

**PROGRAMACIÓN**

**EN**

**HR**



Título del documento

INDICE

- 1 OBJETIVOS..... 8
- 2 INTRODUCCIÓN A LA PROGRAMACIÓN EN HR..... 8
- 3 BASES DE DATOS LÓGICA..... 8
- 4 INFOTIPOS..... 11
  - 4.1 DECLARACIÓN ..... 11
  - 4.2 PROCESAMIENTO DEL INFOTIPO ..... 12
    - 4.2.1 *Procesar de un registro específico del infotipo* ..... 12
    - 4.2.2 *Procesamiento de todo el infotipo* ..... 13
    - 4.2.3 *Actualización de un infotipo*..... 14
  - 4.3 DEFINICIÓN DE UN INFOTIPO ..... 14
    - 4.3.1 *Estructura de la tabla PSnnnn*..... 15
    - 4.3.2 *Estructura de las tablas PAnnnn y PBnnnn*..... 15
      - 4.3.2.1 Tabla PAnnnn..... 16
      - 4.3.2.2 Tabla PBnnnn ..... 16
      - 4.3.2.3 Opciones técnicas ..... 16
    - 4.3.3 *Estructura Pnnnn* ..... 17
    - 4.3.4 *Estructura adicionales para campos de imagen*..... 17
    - 4.3.5 *Modulpool ABAP/4 de infotipo* ..... 17
    - 4.3.6 *Pantallas de un infotipo*..... 18
      - 4.3.6.1 Pantalla inicial ..... 19
      - 4.3.6.2 Pantalla individual..... 19
        - 4.3.6.2.1 Verificaciones..... 19
        - 4.3.6.2.2 Entradas posibles ..... 20
        - 4.3.6.2.3 Configuración de pantalla..... 20
        - 4.3.6.2.4 Lógica de proceso ..... 20
      - 4.3.6.3 Visualización de lista ..... 22
        - 4.3.6.3.1 Configuración de pantalla..... 22
        - 4.3.6.3.2 Lógica de proceso ..... 22
      - 4.3.6.4 Control de pantallas de un infotipo ..... 23
        - 4.3.6.4.1 Control de imágenes según la función a ejecutar ..... 24
        - 4.3.6.4.2 Control de imágenes dependientes de los datos de control ..... 25
      - 4.3.6.5 Status para los infotipos ..... 26
    - 4.3.7 *Módulos de dialogos de infotipo*..... 27
    - 4.3.8 *Características del infotipo* ..... 27
    - 4.3.9 *Módulos de texto de infotipo* ..... 28
      - 4.3.9.1 Visualización y actualización de textos en la pantalla individual ..... 28
        - 4.3.9.1.1 Configurar una imagen individual para visualizar y actualizar módulos de texto..... 28
  - 4.4 GUÍA PARA CREAR UN INFOTIPO ..... 29
    - 4.4.1 *Crear infotipos*..... 30
    - 4.4.2 *Actualizar las características del infotipo* ..... 30
    - 4.4.3 *Implementación funcional del infotipo* ..... 31
  - 4.5 AMPLIACIÓN DE INFOTIPOS ESTÁNDAR ..... 31
    - 4.5.1 *Ampliación de una imagen individual*..... 32
    - 4.5.2 *Ampliación de una visualización de lista* ..... 32

## Programación en HR

4.5.2.1	Borrar campos adicionales.....	32
4.6	MODIFICACIÓN DE INFOTIPOS .....	33
4.7	AMPLIACIÓN INFOTIPO PARA ENTRADA RÁPIDA DE DATOS.....	34
4.7.1	<i>Ampliación infotipo para entrada rápida de datos maestros (PA70)</i> .....	34
4.7.2	<i>Ampliación infotipo para entrada rápida de datos de acción (PA42)</i> .....	35
4.7.2.1	Definición de imagen de entrada.....	35
4.7.2.1.1	Lógica de proceso .....	36
4.7.2.2	Generación de codificación .....	36
4.7.2.3	Actualización de tabla .....	37
4.7.3	<i>Adaptación de status en Gestión de personal</i> .....	37
4.8	DATOS DE TIEMPOS.....	38
4.9	REPETIR ESTRUCTURAS .....	39
4.10	VERIFICACIÓN DE AUTORIZACIONES .....	40
4.11	VISTAS .....	40
4.11.1	<i>Acoplamiento</i> .....	41
4.11.2	<i>Proyección</i> .....	43
4.11.3	<i>Proyección y Acoplamiento</i> .....	44
4.11.4	<i>Tablas de control dependientes del tiempo</i> .....	46
4.11.5	<i>Generalización de la vista</i> .....	46
<b>5</b>	<b>IMPORTAR/EXPORTAR FICHERO EN RECURSOS HUMANOS.....</b>	<b>47</b>
5.1	FICHEROS PCL1, PCL2, PCL3 Y PCL4 .....	47
5.1.1.1	Información de los ficheros .....	47
5.1.1.2	Estructura del fichero.....	47
5.1.1.3	Convención de nombres.....	47
5.1.1.4	Importar/Exportar datos .....	47
5.1.1.5	Clave xx.....	48
5.2	ALMACENAMIENTO DE DATOS EN FICHEROS PCLN .....	48
5.2.1	<i>Ejemplo de descripción de datos mediante el cluster RX</i> .....	48
5.3	MEMORIA INTERMEDIA DE LOS FICHEROS PCLN .....	49
5.3.1	<i>Retroactividad resultados cálculo nomia</i> .....	49
5.3.2	<i>Inicio del cálculo de la nómina en el modo de test</i> .....	50
5.3.2.1	Export mediante la memoria intermedia de datos.....	50
5.3.2.2	Import mediante la memoria intermedia de datos .....	50
5.4	DIRECTORIO DEL CLUSTER.....	51
5.5	INTERFASE.....	52
5.5.1	<i>Lectura de un registro del cálculo de nómina</i> .....	52
5.5.2	<i>Evaluaciones estándar del cálculo de la nómina</i> .....	53
5.5.3	<i>Módulos de funciones: RP_EVALUATION_PERIODS</i> .....	53
5.5.3.1	Lectura resultado cálculo nómina usando RP_EVALUATION_PERIODS.....	54
5.5.4	<i>Indicadores de control</i> .....	54
5.5.5	<i>Acción</i> .....	55
5.5.6	<i>Códigos de retorno</i> .....	55
5.5.7	<i>Lectura de un registro del cálculo de la nómina</i> .....	55
5.5.8	<i>Importar resultado cálculo nómina de cluster RX</i> .....	56
<b>6</b>	<b>COMANDOS ESPECÍFICOS .....</b>	<b>56</b>
6.1	MÓDULOS DE FUNCIONES EN HR.....	56
6.2	MÓDULOS DE MACROINSTRUCCIONES.....	57
6.2.1	<i>Definición y llamada</i> .....	57
<b>7</b>	<b>MÓDULOS RMAC ESTÁNDAR EN HR.....</b>	<b>57</b>
7.1	DEFINICIÓN DE DATOS .....	58
7.1.1	<i>RP-LOWDATE-HIGHDATE</i> .....	58
7.1.1.1	Función .....	58
7.1.1.2	Parámetros.....	58
7.1.1.3	Verificación.....	58
7.1.1.4	Condiciones previas.....	58
7.1.1.5	Utilización .....	58
7.2	EVENTO <i>INITIALIZATION</i> .....	58

## Programación en HR

7.2.1	<i>RP-SEL-EIN-AUS-INIT</i> .....	59
7.2.1.1	Función .....	59
7.2.1.2	Parámetros .....	59
7.2.1.3	Verificación .....	59
7.2.1.4	Condiciones previas .....	59
7.2.1.5	Utilización .....	59
7.2.1.6	Observaciones .....	59
7.2.2	<i>RP-INIT-WITH-SYDATUM</i> .....	60
7.2.3	<i>Función</i> .....	60
7.3	<b>EVENTO START-OF-SELECTION</b> .....	60
7.3.1	<i>RP-SET-NAME-FORMAT</i> .....	60
7.3.1.1	Función .....	60
7.3.1.2	Parámetros .....	60
7.3.1.3	Verificación .....	60
7.3.1.4	Utilización .....	61
7.3.2	<i>RP-FETCH-ALTER-PERID</i> .....	61
7.3.2.1	Función .....	61
7.3.2.2	Parámetros .....	61
7.3.2.3	Verificación .....	61
7.3.2.4	Utilización .....	61
7.3.3	<i>RP-SET-KEY-DATE</i> .....	62
7.3.3.1	Función .....	62
7.3.3.2	Parámetros .....	62
7.3.3.3	Verificación .....	62
7.3.3.4	Condiciones previas .....	62
7.3.3.5	Utilización .....	62
7.4	<b>SELECCIÓN GET PERNR</b> .....	62
7.4.1	<i>RP-SEL-EIN-AUS</i> .....	62
7.4.1.1	Función .....	63
7.4.2	<i>RP-SEL-CALC</i> .....	63
7.4.2.1	Función .....	63
7.4.2.2	Parámetros .....	63
7.4.2.3	Verificación .....	63
7.4.2.4	Condiciones previas .....	63
7.4.2.5	Utilización .....	63
7.4.3	<i>RP-WRITE-ALTER-PERID</i> .....	63
7.4.3.1	Función .....	64
7.4.3.2	Parámetros .....	64
7.4.3.3	Verificación .....	64
7.4.3.4	Utilización .....	64
7.4.4	<i>RP-MAKE-ADDRESS</i> .....	64
7.4.5	<i>Función</i> .....	64
7.4.5.1	Parámetros .....	65
7.4.5.2	Verificación .....	65
7.4.5.3	Utilización .....	65
7.4.6	<i>RP-EDIT-ADDRESS</i> .....	65
7.4.6.1	Función .....	65
7.4.6.2	Parámetros .....	66
7.4.6.3	Verificación .....	66
7.4.6.4	Utilización .....	66
7.4.7	<i>RP-READ-ALL-TIME-ITY</i> .....	66
7.4.7.1	Función .....	67
7.4.7.2	Parámetros .....	67
7.4.7.3	Verificación .....	67
7.4.7.4	Condiciones previas .....	67
7.4.7.5	Utilización .....	67
7.4.7.6	Observaciones .....	68
7.4.8	<i>RP-EDIT-NAME</i> .....	68
7.4.8.1	Función .....	68
7.4.8.2	Parámetros .....	68
7.4.8.3	Verificación .....	69
7.4.8.4	Utilización .....	69
7.4.9	<i>RP-INIT-BUFFER</i> .....	69
7.4.9.1	Función .....	69

## Programación en HR

7.4.9.2	Parámetros.....	69
7.4.9.3	Verificación.....	69
7.4.9.4	Condiciones previas.....	69
7.4.9.5	Utilización.....	69
<b>7.4.10</b>	<b><i>RP-IMP-Cn-xx</i></b> .....	<b>70</b>
7.4.10.1	Función.....	70
7.4.10.2	Parámetros:.....	70
7.4.10.3	Verificación:.....	70
7.4.10.4	Condiciones previas:.....	70
<b>7.4.11</b>	<b><i>RP-EXP-Cn-xx</i></b> .....	<b>70</b>
7.4.11.1	Función.....	70
7.4.11.2	Parámetros.....	71
7.4.11.3	Verificación.....	71
7.4.11.4	Condiciones previas:.....	71
<b>7.4.12</b>	<b><i>RP-REF-Cn-xx</i></b> .....	<b>71</b>
7.4.12.1	Función.....	71
7.4.12.2	Parámetros.....	71
7.4.12.3	Verificación.....	71
7.4.12.4	Condiciones previas.....	71
<b>7.5</b>	<b>EVENTO TOP-OF-PAGE</b> .....	<b>72</b>
<b>7.5.1</b>	<b><i>RP-STICHTAG</i></b> .....	<b>72</b>
7.5.1.1	Función.....	72
7.5.1.2	Parámetros.....	72
7.5.1.3	Verificación.....	72
7.5.1.4	Condiciones previas.....	72
7.5.1.5	Utilización.....	72
<b>7.5.2</b>	<b><i>RP-ZEITRAUM</i></b> .....	<b>72</b>
7.5.2.1	Función.....	72
7.5.2.2	Parámetros.....	72
7.5.2.3	Verificación.....	73
7.5.2.4	Condiciones previas.....	73
7.5.2.5	Utilización.....	73
<b>7.6</b>	<b>EN CUALQUIER PUNTO</b> .....	<b>73</b>
<b>7.6.1</b>	<b><i>Obtención de datos</i></b> .....	<b>73</b>
7.6.1.1	<b>RP-READ-INFOTYPE</b> .....	<b>73</b>
7.6.1.1.1	Función.....	73
7.6.1.1.2	Parámetros.....	73
7.6.1.1.3	Verificación.....	74
7.6.1.1.4	Condición previa.....	74
7.6.1.1.5	Utilización.....	74
7.6.1.1.6	Observaciones.....	74
7.6.1.2	<b>RP-READ-T001P</b> .....	<b>75</b>
7.6.1.3	Función.....	75
7.6.1.4	Parámetros.....	75
7.6.1.5	Utilización.....	75
7.6.1.6	<b>RP-READ-T5DOP</b> .....	<b>75</b>
7.6.1.6.1	Función.....	75
7.6.1.6.2	Parámetros.....	75
7.6.1.6.3	Utilización.....	76
7.6.1.7	<b>RP-READ-PAYROLL-DIR</b> .....	<b>76</b>
7.6.1.7.1	Función.....	76
7.6.1.7.2	Parámetros.....	77
7.6.1.7.3	Verificación.....	77
7.6.1.7.4	Condiciones previas.....	77
7.6.1.7.5	Utilización.....	77
<b>7.6.2</b>	<b><i>Procesamiento de datos</i></b> .....	<b>77</b>
7.6.2.1	<b>RP-PROVIDE-FROM-FRST</b> .....	<b>77</b>
7.6.2.1.1	Función.....	77
7.6.2.1.2	Parámetros.....	77
7.6.2.1.3	Verificación.....	78
7.6.2.1.4	Condiciones previas.....	78
7.6.2.1.5	Utilización.....	78
7.6.2.1.6	Observaciones.....	78

## Programación en HR

7.6.2.2	RP-PROVIDE-FROM-LAST .....	78
7.6.2.2.1	Función.....	79
7.6.2.2.2	Parámetros .....	79
7.6.2.2.3	Verificación .....	79
7.6.2.2.4	Condiciones previas.....	79
7.6.2.2.5	Utilización .....	79
7.6.2.2.6	Observaciones.....	80
7.6.3	<i>Modificación de datos</i> .....	80
7.6.3.1	RP-UPDATE .....	80
7.6.3.1.1	Función.....	80
7.6.3.1.2	Parámetros .....	80
7.6.3.1.3	Verificación .....	80
7.6.3.1.4	Condiciones previas.....	80
7.6.3.1.5	Utilización .....	80
<b>8</b>	<b>UTILIDADES EN HR.....</b>	<b>80</b>
8.1	UTILIDADES GENERALES .....	80
8.2	UTILIDADES DE PROGRAMACIÓN.....	81
8.3	UTILIDADES DE CLUSTER.....	81

## Programación en HR

### 1 *Objetivos*

La finalidad de este manual es enseñar como acceder a los distintos tipos de contenedores de información, a nivel de ABAP/4, que SAP usa en Recursos Humanos (a partir de este momento lo denominaremos HR) para almacenar datos y mostrar las instrucciones especiales existentes en HR.

