que tiene el programa.
- Barra de menús → Muestra las barras de menús de las diferentes dynpros que tengamos.
- Lista de menús → Muestra las listas de menús de las diferentes dynpros que tengamos.
- Definiciones teclas-F → Muestra los nombres de los códigos que se utilizan en la sentencia *SET PF-STATUS*.
- Lista funciones → Muestra los códigos de funciones que se utilizan en el programa.
- Lista títulos → Muestra las barra de títulos que están definidas en el programa. Estas barras son llamadas con la sentencia *SET TITLEBAR*. Como crear las barras de títulos se verán más adelante.

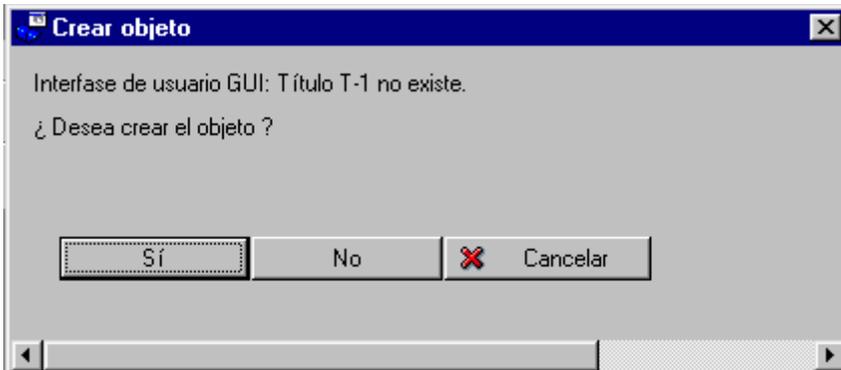
Esta es una forma manera de hacerla. La otra forma es que la siempre utilizo yo, y es crearla desde el propio programa cuando ponemos la sentencia *SET PF-STATUS*.

Cojamos una dynpro cualquier que hayamos utilizado para hacer los ejemplos o hagamos una nueva. En la lógica de proceso desasterisquemos el módulo *MODULE STATUS_9000* (este sería mi módulo en mi caso) y le daremos doble-clic para crearlo, yo recomiendo para este caso que lo creéis en vuestro propio programa marco o de control.

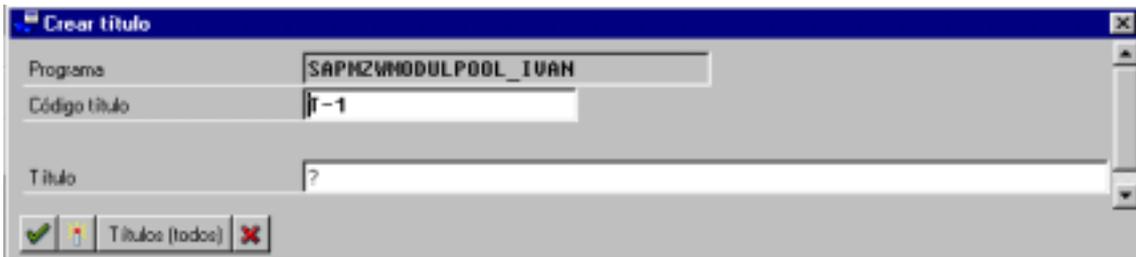
Una vez creado veréis dos instrucciones asteriscadas que son: *SET PF-STATUS 'xxxxxxx'* y *SET TITLEBAR 'xxx'*. Primero desasteriscaremos la segunda instrucción (la de la barra de título) y sustituiremos las tres 'x' por 'T-1', es decir, como en esta imagen:

```
* SET PF-STATUS 'xxxxxxx'.  
SET TITLEBAR 'T-1'.
```

Os aviso que las letras que van en el espacio de las 'x' han de estar siempre en mayúsculas, tanto para la sentencia *SET TITLEBAR* como para *SET PF-STATUS*. Cuando le demos *doble click* donde pone 'T-1' saldrá una ventana que nos saldrá lo siguiente:



Le diremos que sí y nos saldrá una ventana para que grabemos el programa marco (sale, sino hemos grabado el programa marco). Por supuesto que le diremos que si y no saldrá la siguiente pantalla:

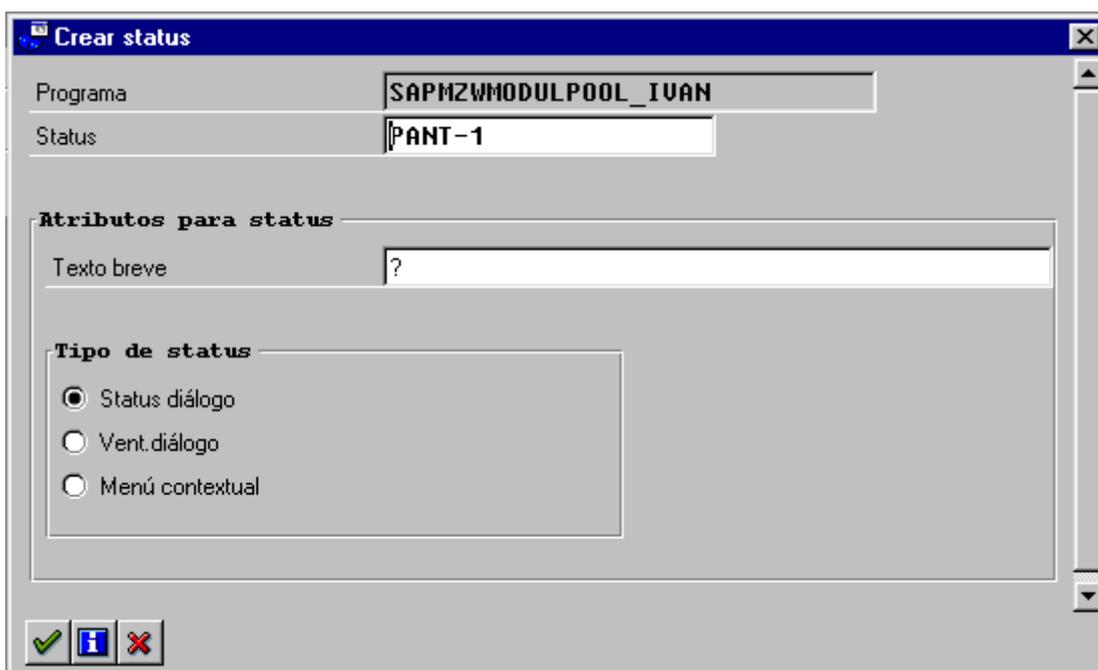


Los campo que salen en la pantalla son los siguientes:

- Código título → Es título o nombre de la barra de título. Es recomendable no tocarlo y dejar el que hemos puesto en la sentencia.
- Título → Es el título que saldrá en la dynpro. Como por ejemplo el que pone "Crear título" en la imagen anterior.

Cuando hayamos escrito el título tenemos que activar el título, en el botón de la cerilla. Una vez activado ya esta listo para usarlo. Es importante activar el título cada vez que lo modifiquemos.