### 0 *Introducción a la programación en HR*

Desde siempre hemos visto que toda la información en SAP se guarda en tablas de diccionario. HR no es una excepción, pero además aparecen dos nuevos contenedores de información que son: **Bases de datos lógicas de HR, Infotipos y Tablas de Base de datos.**

En HR tenemos dos bases de datos lógicas que son *PNP* y *PAP*. La primera se refiere a los datos de los empleados y la segunda a los datos a los candidatos. El uso de Base de datos lógica facilita el acceso a los infotipos.

En los **Infotipos** se guarda la información referente al personal de la empresa: Datos de personales, Datos bancarios, Datos de nómina, Formación, Salud, Derecho de vacaciones, etc... en el sistema hay aproximadamente unos trescientos infotipos distintos, aunque muchos de ellos son específicos de cada país (como por ejemplo el infotipo 88 que guarda la información de la maternidad en Gran Bretaña).

Las **Tablas de base de datos** se dividen en áreas o clusters, estas áreas o cluster guardan información referente a las a los resultados de nóminas, Impuestos, Seguridad Social, etc... El acceso a estas tablas es el más complicado de todos, debido a que no se accede como en una tabla cluster normal y corriente, como la *INDX*, sino que se accede a través de macroinstrucciones.

Para acceder a los infotipos existen sentencias específicas para el acceso a infotipo y macroinstrucciones que es una combinación de estas últimas.

Las **Sentencias específicas** para el acceso a infotipos son aquellas que usaremos para declarar el infotipo en el programa (el equivalente a la sentencia *Tables*) y el acceso a dicho(s) infotipo(s), su equivalente sería el *Select*.

Las **Macroinstrucciones** es una posibilidad más de modularizar programas. Las macroinstrucciones contienen sentencias ABAP/4 que permiten el ahorro de trabajo.

### 0 *Bases de datos lógica*

Como se ha mencionado antes SAP tiene dos base de datos lógica: *PNP* que se refiere a los datos de empleado y la *PAP* que se refiere a los datos de candidatos. La base de datos que más se usa es la *PNP* y será dicha tabla que usaremos como ejemplo.

Cuando ejecutamos un report que accede a la base de datos lógica, esta carga los datos de personal de cada empleado en memoria para su tratamiento.

## Programación en HR

En memoria se carga el historial completo del empleado. Estos datos se pierden cuando se carga un nuevo empleado.

Cada vez que se usa la base de datos lógica se realizan dos verificaciones una de personal y otra de datos.

La primera se verifica que el usuario tiene acceso al empleado siguiendo los criterios de asignación organizacional. Si el usuario no tiene acceso al empleado, este no se evalúa.

La segunda verifica que el usuario tiene acceso a los infotipos especificados en el report.

Un ejemplo sencillo sería:

En los atributos del programa:

The screenshot shows a configuration window for a report class. At the top, there are icons for editing, a lightbulb, and a document, followed by tabs for 'Código fuente' and 'Clase report HR'. The main area contains the following fields:

- Título:** zpruebas
- Idioma maestro:** ES Español
- Creado:** 13.01.2003 DE00050
- Modificado por:** 13.01.2003 DE00050
- Status:** (empty)

Below these fields is a section titled 'Atributos' with the following settings:

- Tipo:** 1 Programa ejecutable
- Status:** (empty)
- Aplicación:** \* Multiaplicaciones
- Grupo autorizaciones:** (empty)
- Clase de desarrollo:** \$TMP ¡Los objetos temporales no se transportan nu...
- Base de datos lógica:** PNP MAESTRO RP
- Versión im.selección:** (empty)

At the bottom, there are four checkboxes:

- Mayúsculas/Minúsculas
- Cálculo de coma fija
- Bloqueo de editor
- Ejecución vía variante

El código fuente:

REPORT zprueba1.

\* PERNR es una estructura que es llenada por la base de datos lógica PNP

\* con los datos del empleado

TABLES: pernr." --> Datos de personal

\* Declaración de los infotipos

## Programación en HR

INFOTYPES: 0001.

START-OF-SELECTION.

```
GET pernr. " --> Usa base de datos lógica PNP
PROVIDE * FROM p0001 BETWEEN pn-begda AND pn-endda.
WRITE:/'P0001:', p0001.
ENDPROVIDE.
```

La pantalla de selección que nos saldría al usar la base de datos lógica *PNP* sería la siguiente:

The screenshot shows a software interface for data selection. At the top, there are three tabs: 'Otras selecciones', 'Ay.búsq.', and 'Clasificación'. Below the tabs, the 'Período' section contains radio buttons for 'hoy', 'hasta hoy', and 'otro período' (which is selected). There are also radio buttons for 'Mes actual', 'a partir de hoy', and 'Año actual'. Below these are input fields for 'Período selección personas' and 'Período cál.nómina'. The 'Selección' section below has a table with columns for field names, values, and a selection indicator. The fields listed are: 'Número de personal', 'Status ocupación', 'Sociedad', 'Área de nómina', 'DivPer/SubDPers./Centro cost', and 'Grupo personal/Área personal'. Each row has a 'a' in the middle column and a right-pointing arrow button in the right column.

Field	Value	Indicator	Action
Número de personal		a	→
Status ocupación		a	→
Sociedad		a	→
Área de nómina		a	→
DivPer/SubDPers./Centro cost		a	→
Grupo personal/Área personal		a	→

Los campos de la selección de datos están definidos en la estructura *QPPNP*, mientras que los campos de selección de personal están definidos en el programa include *DBPNPSEL* (Tablas internas que se completan mediante APPEND).

Cuando se usa la base de datos lógica *PNP* hay que declarar en el programa la estructura *PERNR*, en esta estructura se guardarán los números de personal seleccionados mediante la pantalla de selección.

En la estructura *PERNR* solo hay que leer el campo *PERNR*, que contiene el número de empleado, el resto de campos se encuentran allí por motivos internos.

Si se usa la base de datos lógica *PAP* habrá que declarar la tabla *APPLICANT*, para obtener el número de candidato.

## Programación en HR

Cuando se realiza el *GET* se rellenan las tablas internas (el nombre de la tabla interna es *Pnnnn*, *nnnn* es el número del infotipo) de los infotipos (declarados en la sentencia *INFOTYPES*) con los registros, desde los datos inferior hasta los datos superiores, del número de personal leído en el *GET*.

Más adelante se explicaría como se accede a los infotipos.

### 0 INFOTIPOS

Los infotipos se utilizan en HR y representan grupos de datos relacionados con un tema en particular. Los datos que se almacenan en un infotipo dependerán del número de personal de un empleado o del número de un candidato, es decir, cada empleado o candidato siempre se le asigna un registro de infotipo.

Cada infotipo está compuesto por un número de cuatro dígitos *nnnn*. El rango de número de 9000 a 9999 está reservado para los infotipos del cliente.

Mediante la transacción *PMO1 – Diálogos en la gestión de personal* se pueden crear y tratar infotipos.

En un infotipo no podemos añadir ni borrar información, solo podemos modificarla.

#### 0.0 Declaración

Antes de procesar un infotipo hay que declararlo. Para declararlo hay que usar la sentencia *INFOTYPES*.

El rango de infotipos es el siguiente:

- Entre el 0000 y 0999 tenemos los datos maestros del infotipo.
- Entre el 1000 y el 1999 tenemos los datos de planificación.
- Entre el 2000 y el 2999 tenemos los datos de tiempo.
- Entre el 3000 y 8999 no tenemos nada.
- Y entre el 9000 y 9999 tenemos los infotipos de cliente.

La sintaxis sería:

```
INFOTYPES: nnnn [NAME c] [OCCURS n] [MODE N] [VALID FROM comp1 TO comp2]
```

Donde *nnnn* es el número del infotipo que vamos a declarar.

## Programación en HR

Cuando se declara un infotipo en memoria se crea una tabla interna con la siguiente estructura:

```
DATA BEGIN OF Pnnnn OCCURS 10.  
  INCLUDE STRUCTURE Pnnnn.  
DATA END OF Pnnnn VALID BETWEEN BEGDA AND ENDDA.
```

La cláusula *VALID BETWEEN BEGDA AND ENDDA* indica que período se va a coger registros.

Los parámetros opcionales son:

- *NAME C* → Nos crea una tabla interna, con la misma estructura que el infotipo declarado, con el nombre indicado en *C*. *C* tiene un longitud máxima de 20 caracteres.
- *OCCURS n* → Nos crea una tabla interna poniendo el valor *n* en la cláusula *OCCURS*.
- *MODE n* → Esta cláusula solo esta disponible para las bases de datos lógica *PNP* y *PCH*, y su efecto es que no llena el infotipo cuando se ejecuta la sentencia *GET PERNR*.
- *VALID FROM comp1 TO comp2* → Cuando se ejecuta la sentencia *GET PERNR* el infotipo se llenará con los registros cuyo período este comprendido entre *comp1* y *comp2*.

### 0.0 Procesamiento del infotipo

Existen dos maneras de procesar un infotipo:

- Procesamiento de un registro específico del infotipo, el más reciente/más antiguo.
- Procesamiento de todo el infotipo.

#### 0.0.0 *Procesar de un registro específico del infotipo*

En muchos casos no necesitaremos leer todos los registros del infotipo, sino que necesitaremos leer al último o primer registro. Para ello utilizaremos las siguientes macroinstrucciones:

*RP-PROVIDE-FROM-LAST Pnnn SPACE PN-BEGDA PN-ENDDA* → Nos devuelve el último registro.

*RP-PROVIDE-FROM-FIRST Pnnn SPACE PN-BEGDA PN-ENDDA* → Nos devuelve el primer registro.

## Programación en HR

Donde *nnnn* es el numero de infotipo, en la variable *PN-BEGDA* o *PNPENDDA* tenemos la fecha inicial del periodo introducido por pantalla, y en *PN-ENDDA* o *PNPENDDA* tenemos la fecha final del periodo introducido por pantalla.

Un ejemplo sería:

```
REPORT zprueba1.
* PERNR es una estructura que es llenada por la base de datos lógica PNP
* con los datos del empleado
TABLES: pernr." --> Datos de personal
* Declaración de los infotipos
INFOTYPES: 0001.

START-OF-SELECTION.

GET pernr. " --> Usa base de datos lógica PNP
  rp_provide_from_last p0001 space pn-begda pn-endda.
* Si hemos encontrado datos entonces se muestra el registro.
  IF pnp-sw-found = '1'.
    WRITE:/ 'P0001:', p0001.
* En caso contrario leemos el siguiente numero de personal
  ELSE.
    REJECT.
  ENDIF.
```

Las macroinstrucciones se definen en la tabla *TRMAC*.

### *0.0.0 Procesamiento de todo el infotipo*

Ya hemos comentado que cuando se realiza un *GET PERNR* las tablas internas de los infotipos son llenadas con el historial completo (del más antiguo al más reciente) del empleado leído. Por lo tanto podemos tener que en una tabla interna existan más de un registro o períodos o intervalos de validez.

El procesamiento de registros del infotipo es dependiente del tiempo, es decir, depende del periodo de selección de datos introducido en la imagen de selección. Pueden procesarse al mismo tiempo los datos de varios infotipos y pueden prepararse para un período de tiempo parcial específico.

Las tablas internas del infotipo se procesan con la sentencia *PROVIDE*. La sintaxis es:

```
PROVIDE * FROM Pnnnn BETWEEN PN-BEGDA AND PN-ENDDA.
.....
ENDPROVIDE.
```

## Programación en HR

Donde *nnnn* sería el número del infotipo. La relación entre el infotipo y el período de selección de datos de la imagen de selección se establece mediante las variables *PN-BEGDA* (Inicio) y *PN-ENDDA* (Fin).

### ***0.0.0 Actualización de un infotipo***

Como hemos visto antes a los infotipos no se les puede insertar ni borrar, pero sí que podemos modificar sus datos. Para ello está la macroinstrucción *RP-UPDATE*. Cuya sintaxis es:

*RP-UPDATE datos\_antiguos datos\_nuevos.*

Un ejemplo sería:

REPORT ZIVAN .

TABLES: PERNR.

\* Definimos una tabla interna llamada OLD/NEW con la misma estructura

\* que 0001.

INFOTYPES: 0001 NAME OLD,  
                  0001 NAME NEW.

START-OF-SELECTION.

GET PERNR.

\* Cuando se efectura el GET las tablas internas OLD y NEW están llenas

\* con el resultado de la búsqueda

PROVIDE \* FROM OLD BETWEEN PNPBEGDA AND PNPENDDA.

WRITE:/, OLD-PERNR, 'OLD-PERSK:', OLD-PERSK.

\* Ponemos el nuevo valor a la tabla NEW

NEW-PERSK = '22'.

WRITE:/ 'NEW-PERSK:', NEW-PERSK.

\* Modificamos la tabla interna, no insertamos porque ya está llena

MODIFY NEW INDEX SY-TABIX.

ENDPROVIDE.

\* Llamamos a la macroinstrucción *RP-UPDATE* pasando la tabla interna

\* con los viejos y nuevos datos.

RP-UPDATE OLD NEW.

### **0.0 Definición de un infotipo.**

Un infotipo *nnnn* está formado por al menos dos estructuras y una tabla:

## Programación en HR

- Estructura *PSnnnn* → Esta estructura contiene todos los campos de datos del infotipo.
- Estructura *Pnnnn* → Contiene los campos clave del infotipo y los campos de la estructura *PSnnnn*.
- Tabla transparente *Pannnn* y/o *PBnnnn* → La tabla transparente *PAnnnn* se usa para los infotipos de gestión de personal. La tabla transparente *PBnnnn* se usa para los infotipos de contratación de personal.

Cuando se crea un infotipo hay que hacerlo a través del *Data Dictionary* del ABAP, dependiendo del release también se puede crear desde la transacción *PMO1 – Diálogos en la gestión de personal*.

Es muy posible que se necesite definir otras estructuras y tablas para infotipos específicos.

### ***0.0.0 Estructura de la tabla PSnnnn***

Cada infotipo incluye campos de datos almacenados únicamente dentro de ese infotipo en cuestión. Los campos son necesarios para definir las estructuras de los datos del infotipo y las tablas de las bases de datos. Los campos de datos se agrupan en la estructura *PSnnnn* para evitar la redundancia en la definición.

Cuando se crea la estructura *PSnnnn* hay que tener en cuenta que los campos no pueden estar formados por más de cinco dígitos y tampoco se pueden usar campos de tipo *INT4*.

Si deseamos usar subtipos hemos de duplicar el campo clave *Pnnnn-SUBTY* en la estructura *PSnnnn*. Este campo necesita su propio nombre y elemento de datos. El campo de subtipo se debe incluir en todas las pantallas del infotipo y además debe permitir efectuar entradas.

Cuando se actualizan las características del infotipo se debe especificar el nombre del campo del subtipo duplicado en el campo *Campo del subtipo* de la tabla T582A.

Los módulos centrales del infotipo registran automáticamente los datos en el campo clave *Pnnnn-SUBTY* a partir de las entradas en este campo.

Al usar un subtipo tendremos las siguientes ventajas:

- Se usan tablas de verificaciones especiales para el subtipo asignado al infotipo.
- Se puede crear documentación específica de campo para el subtipo y visualizarla mediante la tecla de ayuda.

### ***0.0.0 Estructura de las tablas PAnnnn y PBnnnn***

Los registros del infotipo *nnnn* se almacenan en las tablas de las bases de datos *PAnnnn* y *PBnnnn*. Dependiendo en que área se utilice el infotipo se utiliza una tabla u otra.

## Programación en HR

- Para infotipos de gestión de personal se usara la *PAnnnn*.
- Para infotipos de contratación se usara la tabla *PBnnnn*.

También se debe especificar las tablas de la base de datos que desea utilizar en la tabla *T852A – Características del infotipo*.

### *0.0.0.0 Tabla PAnnnn*

Nombre de campo	Clave	Elemento de datos	Tipo	Longitud	Tabla de verificación	Texto breve
MANDT	X	MANDT	CLNT	3	T000	Mandante
.INCLUDE	X	PAKEY				
.INCLUDE		PSHD1				
.INCLUDE		PSnnnn				

### *0.0.0.0 Tabla PBnnnn*

Nombre de campo	Clave	Elemento de datos	Tipo	Longitud	Tabla de verificación	Texto breve
MANDT	X	MANDT	CLNT	3	T000	Mandante
.INCLUDE	X	PBKEY				
.INCLUDE		PSHD1				
.INCLUDE		PSnnnn				

### *0.0.0.0 Opciones técnicas*

En estas tablas no es necesario crear índices secundarios ya que los datos se leen a través del índice primario.

En los campos de *Parámetros memoria lógicos* se ha de introducir lo siguiente:

- Clase de datos → El valor *APPL0*.
- Categoría de tamaño → El valor variará dependiendo de la utilización del infotipo.

En los campos de *Grabación en memoria intermedia* indicaremos que no se pueda grabar registros en memoria intermedia, esto es debido a que los programas que trabajan con estos infotipos necesitan datos actuales.

Los datos de los infotipos se graban en la memoria intermedia dentro de las aplicaciones de Gestión de personal independientemente de las parametrizaciones del Diccionario ABAP/4.

## Programación en HR

Se pueden introducir modificaciones en los registros de infotipo en forma de protocolos de modificación mediante la función de creación de logs de infotipos en Gestión de personal. El report RPUAUD00 le permite visualizar estos documentos. Rara vez es necesario realizar una grabación en log de las modificaciones de datos en el Diccionario ABAP.

### ***0.0.0 Estructura Pnnnn***

La estructura *Pnnnn* contiene los campos de datos de la estructura *PSnnnn* y los campos de datos incluidos en todos los infotipos. La estructura *Pnnnn* se compone de *PSHDR* y *PSnnnn*. La estructura de la tabla *PSHDR* contiene *PSKEY* y *PSHD1*.

La estructura *Pnnnn* contiene casi los mismos campos que las tablas *PAnnnn* y *PBnnnn*. Existen diferencias en los campos claves incluidos (*PSKEY* en lugar de *PAKEY* y *PBKEY*). Además en la estructura no es necesario poner el campo mandante.

Esta estructura se usa en los informes y modulpools de los infotipos. Además sirve como interfase entre el programa y la base de datos.

Los infotipos del cliente se incluyen automáticamente en la base de datos lógica *PNP*.

### ***0.0.0 Estructura adicionales para campos de imagen***

Al definir campos de imagen en el Screen Painter ABAP, no especificaremos datos estructurales (como tipo y longitud de datos) directamente cuando actualicemos la imagen. Es mejor especificar dichos datos indirectamente con el fin de que se tomen de la definición de objetos del Diccionario ABAP.

Cada campo se encuentra en una estructura diferente, dependiendo de su significado:

- En la estructura *RP50M* nos encontramos los campos que se visualizan para todos los infotipos, como por ejemplo las cabeceras de pantallas individuales.
- En la estructura *PSnnnn* se almacena los campos específicos del infotipo.

En una pantalla podemos incluir campos que aún no estén incluidos en una estructura. Para ello debemos crear la estructura *Znnnn* además de la estructura *Znnnn*. En la tabla *Znnnn* se pueden incluir todos los campos que deban visualizar en pantalla pero que aún no estén incluidos una estructura. El nombre de la estructura correspondiente para infotipos es *Qnnnn*.

### ***0.0.0 Modulpool ABAP/4 de infotipo***

Cada infotipo consta de un modulpool que constituye el programa principal de la interfase de la actualización del infotipo. El nombre del programa es *MPnnnn0*. *P* se refiere al número de personal y *nnnn* representa el número de infotipo.

## Programación en HR

El programa *MPnnnn0* solo contiene *includes*, si creamos el programa principal a través de la *PMOI* el sistema nos creará los siguientes *includes*:

- *MPnnnn10* → Declaración de objetos comunes
- *MPnnnn20* → Módulos PBO del programa.
- *MPnnnn30* → Módulos PAI del programa
- *MPnnnn40* → Subrutinas del programa.

Además de los *includes* propios del programa principal SAP inserta los siguientes *includes*:

- *FP5OPPSB* → Es la declaración de objetos comunes. Esta área se utiliza como memoria intermedia para registros info importados e información sobre actualización. Las variables de este *include* se usan como parámetros export/import cuando se accede al módulo de dialogo del infotipo.
- *MPPDAT00* → Declaración de objetos de datos comunes.
- *MPPER000* → Módulos de infotipos estándar.
- *MPPIRC00* → Definición de códigos de retorno de infotipos.
- *MPPREF00* → Definición de dos objetos de datos que contienen la cantidad de números de personal de referencia en la estructura *PO031* ó *PO121*.

A veces nos encontramos que muchos infotipos necesitan módulos que solo funcionan para un determinado país. Las definiciones de datos, PBO, PAI y subrutinas se han de almacenar por separado en *includes* propios. Después se introduce el indicador de país de HR que se asigna en la tabla *T500L* en la agrupación de países adecuadas, al final del nombre de *include* correspondiente. La nomenclatura para funciones adicionales sería, siguiendo la las convenciones:

- *MPnnnn5x* → Declaración de objetos comunes
- *MPnnnn6x* → Módulos PBO del programa.
- *MPnnnn7x* → Módulos PAI del programa
- *MPnnnn8x* → Subrutinas del programa.

La *x* al final del nombre se puede utilizar para una mayor distinción entre los distintos *includes*.

### ***0.0.0 Pantallas de un infotipo***

Cada infotipo dispone de al menos tres pantallas:

- Pantalla inicial
- Pantalla individual
- Visualización de lista

## Programación en HR

También es posible ajustar el control de pantallas para sustituir la pantalla individual o inicial por pantallas alternativas. También es posible usar otras pantallas individuales o visualizaciones de lista en HR, para adaptar una pantalla según las necesidades de un país determinado.

### ***0.0.0.0 Pantalla inicial***

La pantalla inicial de un infotipo se utiliza como una interfase técnica entre transacciones dentro de Gestión de personal y el infotipo en sí. Se accede a través del módulo de diálogo asignado al infotipo en cuestión.

El número de dynpro asignado para la pantalla inicial es el 1000. La pantalla 1000 del modulpool *MPMMMMMOO* se utiliza como modelo.

La pantalla inicial del infotipo se procesa en fondo, o sea, no se visualice aunque se procese. La funcionalidad de dicha pantalla es:

- Procedimiento de inicialización general, es necesaria para todos los infotipos.
- Acceso a la pantalla individual.
- Etapas de proceso generales una vez que se ha procesado la pantalla individual

La imagen inicial siempre se tiene que crear con la transacción *PMO1 – Diálogos en HR*, ya que la genera con todas las funciones requeridas. **Esta pantalla no es modificable.**

### ***0.0.0.0 Pantalla individual***

La imagen individual permite la comunicación entre el sistema y usuario. Y permite: crear, visualizar o actualizar registros individuales de infotipos.

Por regla general a la pantalla individual se le asigna la dynpro 2000, pero esto es susceptible de ser cambiado. La dynpro 2000 del modulpool *MPMMMMMOO*.

Además se pueden crear pantallas individuales propias para infotipos estándar de SAP. El área de nombre que se le asigna es entre el 2900 al 2999.

#### ***0.0.0.0.0 Verificaciones***

Los valores del infotipo *0001 - Asignación organiz.* válidos al principio del período de validez del registro y las entradas en las tablas *T001, T001P, T500P* y *T503* válidas en la estructura *PSYST* le permiten efectuar verificaciones de entrada específicas de infotipo.

Esto significa que el sistema no necesita leer el infotipo *0001* o las tablas mencionadas arriba. Tan solo hay que declararlas en la sentencia *TABLES*.

## Programación en HR

Hay que tener en cuenta que los valores de la estructura *P0001* y *P0002* no se suelen inicializar y por lo tanto no es recomendable su uso.

### *0.0.0.0.0 Entradas posibles*

En el sistema se visualizarán todas las entradas posibles que se verifiquen de acuerdo con una tabla.

Si aun campo del diccionario se le asigna una tabla de verificación capaz de verificarse automáticamente, el sistema mostrará automáticamente las posibles entradas.

### *0.0.0.0.0 Configuración de pantalla*

Las seis primeras líneas de la pantalla inicial siempre son iguales para todos los infotipos:

- De la primera a la tercera línea se muestran los datos del empleado.
- La cuarta línea permanece en blanco.
- La quinta línea se visualizan los campos de *Fecha DE*, *Fecha A*, *Campo de bloqueo*, *Campo de texto*, *Modificado por* y *Modificado el*.

Los campos específicos del infotipo se visualizan de la séptima línea a la vigésima primera.

Todos los campos en pantalla se han de mostrar en un marco. Aunque el campo que indica el subtipo puede estar por encima del primer marco.

### *0.0.0.0.0 Lógica de proceso*

Si se crea la lógica de proceso del infotipo mediante la transacción *PMO1 – Diálogos en la Gestión de personal* el sistema nos creará la siguiente lógica de proceso estándar:

En el PBO tendríamos:

*PROCESS BEFORE OUTPUT.*

*MODULE BEFORE\_OUTPUT. → No modificable*

*MODULE Pnnnn.*

*MODULE HIDDEN\_DATA. → No modificable*

Dentro del módulo PBO *Pnnnn* se pueden llevar a cabo procedimientos de inicialización específicos de infotipo. Le permiten, por ejemplo, completar los campos de pantalla almacenados en las estructuras *Qnnnn* y *Znnnn*

## Programación en HR

Si se evalúan indirectamente las CC-nóminas, debe completarse el campo *Importe Q0014-BETRG* dentro de la lógica de proceso del infotipo 0014 *Dev./deducc. period.* porque el importe no se almacena en la base de datos.

En el PAI tendríamos:

*PROCESS AFTER INPUT.*

*MODULE EXIT AT EXIT-COMMAND  
CHAIN.*

*FIELD Pnnnn-campo1, ... MODULE INPUT\_STATUS ON CHAIN-REQUEST.*

*ENDCHAIN.*

El módulo *PAI INPUT\_STATUS* se deberá crear cuando el usuario pueda realizar alguna entrada en un campo de pantalla. Por este motivo, se deben tener en cuenta todos los campos de entrada. El módulo *PAI INPUT\_STATUS* fija status de sistemas internos: si se modifica un valor, por ejemplo, el registro de infotipo debe almacenarse con posterioridad.

*MODULE PRE\_INPUT\_CHECKS.*

El módulo *PAI PRE\_INPUT\_CHECKS* se utiliza para procesar el código de función antes de la verificación de entrada. Si selecciona la función *Finalizar*, por ejemplo, el sistema interrumpe el proceso de la pantalla individual actual. Una vez que este módulo ha finalizado podemos incluir módulos propios o verificaciones a medida.

Cuando hemos llegado a este punto debemos tener en cuenta que las verificaciones a los campos de debe estar completa, ya que en el proceso siguiente ya no se podrá modificar el contenido del campo.

El módulo *PAI POST\_INPUT\_CHECKS* procesa el código de función después de la verificación de entrada. Además también efectúa la verificación de entrada general. Esto garantiza, por ejemplo, que la fecha DE de un registro de infotipo sea anterior a la fecha A del mismo registro.