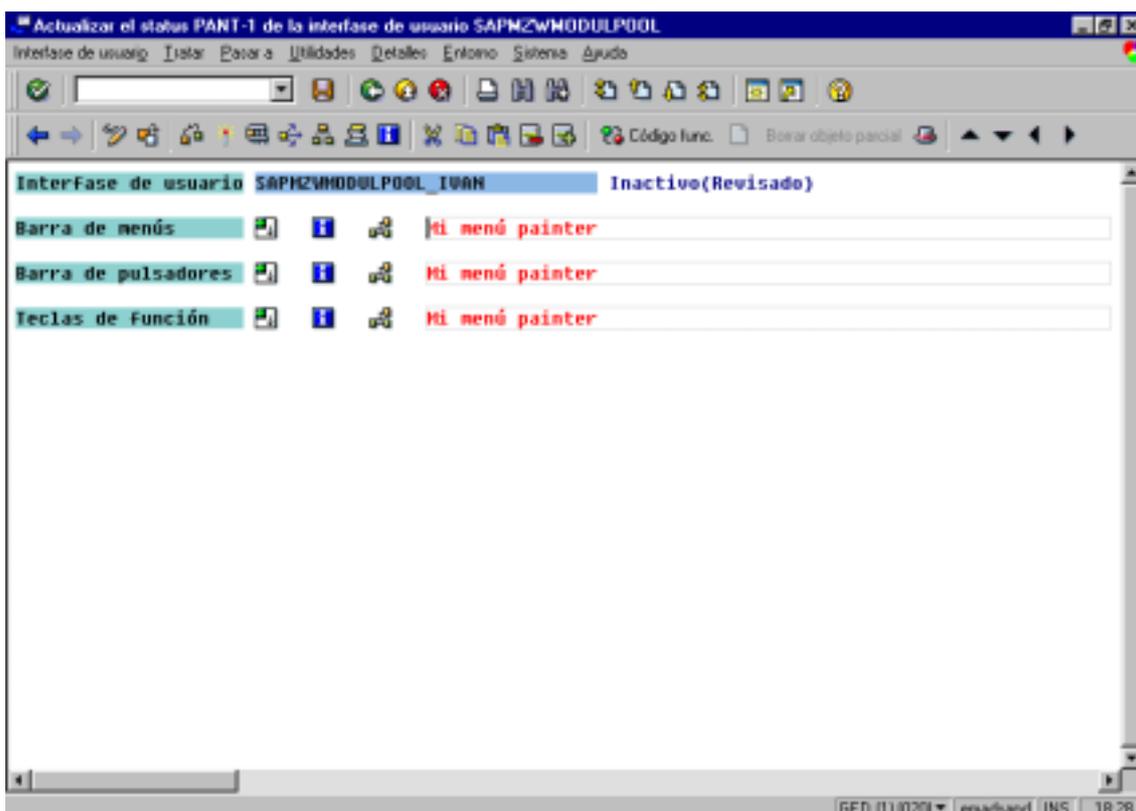
Lo siguiente es crear el menú painter en sí. Para ello en la sentencia *SET PF-STATUS* sustituiremos las 'x' por 'PANT-1'. Después le daremos doble clic a *PANT-1* y os dirá que el *status* no existe y si lo queremos crear. Le diremos que SI, después si no hemos grabado el programa o la *include* donde estábamos nos preguntara si la queremos crear y también le diremos que SÍ. Una grabado el programa marco no saldrá la siguiente ventana:



En esta ventana definiremos los atributos del *Menu painter*, las opciones que tenemos son:

- Status → El nombre del *status* o *Menu painter*. Es recomendable no tocarlo y dejar el que hemos puesto en la sentencia.
- Título → El título que le daremos al *status* o *Menu Painter*.
- Tipo de status -> Como va a salir el *Menu Painter* o *Status* en pantalla. Tenemos tres tipos de *status* que son:
 1. Status diálogo → Es el *Status* que se utiliza para *Report* y *modulpool*. Es decir dynpros a pantalla completa.
 2. Ventana de diálogo → Son aquellos que se muestran en ventanas pequeñas, en este tipo de *status* no se puede poner menús. Un ejemplo de este tipo de ventana es la que os he puesto para poner los atributos del *Menu painter*.
 3. Menú contextual → Es el típico de Windows que aparece cuando le damos con el botón derecho del ratón.

Una vez puesto la descripción que queramos, para el ejemplo dejar el tipo de *status* que pone SAP. Le daremos al botón de la flecha verde o pulsar ENTER y nos saldrá la siguiente ventana:



Desde esta ventana podremos definir los menús, teclas de función y los botones típicos de SAP. Personalmente me gustaba más el *Menu Painter* de la versión 3.1H de SAP. Las tres partes que tenemos son las siguientes:

- Barra de menús → Es donde se crean los menús de la *dynpro*.
- Barra de pulsadores → Es donde ponemos los botones que se crean en *Teclas de función*.
- Teclas de función → Lugar donde crearemos nuestros botones para luego ponerlos en la *Barra de pulsadores* y es también donde podremos los códigos de función a la barra de herramientas estándar de SAP.

Los iconos comunes en las tres barras son los siguientes:

 → Sirve para expandir los datos de las diferentes áreas. Cuando pulsemos el botón aparecerá este otro  que será el que contraiga los datos de las áreas.

 → Con este icono nos se visualizará una ventana donde se muestra jerárquicamente como está formado el área donde hayamos pulsado el botón.