Todos los campos que se visualicen en la pantalla individual deben tenerse en cuenta en la siguiente cadena:

*CHAIN.*

*FIELD Pnnnn-feld1,RP50M-SPRPS,Tnnn-felda,...*

*MODULE POST\_INPUT\_CHECKS.*

*ENDCHAIN.*

Los módulos siguientes no pueden modificarse: *PAI EXIT*, *INPUT\_STATUS*, *PRE\_INPUT\_CHECKS* y *POST\_INPUT\_CHECKS*.

## Programación en HR

### *0.0.0.0 Visualización de lista*

Este tipo de visualización permite ver todos los registros de datos de un infotipo específico creado para un número de personal.

Este tipo de visualización usa la dynpro 3000, pero también es susceptible de ser cambiado. Como modelo se coge la dynpro 3000 del modulpool *MPMMMMMOO*.

También se pueden crear visualizaciones de lista propias de infotipos estándar, el ámbito de creación sería entre el 3000 y 3999.

### *0.0.0.0.0 Configuración de pantalla*

La visualización de lista consta de tres áreas:

- De las líneas uno y tercera línea se visualiza las líneas de cabecera.
- De las líneas quinta y decimonovena contiene la lista de los registros del infotipo.

Por regla general se usa los campos asignados a la estructura *Pnnnn*. Si deseamos mostrar otros campos podemos actualizarlos con otros campos. Los registros de infotipo se pueden visualizar en una o varias líneas.

- Los campos de selección: *RP50M-BEGDA*, *RP50M-ENDDA*, *RP50M-SUBTY*, *RP50M-ABGRD* y *RP50M-PAGEA* se muestran en la línea vigésima.

Estos campos permiten seleccionar los registros del infotipo basándose en el período validez, el subtipo, la fecha de limitación o la posición de lista.

En estos campos, exceptuando en el *RP50M-ABGRD* (Fecha de limitación), está permitido efectuar entradas. El campo de fecha de limitación solo debería mostrarse si la intención es realmente imitar.

Si la visualización de lista se crea a través de la transacción *PMO1 – Diálogos en la Gestión de personal* el sistema configura automáticamente la visualización de lista.

### *0.0.0.0.0 Lógica de proceso*

Si creamos la visualización de lista mediante la transacción *PMO1 – Diálogos en la Gestión de personal* el sistema creará la lógica de proceso.

En el PBO:

*PROCESS BEFORE OUTPUT.*  
*MODULE BEFORE\_OUTPUT.*  
*LOOP.*

## Programación en HR

*MODULE PSLIST.* → No modificable.

*MODULE Pnnnn.*

*ENDLOOP.*

Dentro del módulo *Pnnnn* se lleva a cabo los procedimientos de inicialización específico del infotipo. Si se necesitan crear procedimientos de inicialización diferentes se han de crear en un módulo aparte que deberá llamarse *PnnnnL*.

En el PAI:

PROCESS AFTER INPUT.

MODULE EXIT AT EXIT-COMMAND.

LOOP.

FIELD RP50M-SELEC MODULE MARK ON REQUEST.

ENDLOOP.

CHAIN.

FIELD RP50M-BEGDA.

FIELD RP50M-ENDDA.

FIELD RP50M-SUBTY.

MODULE SELECT\_FOR\_LIST ON CHAIN-REQUEST.

ENDCHAIN.

MODULE POST\_INPUT\_CHECKS.

Los módulos que no se pueden modificar: *PAI EXIT*, *MARK*, *SELECT\_FOR\_LIST* y *POST\_INPUT\_CHECKS*.

### 0.0.0.0 *Control de pantallas de un infotipo*

Al crear imágenes individuales y visualizaciones de lista con el *Screen Painter*, pueden determinarse los atributos de las imágenes individuales. Sin embargo, siempre se utiliza la misma pantalla para varias funciones, como crear, visualizar, actualizar y borrar registros de infotipo. Por esta razón, no se puede especificar si se pueden efectuar entradas en un campo de pantalla al actualizar la pantalla. También es posible que determinados campos de pantalla se encuentren ocultos, en función de los datos organizativos del empleado.

Es decir, algunos atributos no están especificados por norma, sino que se especifican en el tiempo de ejecución. El aspecto de las pantallas cambia dependiendo de la función que seleccione el usuario o de los datos que se procesen.

Las mismas pantallas individuales se utilizan para las funciones *Visualizar datos maestros* y *Actualizar datos maestros*. No obstante, sólo debe ser posible efectuar entradas en los campos si se selecciona la función *Actualizar datos maestros*. Por este motivo, debe determinarse si se puede o no realizar entradas en los campos, dependiendo de la función a ejecutar.

Las pantallas utilizadas para introducir datos del coche de la empresa en el infotipo 0032 *Datos de la empresa* deben estar ocultos para los empleados asignados al grupo de personal de pensionistas.

## Programación en HR

Las funciones de control de pantallas ya se han implementado en el programa principal de infotipos. Estas funciones leen los valores desde los grupos de modificación de cada uno de los campos de la pantalla y fijan los atributos en función de los valores. El significado de los valores de los grupos de modificación se determina en las tablas.

Al desarrollar infotipos, dispone de las siguientes opciones para el control de pantallas:

- Se puede determinar que las entradas se realicen en campos o bien optar por ocultar los campos de pantalla, dependiendo de la función a ejecutar.

El valor del *Grupo de modificación 1* le permite determinar que se puedan realizar entradas en campos de pantalla. También es posible ocultar cada uno de los campos de una pantalla.

El valor del *Grupo de modificación 1* se debe actualizar para todos los campos de entrada.

- Se pueden utilizar dynpros alternativos o determinar que se puedan realizar entradas en campos u ocultar campos de pantallas mediante los datos de control de la tabla *T588M*.

En relación con la tabla *T588M*, el valor del *Grupo de modificación 3* determina la actividad y si se pueden realizar entradas en campos.

Si se utilizan ambas posibilidades para el control de pantallas de un campo de pantalla, tiene prioridad el control de pantallas mediante la tabla *T588M*.

El *Grupo de modificación 2* se utiliza internamente.

El *Grupo de modificación 4* no se utiliza en la versión estándar porque se reserva para clientes.

### 0.0.0.0.0 Control de imágenes según la función a ejecutar

Dependiendo de la función que vayamos a ejecutar podemos: Determinar si las entradas se pueden realizar en campos de pantallas individuales y ocultar los campos de imágenes individuales.

El *screen painter* nos permite actualizar el valor del *Grupo de modificación 1* para los campos de imagen en cuestión. El valor del *Grupo de modificación 1* se debe actualizar para todos los campos de pantalla en los que se pueden realizar entradas.

El significado de los valores en el *Grupo de modificación 1* se determina en la tabla *T589A*. Para determinar si se pueden efectuar entradas en campos de pantalla en la versión estándar, se definen las siguientes constantes:

Se puede efectuar una entrada en campo de pantalla para la función	Constante hexadecimal para grupo de modificación 1
Visualizar	001
Modificar	002
Añadir y copiar	004
Borrar	008
Bloquear/desbloquear	010

## Programación en HR

Para ocultar campos de pantalla, en la versión estándar se definen las siguientes constantes:

Se oculta campo de pantalla para la función	Constante hexadecimal para grupo de modificación 1
Limitar en visualización de lista	200
Visualizar en visualización de lista y Modificar en visualización de lista	400
Añadir y copiar	800

El valor del *Grupo de modificación 1* se interpreta por bits. Se puede combinar más de una constante entre sí. No hay que olvidar que se debe actualizar el valor del *Grupo de modificación 1* en forma hexadecimal.

- Si desea realizar entradas en un campo de pantalla al utilizar las funciones *Añadir y Modificar* debe actualizar el valor 006 en el *Grupo de modificación 1*.
- Si desea realizar entradas en un campo de pantalla al utilizar todas las funciones debe actualizar el valor 00F en el *Grupo de modificación 1*.

En el caso de campos específicos de pantalla para pantallas individuales o visualizaciones de lista, se especifica previamente para todos los infotipos la parametrización que determinará si se puede o no realizar entradas. Si se crea la pantalla individual o la visualización de lista mediante la transacción *PMO1 - Diálogos en la Gestión de personal*, el sistema introduce el valor correcto en el *Grupo de modificación 1* para estos campos de pantalla.

- Normalmente se pueden efectuar entradas en los campos *BEGDA* y *ENDDA* para todas las acciones, aparte de visualizar registros. Por esta razón, para estos campos al atributo *Grupo de modificación 1* se le asigna el valor 00E.
- El *Grupo de modificación 1* tiene el valor 800 para los campos *AEDTM* y *UNAME*. Esto garantiza que estos campos se encuentren ocultos cuando se añada un registro.
- La visualización de lista permite realizar entradas en los campos *RP50M-BEGDA*, *RP50M-ENDDA*, *RP50M-SUBTY* y *RP50M-PAGEA* con el fin de poder seleccionar registros. A estos campos se les asigna el valor 00F porque debe ser posible realizar una entrada en cada operación.
- La fecha de limitación en el campo *RP50M-ABGRD* sólo debería visualizarse en la visualización de lista si la función actual es realmente limitar. Por este motivo, el *Grupo de modificación 1* se actualiza mediante el valor 400.
- Sólo se pueden seleccionar registros múltiples en la visualización de lista si se utilizan las funciones de visualización y limitación. Al campo *RP50M-SELEC*, contenido en un loop, se le asigna el valor 009 para el *Grupo de modificación 1*.

### 0.0.0.0 Control de imágenes dependientes de los datos de control

En el control de pantallas según la función a ejecutar, se puede

## Programación en HR

- Sustituir la imagen estándar por una imagen alternativa
- Determinar si las entradas pueden realizarse en campos de imagen individual
- Ocultar los campos de imágenes individuales.

El control de pantalla se puede llevar a cabo:

- De forma general
- En función de los datos organizativos del empleado
- En función del subtipo del registro de infotipo.

El *screen painter* permite actualizar el valor del *Grupo de modificación 3* para los campos de imagen en cuestión.

En el *Grupo de modificación 3*, a cada campo de pantalla se le asigna un valor entre 001 y 001. Si desea modificar campos de pantalla de la misma forma, utilice el mismo valor. Si se trata de un campo de entrada/salida, se utiliza el mismo valor que el de la palabra clave correspondiente y un texto explicativo que se puede visualizar. Si los campos de pantalla no se pueden modificar mediante la tabla *T588M*, asigne el valor *SPACE* en el *Grupo de modificación 3*.

A continuación hay que utilizar la tabla *T588M* para determinar

- si se han utilizado imágenes alternativas y cuáles
- cómo se han modificado los campos de imagen individuales.

Si necesita más información sobre control de pantallas según datos de control, consulte la IMG sobre *Gestión de personal* y la documentación sobre la etapa *Fijar modificación de pantalla* de la sección *Adaptación de las superficies*.

### **0.0.0.0 Status para los infotipos**

La interfase de pantallas individuales y de visualizaciones de lista es estándar en todos los infotipos. Se utiliza un status de interfase específico según la función a ejecutar. Cuando se utilizan determinadas funciones es posible desactivar opciones de menú particulares o teclas de función definidas en el status de interfase.

El status de interfase se configura en un módulo PBO de la versión estándar. Si se crea el infotipo mediante la transacción *PM01 - Diálogos en la Gestión de personal*, la lógica de proceso de las pantallas de infotipo accede automáticamente al módulo PBO. Por esta razón, no es necesario programar el status de interfase.

El módulo PBO que fija el status de interfase sólo funciona correctamente si el nombre y la estructura del status de la interfase a utilizar cumplen las convenciones SAP. Por este motivo, debe utilizarse también la transacción *PM01 - Diálogos en la Gestión de personal* para crear la interfase del infotipo.

En el único sitio donde no hay que crear un status es la pantalla inicial.

La listade status necesarias es la siguiente:

Pantalla	Status de interfase	Uso del status de interfase para la función
----------	---------------------	---

## Programación en HR

Pantalla individual	DIS	Visualizar
	MOD	Modificar
	DEL	Borrar
	COP	Copiar
	INS	Añadir
	EDQ	Bloquear
Visualización de lista	LISO	Visualización de lista/visualizar
	LIS1	Visualización de lista/actualizar
	LIS9	Visualización de lista/limitar

### 0.0.0 Módulos de diálogos de infotipo

Cada infotipo necesita un módulo de diálogo que represente la interfase entre las transacciones utilizadas en Gestión de personal y el infotipo en sí. El nombre del módulo de diálogo debe ser *RP\_nnnnr*, *nnnr* se refiere al número del infotipo.

Al actualizar el módulo de diálogo, éste se asigna a un infotipo. Se debe especificar el nombre del modulpool ABAP/4 y el número de la pantalla inicial del infotipo.

El infotipo se asigna al módulo de diálogo en la tabla *T582A* o por el nombre del módulo de diálogo.

El infotipo *0002 - Datos personales* utiliza el modulpool *MP000200* y la dynpro 1000 como pantalla inicial. Por lo tanto, este infotipo necesita un módulo de diálogo llamado *RP\_0002* que accede a la pantalla 1000 del modulpool *MP000200*.

### 0.0.0 Características del infotipo

Las características de un infotipo las determinan las entradas almacenadas en varias tablas. Para cada infotipo se deben de actualizar las tablas siguientes:

- *T582A* → Características básicas del infotipo: Pantallas, módulos de dialogo, etc...
- *T582S* → Texto breve de los infotipos.

La vista *V\_T582A* permite actualizar estas dos tablas.

Otras posibles tablas que pueden ser necesarias modificar:

- *T591A* → Esta tabla se usa cuando el infotipo esta dividido en subtipos. La tabla que se usa para guardar las características del subtipo es la *T582A*, como tabla de subtipos.

## Programación en HR

- *T588G* → Esta tabla controla el lanzamiento de retroactividad por campos.
- *T588M* → Permite adaptar las pantallas del infotipo. Se puede especificar una pantalla alternativa o siguiente o también se puede especificar un control de pantalla dependiente de usuario.
- *T588B* → *Menús de infotipo*.
- *T588Z* → Medidas dinámicas.

Las entradas almacenadas en estas tablas deben actualizarse manualmente. La transacción *PM01- Diálogos en Gestión de personal* le permite actualizar las características básicas de infotipo y configurar los menús de infotipo.

### ***0.0.0 Módulos de texto de infotipo***

SAP nos permite crear un módulo de texto al introducir los datos maestros de registros individuales de datos de infotipo. Estos elementos de texto se almacenan en el fichero *PCLI*, en el ID de cluster *TX*.

Para asegurarse de que los módulos de texto puedan crearse para un infotipo, al actualizar las características de infotipos (tabla *T582A*), se debe marcar el campo *Texto admitido (T582A-INFTX)*.

Al visualizar o actualizar un registro de infotipo, es posible visualizar o actualizar el texto. Para ello, hay que ir a la función *Tratar* → *Visualizar texto* o *Tratar* → *Actualizar texto* de la pantalla individual del infotipo.

#### ***0.0.0.0 Visualización y actualización de textos en la pantalla individual***

Se puede también visualizar o actualizar las tres primeras líneas de texto en la pantalla individual del infotipo.

Si queremos utilizar esta función, tenemos que ajustar simplemente la pantalla individual en cuestión. No es necesario modificar las estructuras ni las tablas del infotipo del Dictionary ABAP/4.

Las tres primeras líneas del texto de la pantalla individual del infotipo 0019 *Seguimiento fechas* se visualizan o pueden actualizarse.

##### ***0.0.0.0.0 Configurar una imagen individual para visualizar y actualizar módulos de texto***

Si deseamos visualizar o poder actualizar las tres primeras líneas de la pantalla individual del infotipo, hay que seguir estos pasos:

- Se ha de modificar la visualización de la pantalla individual del infotipo en cuestión.
- Incluir los campos *RP50M-TEXT1*, *RP50M-TEXT2*, *RP50M-TEXT3* en la pantalla individual.

Para asegurarse de que se pueden realizar entradas en estos campos al utilizar las funciones *Añadir* y *Modificar*, hay que actualizar el valor 006 en el *Grupo de modificación 1* en los tres campos.

## Programación en HR

- Aumentar la lógica de proceso de la acción *PROCESS BEFORE OUTPUT*.

Insertar el módulo *GET\_TEXT* después del módulo *HIDDEN\_DATA*. Por lo tanto, el módulo *GET\_TEXT* es el último módulo de esta acción al que se accede.

- Aumentar la lógica de proceso de la acción *PROCESS AFTER INPUT*.

Insertar las siguientes líneas después del módulo *PRE\_INPUT\_CHECKS* y antes de las verificaciones de entrada específicas de infotipo:

*CHAIN.*

*FIELD: RP50M-TEXT1, RP50M-TEXT2, RP50M-TEXT3.*

*MODULE UPDATE\_TEXT ON CHAIN-REQUEST.*

*ENDCHAIN.*

También se han de incluir los campos *RP50M-TEXT1*, *RP50M-TEXT2*, *RP50M-TEXT3* en la cadena para el módulo *POST\_INPUT\_CHECKS* con el fin de que se puedan realizar entradas en estos campos cuando aparezca el mensaje *W200 Grabe la entrada*.

- Verificar que el indicador *Texto permitido* está activado en la tabla *T582A*.

Si no se ha activado este indicador, los campos *RP50M-TEXT1*, *RP50M-TEXT2*, *RP50M-TEXT3* se ocultan.

Con ello se puede actualizar los textos del infotipo. Las tres primeras líneas del texto de la pantalla individual se visualizan o pueden actualizarse.

### 0.0 Guía para crear un infotipo

Es importante recordar que los infotipos de cliente se asignan al ámbito del 9000 al 9999.

Lo primero que hay que hacer a la hora de crear un infotipo es crear los objetos parciales para ello disponemos de la transacción *PMO1 - Diálogos en la Gestión de personal*. Una vez creados los objetos parciales, se integrará el infotipo a las transacciones de gestión de personal y/o Contratación de personal. Los infotipos se incluirán automáticamente a la base de datos lógica *PNP*.

Los infotipos se crean en cuatro etapas:

- Primero se crea la definición del infotipo en el diccionario.
- Segundo se crea el programa principal que contiene la función estándar del infotipo.
- Tercero se crea el módulo de diálogo que accederá a la pantalla inicial del infotipo.
- Cuarto se actualizan las características del infotipo.

Para los puntos 2 y 3 se puede usar la transacción *PMO1 - Diálogos en la Gestión de personal*.

## Programación en HR

Si se crean las definiciones *PSnnnn* y *Pnnnn* en el diccionario antes de generar el programa principal, esto se tendrá en cuenta cuando se genere dicho programa. Y también cuando se genera el programa todos los campos de la estructura *PSnnnn* se incluyen en la lógica de proceso.

### 0.0.0 Crear infotipos

Para crear un infotipo tenemos que ir a la *PM01 - Diálogos en la Gestión de personal* y seguir los siguientes pasos:

- En el campo *Nº Infotipo* hay que introducir un número de cuatro dígitos (hay que introducir ceros a la izquierda) del infotipo que vamos a crear.
- En el marco o *frame* de *Objeto Parcial* seleccionar *Estructura PS*. Se puede crear un objeto parcial si se selecciona la casilla correspondiente en el marco *Objetos Parciales* y luego pulsar sobre *Crear*.
- Una vez seleccionado el botón/Icono de *Crear* nos saldrá el *ABAP Dictionary*.
- Una vez en el *Dictionary* crearemos la estructura *PSnnnn* y activamos la estructura.
- Volvemos a la imagen de *Crear Infotipo*.
- Seleccionamos *Crear todo* y con esto crearemos la estructura *Pnnnn*.
  - Si se ha marcado el *Infotipo de empleado* se creará la tabla *PAnnnn*.
  - Si se ha seleccionado el *Infotipo de candidatos* se creará la tabla *PBnnnn*.
  - También se crearán los siguientes subobjetos:
    - ModulPool ABAP/4. Programa principal: *MPnnnn00*. Includes del programa: *MPnnnn10*, *MPnnnn20*, *MPnnnn30* y *MPnnnn40*.
    - Imágenes. Dynpros 1000, 2000 y 3000 para el modulpool *MPnnnn00*.
    - Interfase donde contendrá todos los status de interfase necesarios.
    - Módulo de dialogo *RP\_nnnn*.
    - Entrada en la tabla *T777D* con las características técnicas del infotipo.
    - Entrada en la tabla *T777ID* para la estructura de campo de datos *PSnnnn* para el infotipo
- Por último podemos crear las estructuras adicionales que se necesiten.

Una vez creado el infotipo tendremos que realizar las configuraciones adecuadas en la tabla *T582A* en el customizing para *Gestión de personal*.

### 0.0.0 Actualizar las características del infotipo

Cuando se crea un infotipo, el sistema no crea las entradas de tabla que definen las características del infotipo, que son las tablas *T582A* y *T582S*. Por esto motivo tenemos que realizar las entradas manualmente en estas tablas o de una manera más cómoda en la vista *V\_T582*.

## Programación en HR

Para actualizar las características del infotipo hay que ir a la transacción *PM01 - Diálogos en la Gestión de personal* y realizar lo siguiente:

- Seleccionar *Diálogo* → *Crear infotipo*
- Seleccionar *Características IT* y nos llevará a *Visualizar vista infotipo: Imagen resumen*.
- Creamos una nueva entrada con los datos del infotipo. También se puede crear una copia de una entrada ya existente para crear la nuestra.

### **0.0.0 Implementación funcional del infotipo**

En esta etapa es donde realizaremos las configuraciones necesarias para las visualizaciones de lista e individuales, crear procedimientos nuevos de inicialización para campos de pantalla, realizar validaciones propias, etc...

Para realizar todas estas tareas podemos usar la transacción *PM01 - Diálogos en la Gestión de personal* o usar el *Object Browser* del *Workbench ABAP*.

## **0.0 Ampliación de infotipos estándar**

La ampliación de infotipos estándar consta de:

- Realizar validaciones adicionales, mediante la transacción *CMOD - Ampliación de sistema*
- Incluir campos adicionales al infotipo. Esto quiere decir que se puede incluir campos nuevos a la estructura *PSnnnn*, con lo cual se pueden actualizar los campos en la pantallas individuales.

Si incorporamos campos nuevos en un infotipos, estos serán tratados de la misma manera que los campos estándar de SAP en la gestión de informes, cuando se crean los documentos y en los eventos dinámicos.

La ampliación de un infotipo estándar no crea problemas a la hora de realizar un upgrade de liberación.

En las ampliaciones de un infotipo estándar tenemos las siguientes limitaciones:

- Los infotipos a los cuales no se les puede hacer una ampliación son:
  - *0000 - Acciones*
  - *0302 - Acciones adicionales*
  - *2nnn - Gestión de tiempos*
- la longitud de la estructura *PSnnnn* del campo de datos y del *include* CI no puede superar los 1500 Bytes.

## Programación en HR

- Si se incluyen campos adicionales en el infotipo *0000 – Asignación organizacional*, estos no aparecerán en la base de datos lógica *PNP* en los campos de selección.

### ***0.0.0 Ampliación de una imagen individual***

La ampliación de una imagen individual se realiza con la transacción *PM01 - Diálogos en la Gestión de personal*:

- Seleccionar *Ampliar infotipos*
- Introducir el número del infotipo (introduciendo ceros por la izquierda) que deseamos crear. Los infotipos *0000 – Acciones* y los infotipos de *Gestión de tiempos* no se pueden crear.
- En el marco o *frame Objeto Parcial* marcar *Include CI*.
- Pulsar sobre *Crear*.
- Crear el include CI y activarlo.
- Volver a la imagen de *Crear infotipo* y seleccionar *Crear todo*.

### ***0.0.0 Ampliación de una visualización de lista***

La ampliación de una visualización de lista se realiza con la transacción *PM01 - Diálogos en la Gestión de personal*:

- Seleccionar *Visualización de lista*.
- Introducir el número del infotipo (introduciendo ceros por la izquierda) que deseamos crear.
- En el campo *Numero de imagen* se ha de introducir el numero de la lista a ampliar y seleccionar *Crear*.
- Cuando se seleccione *Crear* se visualizara el *Data Dictionary*
- En el *Dictionary* se crea la estructura de la visualización de lista.
- Activar la estructura y volver a *Ampliar visualización de lista* en la transacción *PM01 - Diálogos en la Gestión de personal* y seleccionar *Crear todo*.

Los campos adicionales que se insertan se visualizan en la lista, aunque todavía esta en blanco.

Se puede introducir datos en estos campos mediante el procedimiento *FORM FILL-LISTSTRUCT* en el programa generado *Zpnnnn00*. Esta rutina se llama por cada registro de la lista.

La estructura *ZPLISnnnn* se detecta vía generación por medio de una certificación de *TABLES* en el programa *ZPnnnn00*. Entonces se pueden introducir los datos en el campo de la estructura *Pnnnn*, o por medio de la lectura de tablas de texto.

### ***0.0.0.0 Borrar campos adicionales***

## Programación en HR

Para borrar campos adicionales incluidos en la visualización de lista estándar seleccionando *Ampliar visualización lista* → *Ampliar infotipos* → *Visualización de lista* → *Borrar infotipo no estándar*.

Esta función borra los campos en la estructura *ZPLISnnnn* de la visualización de lista estándar.

### 0.0 Modificación de infotipos

Cuando se efectúa un cambio de *release* o de versión, la estructura o función de los infotipos de la versión estándar puede modificarse.

Las modificaciones en infotipos de cliente descritas en esta sección no son compatibles con tales cambios. Por esta razón, después de cada actualización de *release* se debe verificar y posiblemente adaptar las modificaciones.

Es posible añadir nuevos objetos parciales a los infotipos de la versión estándar, o sea, se pueden crear: Pantallas específicas del cliente y *includes* específicos del cliente para un infotipo existente.

Podemos usar el *Control de imágenes* (explicado con anterioridad) conjuntamente con la tabla *T588M* para sustituir las imágenes del infotipo por la nuestras propias.

Si lo que se desea es añadir includes específicos del cliente a un infotipo, se debe de modificar el modulpool *MPnnnn00* del infotipo.

Si se añaden nuevos objetos parcial se tiene que usar clases de desarrollo específicas del cliente. Es importante utilizar clases de desarrollo específicas de cliente y observar las convenciones para definiciones de nombre con el fin de que no se pierdan sus desarrollos al actualizar el sistema.

Los objetos parciales específicos de cliente se asignan a los siguientes ámbitos de nombres:

Objeto parcial	Ámbito de nombres
Pantallas individuales específicas de cliente	2900 a 2999
Visualizaciones de lista específicas de cliente	3900 a 3999
Includes para declaraciones de datos específicos de cliente	MPnnnn5x
Includes para módulos PBO específicos de cliente	MPnnnn6x
Includes para módulos PAI específicos de cliente	MPnnnn7x
Includes para subrutinas específicas de cliente	MPnnnn8x

En el nombre de los includes, *nnnn* hace referencia al número de infotipo que desea modificar. El último carácter *x* puede definirse como desee.

### 0.0 Ampliación infotipo para entrada rápida de datos

Si desea utilizar funciones de entrada rápida para su infotipo en Gestión de personal, debe ampliarlo. Se pueden realizar diferentes tipos de ampliación en función del tipo de entrada rápida de datos que se solicite.

#### 0.0.0 Ampliación infotipo para entrada rápida de datos maestros (PA70)

Para introducir datos con rapidez para más de una persona en el infotipo *nnnn* hay que ir a la transacción *PM01 - Diálogos en la Gestión de personal* y realizar lo siguiente:

- Una vez en la transacción *PM01* se han de copiar las dynpros 4000 y 5000 del modulpool *ABAP MPMMMMMOO* al modulpool *MPnnnnOO*.
  - La dynpro 5000 se utiliza para la entrada rápida de registros de infotipos
  - La dynpro 4000 es necesaria para introducir valores de propuesta, si se desea utilizar la función *Crear con valores de propuesta* para entrada rápida de datos.
- Las dynpros 4000 y 5000 del programa *MPnnnnOO* se han de tratar con el screen painter. En el screen painter se incluirán los campos en las pantallas que deben sugerirse como propuesta, o deben actualizarse, para la entrada rápida de datos. Y se tendrá que introducir el valor *006* en el *Grupo de modificación 1* de los campos de entrada.