 → Con este botón permite hacer asignaciones de área de diferentes *status* a nuestra área.

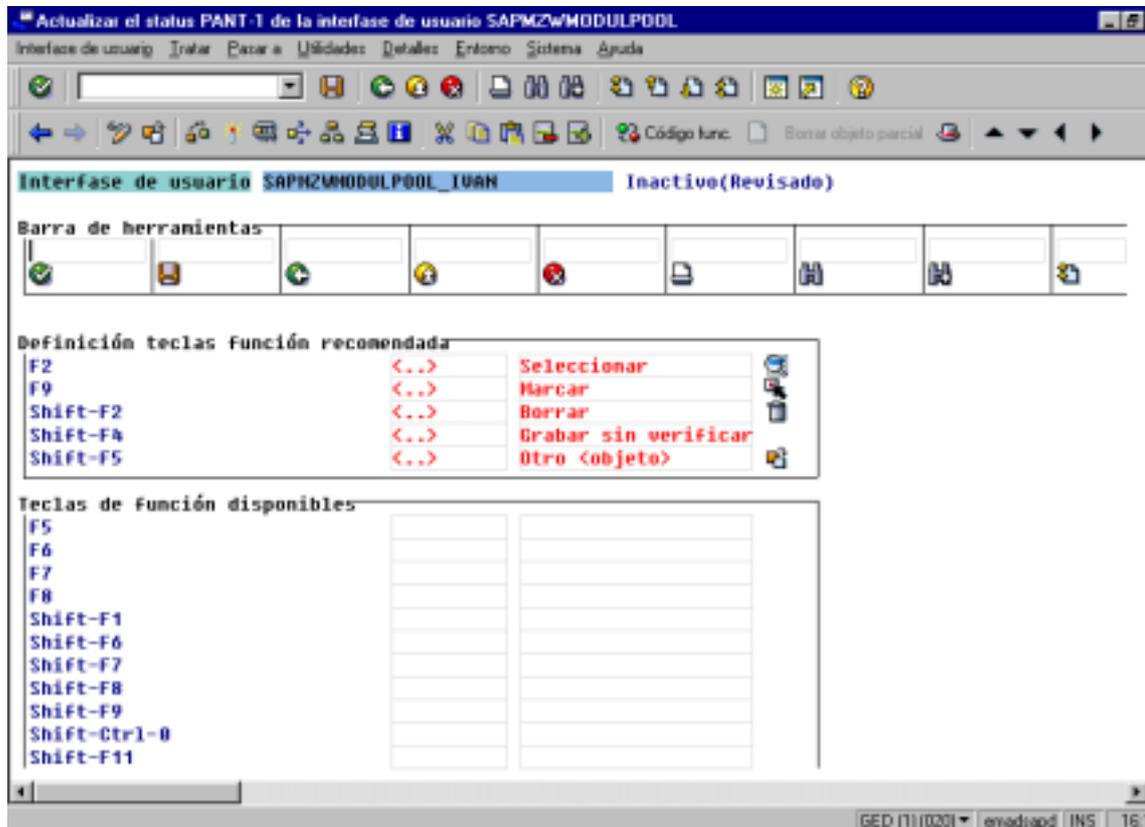
 → En este recuadro podremos cambiar la descripción del área.

Personalmente la única opción que utilizo es la de expandir o contraer las áreas.

Primero veremos como se crea la *Barra de menú*, segundo las *Teclas de función* y por último la *Barra de pulsadores*.

TECLAS DE FUNCIÓN

En esta área es donde crearemos nuestros propios botones que se colocaran en la barra de pulsadores y donde pondremos los códigos de función a los botones estándar de la barra de pulsadores.



Como vemos en la imagen hay tres zonas que son:

- Barra de herramientas → Es donde se ponen los códigos de función para los botones estándar de SAP. Si queremos ver que códigos de función tiene SAP para dichos botones tenemos que ir al menú *Pasar A* → *Listas de objetos* → *Lista de funciones*.
- Definición de teclas de función recomendada → Son teclas que SAP recomienda cuando vayamos a realizar alguna tarea similar al significado del icono de la tecla. Por ejemplo. Si en nuestro *Modul pool* o *Report* tenemos que realizar una acción de borrar, es bueno que utilicemos la tecla *Shift+F2*. Fijaros que cuando queremos borrar algo estándar de SAP siempre pulsamos *Shift+F2* o el icono de la papelera. SAP, eso sí, permite ponerle el código de función que más nos convenga a la tecla que queramos utilizar.
- Teclas de función disponibles → Es lugar donde crearemos nuestras propias funciones, con su respectivo icono, para después ponerlas en la *Barra de pulsadores*.