También se tiene que borrar los indicadores de verificación automática de clave externa en la pantalla 5000. Si hace falta, se puede incluir verificaciones de entrada de valores de propuesta en la lógica de proceso.
- Se ha de verificar que los status se han copiado en ambas pantallas. Para las entradas rápidas de datos maestros se necesitan los siguientes status:
  - FINS → Crear
  - FMOD → Modificar
  - FEDQ → Bloquear/Desbloquear
  - FDEL → Borrar
- Se han de incluir el *include MPPERSFS* en el modulpool *MPnnnnOO*.
- Con la transacción *PM01* se ha de copiar el *include MPMMMMMFS* en el *include MPnnnnFS* y se tiene que incluir en el modulpool *MPnnnnOO*. Después es añadir los campos visualizados como valores de propuesta en las posiciones que se soliciten.
  - Para la definición del registro de campo *PROPOSED\_VALUES*
  - En subrutina *DNNNV*
- Actualizar las tablas *T588R – Reports de selección para entrada rápida* y *T588Q – Tipos de pantalla para entrada rápida* en el infotipo. A través del IMG se puede actualizar dichas tablas: *Gestión de personal* → *Adaptación de procesos de trabajo* → *Menús de infotipo* → *Adaptar menú de infotipo para entrada rápida*.

## Programación en HR

- Incluir el infotipo en el menú para la transacción PA70. A través del IMG: *Gestión de personal* → *Adaptación de procesos de trabajo* → *Menús de infotipo* → *Modificar menú de infotipo*.

En la entrada rápida de datos se fijan los siguientes menús de infotipo: 07 para entrada rápida de datos maestros y 09 para entrada rápida de datos de tiempos.

### ***0.0.0 Ampliación infotipo para entrada rápida de datos de acción (PA42)***

Es posible definir una entrada rápida de datos en cada acción en la tabla T529A.

Se han de introducir los campos necesarios en la imagen de entrada rápida para cada infotipo asignado al infogrupo de la acción. La mayoría de los infotipos ya tienen asignados valores de propuesta de las tablas y generalmente no se especifican en la pantalla.

Los datos especificados se registran en la tabla interna *PROPOSED\_VALUES*, que se pasan al infotipo 0000 - *Acciones*.

El infotipo de acción accede a los infotipos asignados uno tras otro al infogrupo y distribuye los datos introducidos en la imagen de entrada rápida.

Por regla general, los diálogos de infotipo se ejecutan en proceso de fondo. Las pantallas individuales de infotipo se visualizan sólo si existe algún error.

No obstante, es posible utilizar la pantalla de entrada rápida para determinar si se ha ejecutado un infotipo en proceso de fondo si, por ejemplo, se necesitan introducir datos adicionales de un empleado o datos no estándar, que no pueden introducirse en la pantalla de entrada rápida.

### ***0.0.0.0 Definición de imagen de entrada***

Los modelos que se usarán para copiarlos:

- Programa *SAPMP5X0* dynpro 100 para versiones nacionales de la acción de contratación; 'X' debe reemplazarse según convenga.
- Programa *SAPMP50F* dynpro 100 para acciones que no actualicen infotipos nacionales; se pueden añadir otras imágenes.

La pantalla de entrada rápida para acciones debe ser compatible con el infogrupo que pertenezca a la acción.

Si se incluyen otros campos adicionales en la pantalla de entrada rápida, éstos no se pasan a los infotipos ni se almacenan.

Si se pierden campos de los infotipos procesados por la acción, se visualiza la imagen individual del infotipo con el correspondiente mensaje de error.

En la pantalla de entrada se actualizan tres tipos diferentes de campos:

- Campos de estructuras de infotipo *Pnnnn* (campos P). Los contenidos de estos campos se pasan a los infotipos adecuados en donde se verifican y se almacenan. A estos campos se les asigna el valor 001 en el *Grupo de modificación 1*.

## Programación en HR

- Campos de estructuras de infotipo *Qnnnn* (campos Q). Estos campos se usan como las pantallas individuales de infotipo donde el formato externo es distinto al interno.  
El campo *Q0002-ANREX* ("Sr." "Sra.") consta del campo invisible *P0002-ANRED* ("1", "2") con el valor "001" en el *Grupo de modificación 1*. A los campos *Q* no se les asigna ningún valor en el *Grupo de modificación 1*.
- Casillas de selección para procesar pantallas de infotipo individuales si existen adicionales, abreviadas como *campos condicionales* a partir de ahora. A estos campos se les asigna el valor *002* en el *Grupo de modificación 1*.

### *0.0.0.0 Lógica de proceso*

En el PBO:

#### *MODULE INIT OUTPUT.*

El status *f 'RÁPIDO'* se crea en el módulo *INIT*. Si la integración se activa con *RP\_PLAN*, los siguientes campos *P0001-KOSTL*, *P0001-ORGEH*, *P0001-STELL* se modifican para que no se pueden llevar a cabo entradas.

En el PAI:

#### *MODULE EXIT*

Si se introduce "E" en el campo de comandos, se sale de la pantalla de diálogo.

#### *MODULE PERNR*

El número de personal asignado se bloquea y se realiza la lectura de datos.

#### *MODULE FCODE*

Trata el código de función pulsado: En el caso que se pulse el botón de *Grabar*, los datos se registran en la tabla interna *PROPOSED\_VALUES* y se accede al infotipo *0000*.

En el caso de una acción de contratación, el sistema verifica la existencia del número de personal especificado. Los datos no se almacenan.

### *0.0.0.0 Generación de codificación*

La transacción *PM01* permite generar codificaciones desde la pantalla de entrada.

- Definición de datos  
Las estructuras de infotipo *Pnnnn* y *Qnnnn* y los campos adicionales *ADD\_nnnn* se declaran en el *include MPxxxTOP*.
- Módulos de salida  
Las sentencias *MODULE* y *ENDMODULE* de los módulos de salida se incluyen en el *include MPxxx000*.

## Programación en HR

- Módulos de entrada  
Las sentencias *MODULE* y *ENDMODULE* de los módulos de entrada se incluyen en el *include MPxxx100*.
- Subrutinas  
El procedimiento *FORM FILL\_PROPOSED\_VALUES\_nnnn* (nnnn = número de dynpro) se incluye en el *include MPxxxFOO*.  
En esta procedimiento, la tabla interna *PROPOSED\_VALUES* se completa con los nombres y contenidos de los campos procesados por la acción. El *Grupo de modificación 1* de los campos se interpreta en la pantalla de entrada.

El procedimiento sería:

Se crea una nueva pantalla de entrada para un modulpool.

Los *includes* con definiciones de datos, módulos de salida y entrada y procedimientos se complementan convenientemente

Se modifica una pantalla de entrada para la que ya existe una codificación.

Los *includes* con definiciones de datos, módulos de salida y de entrada se complementan con nuevos objetos.

Si se borran campos o módulos, los objetos correspondientes no se borran de los *includes* porque pueden utilizarlos otras pantallas.

Si la rutina *FILL\_PROPOSED\_VALUES\_nnnn* ya existe en la pantalla actual *nnnn*, se visualiza una ventana de diálogo con una consulta de seguridad y se reconstruye la rutina.

### ***0.0.0.0 Actualización de tabla***

En la tabla *T588A*, a la transacción de entrada rápida *PA42* se le asigna un menú de acción que se actualiza en la vista *V\_588B\_M*. Las acciones sólo están permitidas si tienen una entrada rápida de datos.

En la tabla *T529F*, el nombre de un módulo de diálogo en el que el módulo de diálogo esté determinado se asigna a las acciones con entrada rápida de datos.

En la tabla *TDCT*, se incluye una entrada para el módulo de diálogo.

### ***0.0.0 Adaptación de status en Gestión de personal***

Para adaptar los status y procedimientos de Gestión de personal, se ha de realizar lo siguiente:

- Se puede determinar las cabeceras individuales para pantalla individuales y visualizaciones de lista de infotipos. Desde el *customizing: Gestión de personal* → *Adaptación de superficies* → *Fijar cabeceras dynpro*.
- Se puede determinar pantallas individuales y visualizaciones de lista alternativas, ocultar cada uno de los campos en pantalla y especificar si se pueden realizar entradas o no según que campos, mediante la tabla *T588M* para llevar el control de pantallas con la ayuda de datos de control.

## Programación en HR

- Se puede incluir un infotipo en un menú de infotipo para actualizar y visualizar datos maestros de *Gestión de Personal*. Desde *customizing: Gestión de personal* → *Adaptación de los procesos de trabajo* → *Menús de infotipo* → *Modificar menú de infotipo*.
- Para incluir un infotipo en una medida, a través del *Customizing: Gestión de personal* → *Adaptación de los procesos de trabajo* → *Acciones*.

### 0.0 Datos de tiempos

Los infotipos del 2000 al 2999 son infotipos de tiempos. Los datos de tiempos de estos infotipos se almacenan en las tablas de la PA2000 a la PA2999. Como hemos visto los infotipos se declaran con la sentencia *INFOTYPE* y los datos se ponen a disposición para su procesamiento en las tablas internas del infotipo (el infotipo 2011 es una excepción).

No debe cargar en la memoria principal todos los registros de infotipo de tiempos desde los datos del sistema inferiores hasta los superiores. Esto puede producir rápidamente una sobrecarga de memoria, especialmente si existe un sistema de captura de tiempos front end conectado a su sistema de HR.

Éste es el motivo por el cual los datos de tiempos deben leerse sólo para un período determinado.

Utilice el suplemento de declaración del infotipo *MODE N* para definir que las tablas internas de infotipos de tiempos sean declaradas, pero no cumplimentadas en el punto del evento *GET PERNR*.

Más adelante podrá cumplimentar estas tablas mediante una certificación con parámetros de período de selección.

El siguiente ejemplo permite leer los datos de tiempos:

```
REPORT RPABAP05.  
TABLES: PERNR.  
INFOTYPES: 2001 MODE N.  
GET PERNR.  
RP-READ-ALL-TIME-ITY PN-BEGDA PN-ENDDA.  
LOOP AT P2001.  
WRITE: / P2001-ABWTG.  
ENDLOOP.
```

Las macroinstrucciones nos permite leer los datos de tiempo. Esta macroinstrucción utiliza el parámetro del período de selección de datos de la imagen de selección.

Debido al enlace temporal de los infotipos, deberán tenerse en cuenta varias características especiales al procesar datos de tiempos. Las vistas de datos de tiempos no acostumbran a ser prácticas.

En los infotipos de tiempos, los datos se determinan basándose en el período de validez.

Cuando se introduce un registro de absentismo, el número de días de absentismo se calcula sobre la base del período de absentismo.

## Programación en HR

En una vista, los meses parciales nuevos se crean sin que se realice ninguna modificación en los datos del infotipo. Esto produciría resultados incorrectos en los infotipos de tiempos, puesto que estos datos dependen del período de validez.

Por ejemplo, si un registro de vacaciones se extiende desde mediados de enero hasta mediados de febrero y se calculan 20 días de vacaciones para este período, entonces una vista del mes de febrero daría como resultado un registro de vacaciones que se extendería desde el principio hasta mediados de febrero. El número de días de vacaciones no se habría modificado y la información sería incorrecta.

En los datos maestros, el enlace temporal es una característica definitiva del infotipo o subtipo. No existen dependencias de tiempo entre infotipos y subtipos.

Básicamente los datos de tiempos son diferentes. Supongamos, que un empleado se pone enfermo durante las vacaciones. De este modo, el registro de vacaciones se delimita al primer día de la enfermedad y, a continuación, el registro de enfermedad.

De la misma forma, el sistema no permite introducir un registro de vacaciones que coincida con un registro de enfermedad. Ocurre lo mismo en el caso de las horas extras realizadas durante una enfermedad.

La dependencia del tiempo de los registros de infotipos de tiempos no está limitada sólo a dependencias entre registros de uno y el mismo infotipo, sino que cubre todos los infotipos y subtipos.

El enlace temporal de los infotipos de tiempos no es un atributo, sino que se define por las relaciones entre infotipos.

Además, determinados registros de infotipos de tiempos tienen tiempos de llegada y salida específicos. Por consiguiente, podrían existir varios registros para un infotipo en un día concreto.

Puesto que las vistas requieren datos explícitos y los infotipos de tiempos no cumplen esta condición previa, no debería utilizar objetos agregados ni extrapolaciones para datos de tiempos.

Las tablas de infotipos de tiempos se tratan con la certificación *LOOP*, puesto que la certificación *PROVIDE* limita y, por consiguiente modifica, las fechas de inicio y fin de los infotipos con el período de selección de datos.

### 0.0 Repetir estructuras

En muchos infotipos de datos maestros, los datos se introducen en forma de tabla. Esto, por ejemplo, permite introducir hasta veinte CC-nóminas diferentes y sus importes. La imagen de entrada, dispone de siete líneas para entradas de CC-nóminas y, si utiliza la función de desplazamiento, puede introducir hasta veinte CC-nóminas.

La estructura de la línea de CC-nómina está almacenada en la estructura del infotipo *P0008* y los campos individuales están numerados del uno al veinte. Cada campo de la tabla de CC-nóminas está definido.

Al evaluar estructuras de evaluación, hay que asegurarse de que se introducen todos los campos. En el caso del infotipo de emolumentos básicos, se escanean los campos  $20 * 5 = 100$ .

Un *loop* ofrece un método de evaluación más atractivo. En éste, se evalúa una línea de la estructura de repetición cada vez que se ejecuta el *loop*.

Para utilizar este método de evaluación, defina un *string* de campo cuya estructura se corresponda con los campos de una línea de la estructura de repetición.

En este *string* de campo, se evalúa una línea de las CC-nóminas de emolumentos básicos cada vez que se ejecuta el *loop*.

## Programación en HR

```
REPORT RPABAPO6.  
TABLES:PERNR.  
INFOTYPES: 0008.  
DATA: BEGIN OF WAGETYPES,  
      LGA LIKE P0008-LGA01,  
      BET LIKE P0008-BET01,  
      ANZ LIKE P0008-ANZ01,  
      EIN LIKE P0008-EIN01,  
      OPK LIKE P0008-OPK01,  
      END OF WAGETYPES.  
  
GET PERNR.  
RP-PROVIDE-FROM-LAST P0008 SPACE PN/BEGDA PN/ENDDA.  
DO 20 TIMES VARYING WAGETYPES  
FROM P0008-LGA01  
NEXT P0008-LGA02.  
IF WAGETYPES-LGA IS INITIAL.  
EXIT.  
ELSE.  
WRITE: / WAGETYPES-LGA, WAGETYPES-BET.  
ENDIF.  
ENDDO.
```

Las estructuras de repetición se encuentran también en los infotipos de *Derecho a vacaciones*, *Distribución costes*, *Calificaciones* y *Garantía salarial*. Las estructuras se evalúan en consecuencia.

### 0.0 Verificación de autorizaciones

La verificación de autorización en los programas de HR no se diferencia de la realizada en las transacciones. Puesto que el acceso a datos en Reporting consiste siempre en un tipo de lectura, el sistema verifica si existe una autorización para la lectura; el grupo de autorizaciones debe ser **R** o **\***.

Para efectuar una validación simplificada cuando se ejecute un *report*. Se usarán los objetos *RPROGIN* y *RPABAP*, que son necesarios para la verificación; si estas autorizaciones están disponibles, se realiza una verificación más sencilla y rápida.

Si el programa de HR no puede leer determinados datos de personal debido a la falta de autorización, no se procesan todos los datos para estas personas en el evento *GET PERNR*. Aparece una nota al final de la lista informando del número de personas que se han descartado debido a la falta de autorización.

### 0.0 Vistas

Cuando se evalúan los datos, se diferencian entre la vista lógica y la vista física.

La vista física se corresponde con el formato de los datos del infotipo almacenados en las tablas de personal. Estos datos se guardan en registro del infotipo con un período de validez.

En la vista lógica, los períodos de validez de los campos individuales se determinan para varios registros del infotipo. Para una evaluación puede ser importante en qué tabla

## Programación en HR

horaria trabajaba el empleado, en qué trabajo independientemente o si se ha producido una modificación de una sociedad o una división de personal durante este tiempo.

Además, se suministran los datos procedentes de varios infotipos para un mes parcial específico. Cuando se calculan los periodos del cálculo de la nómina, es especialmente importante que los datos de los emolumentos básicos, el plan de horario de trabajo y la distribución de costes se suministren para el mes parcial.

Estos dos tipos de vistas lógicas se implementan en la proyección y el acoplamiento.

### ***0.0.0 Acoplamiento***

Un proceso de acoplamiento de dos o más infotipos. Se suministran los datos procedentes de estos infotipos para un mes parcial específico.

En el siguiente ejemplos nos gustaría saber en qué tabla horaria trabajó un empleado, en qué función y en qué dirección vivía durante este tiempo.

Están disponibles las siguientes direcciones:

- Enero a Junio → Barcelona
- Junio a Diciembre → Lleida

Están disponibles los siguientes datos de función:

- Enero a Abril → Programador
- Mayo a Diciembre → Consultor

Si las direcciones y los datos de función se suministran para meses parciales, esto tiene el siguiente resultado:

- Enero a Abril → Barcelona/Programador
- Mayo a Junio → Barcelona/Consultor
- Julio a Diciembre → Lleida/Consultor

La sintaxis de este acoplamiento sería la siguiente:

```
PROVIDE * FROM Pmmmm  
* FROM Pnnnn  
BETWEEN PN-BEGDA AND PN-ENDDA.
```

Los meses parciales para infotipos *Pmmmm* y *Pnnnn* y para todos los demás infotipos del acoplamiento se definen en los campos *BEGDA* y *ENDDA*.

Los datos de cada infotipo en el acoplamiento deben estar disponibles durante todo el periodo de validez de infotipo. Las tablas horarias de los registros del infotipo pueden no superponerse; por ello, el acoplamiento no puede contener infotipos con un enlace temporal de "tres".

## Programación en HR

La tabla horaria de los registros se superponen, si se lee un infotipo sin ningún tipo de restricciones de subtipo. Por ejemplo, el infotipo *Dirección* tiene los subtipos *residencia permanente*, *segunda residencia* y *dirección en el país de origen*.

Las tablas horarias al final se superpondrán, si se leen todos los destinatarios. Por ello, debe seleccionar siempre un subtipo para un acoplamiento, y este subtipo no necesita tener el enlace temporal de tres.

El programa para el acoplamiento citado más arriba y la dirección es:

*REPORT ZACLOP.*

*TABLES: pernr.*

*INFOTYPES: 0001, 0006.*

*GET pernr.*

```
PROVIDE * FROM p0001 * FROM p0006  
BETWEEN pn-begda AND pn-endda  
WHERE p0006-subty eq '1'.  
WRITE: / pernr-pernr, p0001-stell,  
p0006-stras, p0006-begda, p0006-endda.  
ENDPROVIDE.
```

A veces no hay datos disponibles para un determinado infotipo en el mes parcial seleccionado. Los períodos de validez del infotipo puede que no se superpongan, pero están permitidos los vacíos.

Por ejemplo, los vacíos pueden producirse cuando se acoplan los datos personales con direcciones.

Datos personales:

- Enero 1960 – Mayo 1993 → Pérez
- Mayo 1993 – Diciembre 1993 → Huerta

Direcciones:

- Enero 1993 – Diciembre → Barcelona

El acoplamiento para datos personales y direcciones sería el siguiente:

- Enero 1960 – Diciembre 1992 → Pérez
- Enero 1993 – Abril 1993 → Pérez/Barcelona
- Mayo 1993 – Diciembre 1993 → Huerta/Barcelona

## Programación en HR

En el primer mes parcial sólo están disponibles los datos de personal. Puesto que el registro no suministra la información necesaria, la función de acoplamiento para el suministro de datos de los infotipos asociados no se ha cumplido.

La variable *Pnnnn\_VALID* identifica que sólo los datos incompletos están disponibles para un mes parcial determinado.

Si los datos *están disponibles para el infotipo Pnnnn* en el mes parcial especificado, la variable *Pnnnn\_VALID* contiene el valor *X*.

En el siguiente ejemplo se muestra como se evaluarían las variables:

*REPORT ZEJEMPLO.*

*TABLES: PERNR.*

*INFOTYPES: 0002, 0006.*

*GET PERNR.*

*PROVIDE \* FROM P0002 \* FROM P0006 BETWEEN PN-BEGDA AND PN-ENDDA  
WHERE P0006-SUBTY = '1'.*

*IF P0006\_VALID EQ 'X'.*

*WRITE: / PERNR-PERNR,*

*P0002-BEGDA DD/MM/YYYY,*

*P0002-ENDDA DD/MM/YYYY,*

*P0002-NACHN,*

*P0006-ORT01.*

*ENDIF.*

*ENDPROVIDE.*

Sólo se genera una lista, si las direcciones están disponibles. El primer mes parcial, que sólo tiene datos personales, se suprime.

### ***0.0.0 Proyección***

Todos los datos de un infotipo se almacenan en la base de datos conjuntamente con su período de validez.

Cuando usted modifica uno o más campos de un registro del infotipo, el sistema crea un nuevo registro con una validez temporal nueva. La fecha en la que usted modifica el registro es la fecha de inicio de este nuevo registro.

Los datos de campos que no están afectados por las modificaciones mantienen los mismos datos a lo largo de varios registros del infotipo y de períodos de validez.

Desde una perspectiva lógica, estos campos son válidos en todos los registros del infotipo hasta que se modifican.

Si se mira desde una perspectiva lógica, cada campo de un infotipo tiene su propio período de validez.

Esto se ilustra en el siguiente caso:

Un empleado ha trabajado como programador durante tres años en tres divisiones de personal diferentes.

Los siguientes datos de asignación organizativos están disponibles:

## Programación en HR

- Enero 1992 a Diciembre 1992 → Programador / división de personal 1
- Enero 1993 a Diciembre 1993 → Programador / división de personal 2
- Enero 1994 a Diciembre 1994 → Programador / división de personal 3

Si sólo necesita la tabla horaria durante el tiempo en que un empleado realiza una función específica y no su división de personal para una evaluación, se aplica lo siguiente:

- Enero 1992 a Diciembre 1992 → Programador

La vista física tiene tres registros del infotipo, la vista lógica tiene uno.

Para crear evaluaciones significativas y evitar redundancias, crear vistas lógicas para los registros del infotipo.

Se seleccionan los campos del infotipo importantes para la evaluación y hay que hacer caso omiso de los otros.

Los datos de los otros campos no son válidos para la evaluación, puesto que no se sabe a qué división de personal y a qué empleado pertenecen para los años 1992 - 1994.

Esta vista del período de validez de un grupo de campos del infotipo se conoce como proyección.

El trozo del programa para la proyección es:

```
PROVIDE <campo_1> <campo_2> <campo_n> FROM Penn  
BETWEEN PN-BEGDA AND PN-ENDDA.
```

Los datos del infotipo para una proyección deben estar disponibles a lo largo de todo el período de validez. Si las tablas horarias de determinados registros del infotipo se superponen, los datos no se pueden asignar con claridad a un período. Por ello, no debería utilizar proyecciones para los registros del infotipo junto con el enlace temporal "tres".

El report para la proyección indicada más arriba es:

```
REPORT EJEMPLO.  
TABLES: PERNR.  
INFOTYPES: 0001.  
GET PERNR.  
PROVIDE STELL FROM P0001 BETWEEN PN-BEGDA AND PN-ENDDA.  
WRITE: / PERNR-PERNR, P0001-STELL, P0001-BEGDA,  
P0001-ENDDA.  
ENDPROVIDE.
```

La validez lógica para el período de la función está disponible en el infotipo de los campos *BEGDA* y *ENDDA*.

### **0.0.0 Proyección y Acoplamiento**

Las dos vistas lógicas se pueden combinar en los datos del infotipo, el acoplamiento y la proyección.

## Programación en HR

Se leen los datos de diferentes infotipos y se crean nuevos meses parciales. Seleccionamos los campos del infotipo importantes para la evaluación y combinamos esos meses parciales de nuevo.

En el siguiente ejemplo se ilustra lo dicho con anterioridad.

Un empleado trabaja como programador durante el ejercicio en curso y se casa en mayo. Su nombre (de ella) no se modifica.

Asignación organizacional:

- Enero a Diciembre → Programador

Datos personales:

- Enero a Abril → Eva Sánchez – Soltera
- Mayo a Diciembre → Eva Sánchez - Casada

Cuando se leen los datos de ambos infotipos a la vez, el resultado es:

- Enero a Abril → Eva Sánchez – Soltera/Programador
- Mayo a Diciembre → Eva Sánchez – Casada/Programadora

Puesto que podemos hacer caso omiso de su estado civil en la evaluación, proyectamos en su nombre y apellidos:

- Enero a Diciembre → Eva Sánchez/Programadora

El ejemplo siguiente nuestro lo citado más arriba:

*REPORT ZEJEMPLO.*

*TABLES: PERNR.*

*INFOTYPES: 0001, 0002.*

*GET PERNR.*

*PROVIDE STELL FROM P0001*

*NACHN VORNA FROM P0002*

*BETWEEN FN/BEGDA AND FN/ENDDA*

*IF P0001\_VALID = 'X':*

*WRITE: / P0002-NACHN, P0002-VORNA,*

*P0001-BEGDA DD/MM/YYYY,*

*P0001-ENDDA DD/MM/YYYY,*

*P0001-STELL.*

*ENDIF.*

*ENDPROVIDE.*

## Programación en HR

Este report combina los períodos de validez asociados y suministra los datos importantes de los campos del infotipo para un período específico. Aquellos campos a los que no se accede tienen su valor inicial en la proyección.

El suministro de datos para un mes parcial específico es especialmente importante para un mes parcial del cálculo alícuota en el cálculo de la nómina.

Si los emolumentos básicos de un empleado o la distribución de costes se modifican durante el período del cálculo de la nómina, se ha calcular el sueldo proporcionalmente para los meses parciales resultantes.

De todas formas, si el encargado de la nómina de la unidad organizativa se modifica, esto no tiene ningún efecto sobre el cálculo de la nómina.

Al enlazar un acoplamiento y una proyección, se puede leer los datos maestros para un mes parcial específico.

### ***0.0.0 Tablas de control dependientes del tiempo***

Los datos del infotipo están generalmente codificados como una leyenda (por ejemplo, infotipo *P0006*, tipo de dirección 1 = residencia permanente) para permitir una entrada rápida y un almacenaje que ahorre espacio. Cuando se procesa infotipos, los textos o atributos de las leyendas se leen desde las tablas de control importantes.

En muchas tablas de control, el almacenamiento de los datos depende del tiempo y, por lo tanto, está asignado a un período de validez.

En Recursos Humanos esto se aplica a las siguientes superficies:

- plan de horario de trabajo
- estructura tarifaria
- CC de nómina
- valoración de CC-nóminas
- datos del banco
- posiciones
- códigos del receptor

Cuando se leen los datos para una leyenda del infotipo de las tablas de control dependientes del tiempo, se ha de determinar qué registro es válido para el período de validez especificado.

Si se utiliza una transacción para procesar un infotipo, el sistema lee el registro de la tabla válida en la fecha de inicio.

### ***0.0.0 Generalización de la vista***

Se puede utilizar la vista lógica para tratar y dar salida a los datos según las especificaciones del usuario.

La característica especial de las vistas HR es la dependencia del tiempo de los datos. Los datos personales casi siempre se relacionan con períodos de validez específicos. Una vista HR suministra datos para intervalos de tiempo específicos.

En términos generales, una vista HR es una perspectiva lógica de tablas internas de dependencia interna.

### *O Importar/Exportar fichero en Recursos Humanos*

En esta sección se explica como se accede a los ficheros cluster: *PCL1* y *PCL2*.

#### **0.0 Ficheros PCL1, PCL2, PCL3 y PCL4**

##### **0.0.0.0 Información de los ficheros**

El fichero *PCL1* es la base de datos de las áreas funcionales para HR. Contiene información que se origina en la entrada de tiempos: por ejemplo, hojas de salario de incentivos o textos adicionales de infotipo.

El fichero *PCL2* contiene información derivada, como los resultados del cálculo de la nómina. También contiene todos los esquemas del cálculo de la nómina generados.

El fichero *PCL3* contiene datos de candidatos.

El fichero *PCL4* contiene los documentos de modificación para datos maestros y contratación de personal para HR.

La estructura de los ficheros *PCLn* corresponde a la del fichero *INDX*, con el que deberíamos estar familiarizado gracias a su aparición en otras aplicaciones. La estructura de todos los ficheros *PCLn* (n = 1, 2, 3 y 4) es idéntica.