TECLAS DE FUNCIÓN ESTANDAR

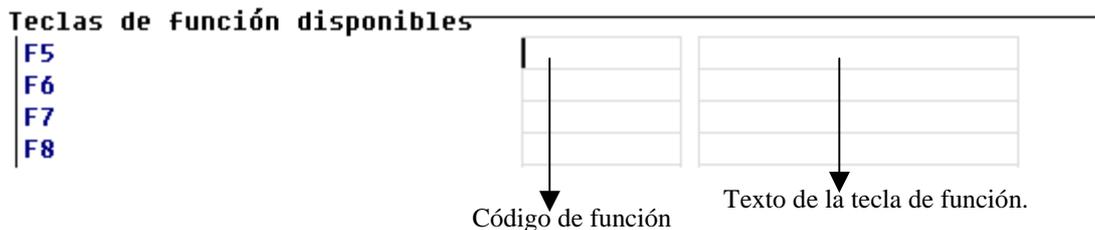
Como hemos visto antes, hay una serie de *Teclas de función* que SAP tiene por defecto y que suelen poner en la *Barra de herramientas*. Un ejemplo de cómo sería una *Barra de herramientas* estándar sería:



Solo tenemos que poner los códigos y pulsar *Enter*.

COMO CREAR UNA TECLA DE FUNCIÓN

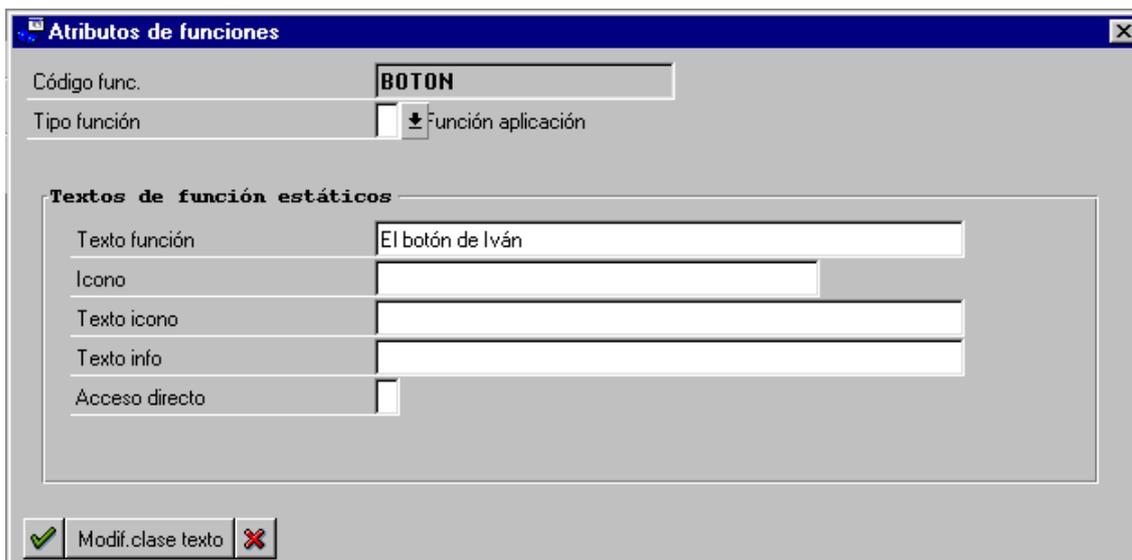
Para crear una *Tecla de función* tenemos que irnos al *marco* de *Teclas de función disponibles*:



Poner un código de función que no exista, después ponerle un nombre y por último pulsar ENTER. El resultado sería el siguiente:



Como vemos es muy fácil hacer una *Tecla de función*, si queremos modificar los atributos de la *Tecla de función* haremos *doble click* sobre donde pone *Botón* (área de código de función) o donde pone *El botón de Iván* (área de texto de la tecla de función) y nos saldrá la siguiente ventana:



Las opciones que tenemos son las siguientes:

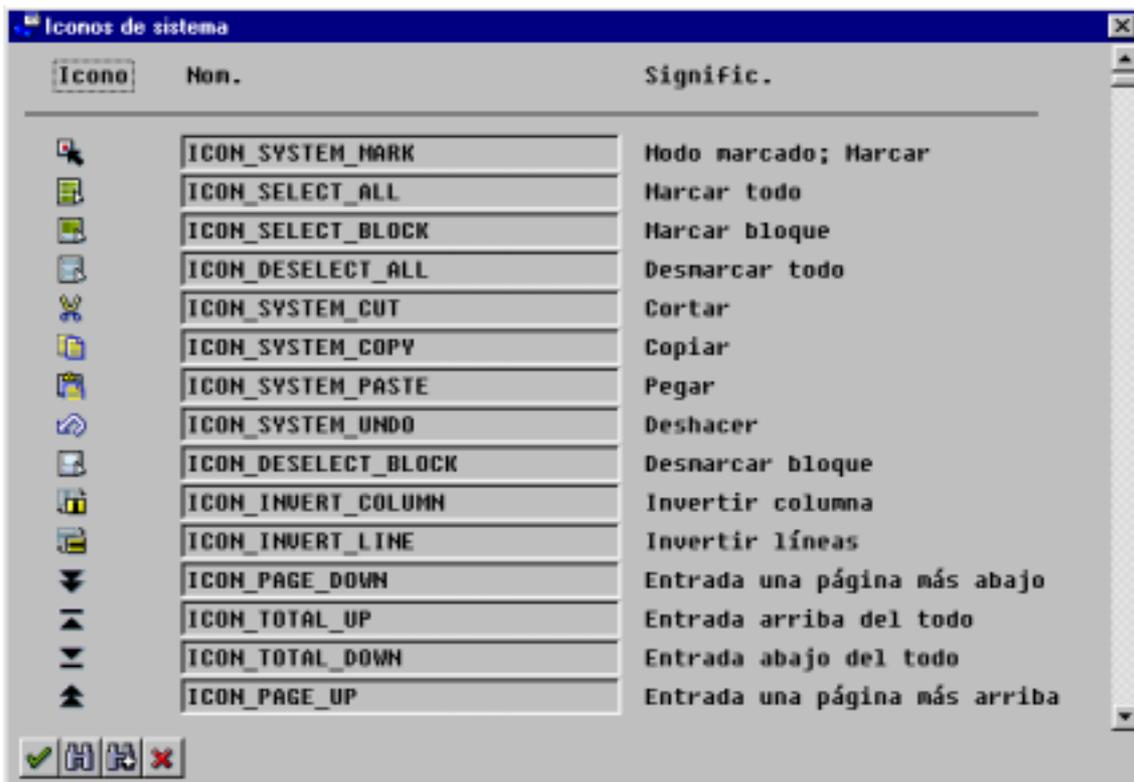
Tipo función → Definimos el tipo de función que queremos para nuestra *Tecla de función*. Dependiendo del tipo escogido la *Tecla de función* tendrá un comportamiento u otro. Los tipos disponibles son los siguientes:



Dependiendo del tipo escogido la *Tecla de función* tendrá un comportamiento u otro.

Texto de función → Es el texto que aparecera en pantalla, a menos, que pongamos un icono.

Icono → En esta campo escogeremos el icono para nuestra *Tecla de función*. Si nos sabemos el icono de memoria lo podemos escribir directamente, pero lo mejor es pulsar sobre el botón *Ayuda para búsqueda* y nos saldría la siguiente ventana:



El icono seleccionado aparecerá al lado de la *Ayuda para búsqueda*:



Texto info → Este texto aparecerá solo cuando pongamos un icono y se visualizara cuando dejemos un momento el cursor del ratón el icono. Por ejemplo si aquí ponemos: “Pegar al portapapeles”, cuando dejemos el cursor el icono que hemos puesto al rato aparece un bocadillo con dicho texto.

Una vez seleccionado nuestro icono la *Tecla de función* quedaría de la siguiente forma:



MANTENIMIENTO DE UNA TECLA DE FUNCIÓN

El menú que nos conducirá al mantenimiento o gestión de las *Teclas de función* es el siguiente: *Pasar A* → *Listas de objetos* → *Lista de funciones*. La pantalla que nos aparecerá es la siguiente:

Funciones programa con textos estáticos		
Código de función	Texto de función	Icono
%EX	Finalizar	
%SC	Buscar	ICON_SEARCH
	Continuar búsqueda	ICON_SEARCH_NEXT
BACK	Back	
BOTON	El botón de Iván	ICON_SYSTEM_PASTE
CANC	Cancelar	ICON_CANCEL
P+	Página siguiente	ICON_NEXT_PAGE
P++	Última página	ICON_LAST_PAGE
P-	Página anterior	ICON_PREVIOUS_PAGE
P	Primera página	ICON_FIRST_PAGE
PRI	Imprimir	ICON_PRINT
SALIR	Salir de la dynpro	

En esta pantalla aparecerán tanto las *Teclas de función* creadas por nosotros como las que SAP pone por defecto para ser utilizadas en la *Barra de herramientas*. También en esta pantalla podremos borrar, modificar, crear, etc... *Teclas de función*.

Os aviso que la única manera de borrar una *Tecla de función* es esta pantalla, ya que aunque la quitemos de la pantalla de *status* (pantalla que sale inicialmente) seguirá esta activa en el *status* hasta que no la borremos desde esta ventana.

BARRA DE PULSADORES

En esta área es donde colocaremos las *Teclas de función* creadas en *Teclas de función*. También podemos poner las *Teclas de función* estándar que hemos visto antes.

El área es la siguiente:

Manual Básico de Modul Pool

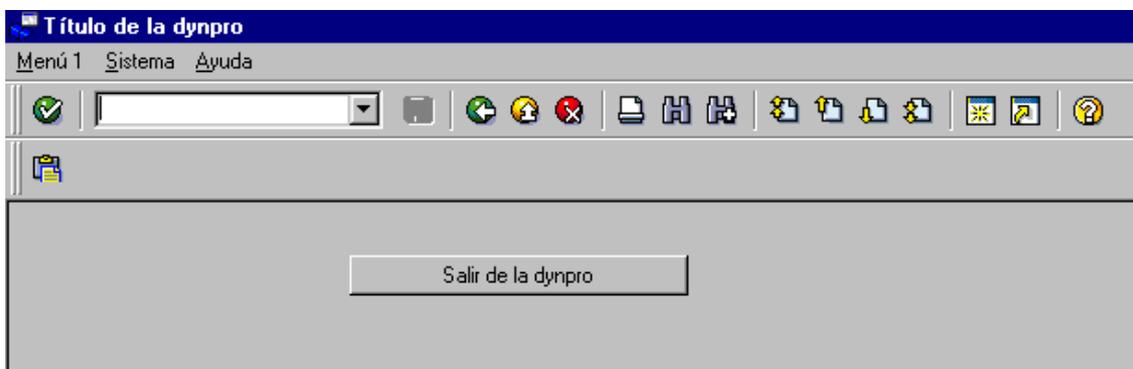
Posiciones 1 - 7						
Posiciones 8 - 14						
Posiciones 15 - 21						
Posiciones 22 - 28						
Posiciones 29 - 35						

↓
Código de función

En esta área solo tendremos que poner los códigos de funciones de las *Teclas de función* que hayamos creado con anterioridad y pulsar *Enter* para que SAP la coja. Voy a poner la *Tecla de función* que he creado anteriormente y el resultado sería este:

Posiciones 1 - 7	BOTON 			
------------------	--	--	--	--

Una vez colocadas las *Teclas de función* grabaremos y activaremos (botón de la cerilla) el *status* o *Menu Painter*. Volveremos hasta donde es llamado el *status* "PANT-1" activaremos el programa marco y volveremos para atrás para activar la *Dynpro*. Si ejecutamos la transacción que llama al *Modul pool* aparecería lo siguiente:



El control de que botón se pulsa en el *Menu painter* se haría desde el módulo *USER_COMMAND_XXXX*