##### **0.0.0.0 Estructura del fichero**

Como en la mayoría de los ficheros de SAP, el elemento clave que tiene mayor prioridad es el mandante; los datos dentro de un mandante se agrupan según relaciones básicas (campo *PCLn-RELID*). El tipo de relación básica se denomina cluster. Define datos según su tipo. El cluster *RX* contiene los resultados del cálculo de la nómina del país *X*, según la tabla *T500L*, y el cluster *TE* los datos de gastos de viaje.

La estructura de *PCLn-SRTFD* depende del cluster y se define en el string del campo de la *CLAVE xx* el cual se define en el include *RPCnxy0*.

##### **0.0.0.0 Convención de nombres**

n = 1, 2, 3 ó 4 (para PCL1, PCL2, PCL3 o PCL4)

xx para el cluster

y = 0 para clusters internacionales

y = lista de países según *T500L* para clusters nacionales

##### **0.0.0.0 Importar/Exportar datos**

Los ficheros import/export *PCLn* se gestionan con los comandos de *IMPORT* y *EXPORT*. Esos comandos almacenan objetos tales como campos, strings de campo o tablas internas en la base de datos, o los lee desde la base de datos. Los datos se leen desde y se escriben hacia la base de datos utilizando una única clave (*CLAVE xx*). Las macroinstrucciones de

## Programación en HR

RMAC *RP-IMP-Cn-xx* y *RP-EXP-Cn-xx* se suministran para importar y exportar datos. Sólo deberían utilizarse esas macroinstrucciones.

### 0.0.0.0 *Clave xx*

El nombre de la clave *xx* depende del cluster.

*RX KEY* SE utiliza para todos los clusters *Rx* y *Xx*. En los demás episodios, la denominación de la clave *xx* se corresponde con la del cluster.

Cluster	Clave xx
RA	CLAVE -RX
B1	CLAVE-B1
G3	CLAVE-G3
XA	CLAVE-RX

## 0.0 Almacenamiento de datos en ficheros *PCLn*

Los datos de las diferentes áreas funcionales de HR se almacenan en clusters de los ficheros *PCLn* ( $n = 1, 2, 3$  ó  $4$ ).

Este grupo de objetos de datos puede consistir en:

- campos internos de report
- strings de campo
- tablas internas

La estructura de los ficheros *PCLn* ofrece un marco para las áreas funcionales individuales. Cada área funcional debe tener un nombre de cluster de dos caracteres (ID de relación). También deberá tener una estructura clave; para esta estructura hay 40 bytes del campo *SRTFD* disponibles.

Cuando se exporta un registro al fichero *PCLn*, el ID del cluster se escribe en el campo *RELID* y el valor clave en el campo *SRTFD*.

La convención para la definición de nombres para includes en la definición de clusters:

*RPCnxy0*  $n = 1, 2, 3$  o  $4$  (para *PCL1*, *PCL2*, *PCL3*, *PCL4*)

*xx* = identificador de cluster

*y* = lista de países

### 0.0.0 Ejemplo de descripción de datos mediante el cluster *RX*

La descripción de datos se almacena en el include *RPC2RX00* de acuerdo con las convenciones para la definición de nombres mencionadas anteriormente.

Disposición de la clave del cluster:

```
Data: BEGIN OF RX-KEY.
```

```
INCLUDE STRUCTURE PC200.
```

```
DATA: END OF RX-KEY.
```

La estructura *PC200* de diccionario contiene los campos *FERNR* (número de personal) y *SEQNO* (número secuencial).

## Programación en HR

La descripción de datos del cluster también contiene otras tablas internas.

Para una lista de clusters de datos disponibles, hay que consultar la descripción de dominio en el diccionario.

### 0.0 Memoria intermedia de los ficheros PCLn

Para reducir la cantidad de accesos a la base de datos al mínimo, los datos de import y export se almacenan en la memoria intermedia de la memoria principal. Las rutinas para la gestión de la memoria intermedia aseguran que los datos exportados se almacenen en los ficheros de *PCLn*.

#### 0.0.0 *Retroactividad resultados cálculo nomia*

En febrero de 1993 se ejecuta una retroactividad para el mes de enero.

FOR PERIOD 199301 IN PERIOD 199302

Se vuelven a calcular los resultados del cálculo de la nómina para enero y a continuación se escriben directamente en la base de datos.

Resultado:

La base de datos contiene ahora los resultados de los siguientes períodos del cálculo de la nómina.

*FOR PERIOD 199301 IN PERIOD 199302*

*FOR PERIOD 199301 IN PERIOD 199301*

A continuación se ejecuta el cálculo de la nómina del mes de febrero.

*FOR PERIOD 199302 IN PERIOD 199302*

Si se produjeran problemas durante el cálculo de la nómina de este período, el registro de febrero no se almacenaría en la base de datos.

Resultado:

El registro actual de enero en la base de datos es:

*FOR PERIOD 199301 IN PERIOD 199302*

Este problema no se produce si utiliza la memoria intermedia, puesto que todos los datos de una transacción se actualizan siempre de forma colectiva. En el ejemplo anterior, el resultado recalculado de enero se almacenaría en la memoria intermedia y, si la ejecución del cálculo de la nómina para febrero se finalizara de forma prematura, no se actualizaría la base de datos.

El registro actual de enero en la base de datos sería, por lo tanto:

*FOR PERIOD 199301 IN PERIOD 199301*

## Programación en HR

### ***0.0.0 Inicio del cálculo de la nómina en el modo de test***

En una ejecución de test, la base de datos no se actualiza. Puesto que los resultados del cálculo de la nómina del período anterior se utilizan como base para el cálculo de los resultados del período siguiente, los resultados del cálculo de la nómina real serían diferentes de los de la ejecución de test, si ésta se ejecutara durante para varios períodos.

La utilización de la memoria intermedia permite el acceso sin problemas a los resultados solicitados del período anterior.

¿Qué se necesita para exportar/importar datos a/de los ficheros PCLn mediante la memoria intermedia?

Los siguientes includes contienen la descripción de los datos de la memoria intermedia.

*RPPFXD00* → Contiene el área común del *BUFFER*.

*RPPFXD10*

Estos *includes* se tiene que poner en el report que utilicemos para escribir o leer datos de la base de datos.

Es necesario también el include *RPPFXM00*, que contiene las rutinas para la gestión de la memoria intermedia.

Las macroinstrucciones para el import y el export de datos deberán seguir la siguiente convención para la definición de nombres:

RP-aaa-Cn-xy

donde:

aaa = IMP/EXP, n=1 para PCL1, 2 para PCL2, 3 para PCL3, 4 o PCL4

y xy = nombre del cluster.

Esto garantiza la consistencia entre el export y el import de datos y también que todos los objetos exportados se importen de nuevo.

### ***0.0.0.0 Export mediante la memoria intermedia de datos***

Cuando se utilizan macroinstrucciones para el export, los registros no se registran directamente en la base de datos, sino en la memoria intermedia de la memoria principal. Cuando se ha completado la ejecución del programa, los registros de la memoria intermedia se almacenan en la base de datos PCLn correspondiente.

### ***0.0.0.0 Import mediante la memoria intermedia de datos***

Cuando se utilizan macroinstrucciones para el import, los registros no se leen directamente desde los *PCLn*. En su lugar, el sistema verifica el directorio de la memoria intermedia para determinar si se encuentra disponible en la memoria principal algún registro con la misma clave. De no ser así, se carga el registro desde el *PCLn* a la memoria intermedia y se pone a disposición desde la memoria intermedia al report.

## Programación en HR

Si se realiza el import correctamente, se configura el código de *retorno RP-IMP-xy-SUBRC = 0* . Cuando los datos se leen desde la memoria intermedia, el sistema realiza una verificación para la autorización del cluster. Los programas import estándares pueden encontrarse bajo la convención para la definición de nombres *RPCLSTxy* (xy = denominación del cluster).

```
report rptcdmg.
tablas: pernr, pcl1, pcl2.
include rpppxd00. "buffer definitions
data: begin of 'BUFFER' common part.
include rpppxd10. "PCLx buffer
data: end of common part.
data: begin of 'CLUSTER_DIRECTORY' common part.
include rpc2cd00. " "cluster directory definitions
data: end of common part.
include rpc2rdd0.
get pernr.
rp-init-buffer. "inicializamos el buffer
cd-key-pernr = pernr-pernr.
rp-imp-c2-cd. "leemos el cluster CD desde el buffer/DB
perform cd_manager using.... * llamada alternativa: call function rp_evaluation_periods...

rx-key-pernr = pernr-pernr.
rx-key-seqno = rgdir-seqnr.
rp-imp-c2-rd. "Se lee el cluster RD del buffer/DB
...
rp-exp-c2-rd. "actualiza el cluster RD en buffer
...
perform prepare_update using 'V'. "actualización bases de datos (DB)
*-----
* CD subrutinas de gestión y cluster buffer
*-----

include rpcmgr00. "Cluster Directory Manager
include rpppxm00. "Modulo pcl1(2)-buffer
```

### 0.0 Directorio del cluster

Los resultados del cálculo de la nómina se almacenan en el cluster *Rx* del *PCL2*.

La clave del cluster no es nemotécnica; contiene sólo los campos *PERNR* (número de personal) y *SEQNO* (número secuencial).

Para cada resultado del cálculo de la nómina se almacena una entrada de directorio en la tabla interna *RGDIR*. Esta entrada es un número secuencial (*RGDIR-SEQNR*) que identifica unívocamente el resultado del cálculo de la nómina.

## Programación en HR

Los resultados del cálculo de la nómina pueden importarse sólo si la clave del cluster de la nómina contiene el número de personal y el número secuencial.

Los módulos de funciones *RP\_* se usan para leer los resultados del cálculo de nómina desde *PCL2*.

Hay que tener en cuenta que cuando se leen los resultados del cálculo de la nómina, los criterios de selección (área del cálculo de la nómina, período del cálculo de la nómina, etc.) se transfieren al administrador CD.

### 0.0 Interfase

En la siguiente tabla se muestra las denominaciones de campo asignadas al tipo de datos, la máscara de entrada y el significado de cada campo:

Campo(Longitud)	Tipo de datos	Máscara de edición	Significado
PERNR(8)	N		Número de personal
ABKRS(2)	C		Área de nómina PARA
FPPER(6)	C	YYYYPP	Período PARA
FBEGD	D	YYYYMMDD	Período PARA/fecha de inicio
FENDD	D	YYYYMMDD	Período PARA/fecha final
IABKRS(2)	C		Área de nómina EN
IPERM(2)	tipo C		Modificador de período EN
INPER(6)	tipo C	YYYYPP	Período EN
INEND	D	YYYYMMDD	Fecha de fin EN
JUPER(2)	C		Persona jurídica
STRZA(1)	C		Indicador de control
TYPE(4)	C		Tipo de cluster
ACTION(1)	C		Acción
SUBRC	como SY-SUBRC		Código de retorno

#### 0.0.0 Lectura de un registro del cálculo de nómina

Para leer un registro hay que especificar los siguientes parámetros:

- Action (=R)
- Número de personal,
- Fecha de inicio PARA + fecha de fin PARA,
- Modificador de período EN y período EN Y/O indicador de control (si se especifican ambos, deberán coincidir) Y/O persona jurídica.

## Programación en HR

Los parámetros que se devuelven son los siguientes:

- SUBRC  
El sistema configura un código de retorno que define si el READ se ha efectuado correctamente o si sólo se ha encontrado un intervalo.  
Si en el período en cuestión se encuentran disponibles varios registros, sólo se devuelve el primero.  
Destacar que también las fechas de inicio y fin de la orden no necesitan coincidir con el intervalo almacenado.  
Por ejemplo: orden: 15.01. - 30.01.  
almacenado: 1.01. - 30.01.  
devuelto: 1.01. - 30.01.  
SUBRC = 4
- Índice del registro leído
- Si la cabecera de la tabla *RGDIR (DIR2)* está cumplimentada, se conoce el número secuencial. Puede utilizar este número para leer el registro del cálculo de la nómina de la memoria intermedia.

### 0.0.0 Evaluaciones estándar del cálculo de la nómina

Los siguientes reports estándar para la evaluación de los resultados del cálculo de la nómina están disponibles:

Convención para definición de nombres: *RPCLSTxy* donde xy = cluster

Report de ejemplo: *RPCLST99*

### 0.0.0 Módulos de funciones: *RP\_EVALUATION\_PERIODS*

Este módulo de funciones escribe en la tabla EVP (períodos de evaluación) todos los períodos que se han contabilizado en el período seleccionado. En esta tabla se almacenan los datos que describen los resultados actuales y los anteriores.

Los parámetros *import* del módulo serían:

Parámetros	Campo/estructura de referencia
LAST_CALCULATED_DAY	P0003-ABRDT
LAST_DAY_IN_PERIOD	T549Q-ENDDA
RETROCALCULATED_DAY	P0003-RRDAT

- LAST\_CALCULATED\_DAY → En este campo, introduzca el último día que se contabilizó para el empleado; esto es importante si desea ejecutar una evaluación del último período contabilizado. Utilice el valor del campo P0003-ABRDT.
- LAST\_DAY\_IN\_PERIOD → En este campo, introduzca el último día del período de evaluación. Utilice el valor del campo PN/ENDDA, si la evaluación ha de ejecutarse en la base de datos PNP.
- RETROCALCULATED\_DAY → En este campo, introduzca la fecha del período de evaluación en qué debería empezar la ejecución de la retroactividad. El módulo de funciones la utiliza como fecha de inicio; solamente aquellos

## Programación en HR

registros que tengan fechas de inicio más tardías o iguales a este valor se almacenarán en la tabla de resultados. Puede definir esta fecha mediante el módulo RP-SEL-CALC-RPDAT de RMAC, que normalmente se encuentra en todos los reports de evaluación del cálculo de la nómina.

Los parámetros de la tabla son:

Parámetros	Campo/estructura de referencia
DIR	PC231
EVP	PEVPT

- DIR (directorio de resultados del cálculo de la nómina) → Tabla de entrada a la cual se transfiere la tabla RGDIR desde el directorio del cluster.
- EVP (tabla para periodos de evaluación) → Tabla de salida que contiene los números y la información de gestión de los resultados de evaluación. A los resultados actuales se les asigna el indicador A, a los antiguos el indicador P.

Las excepciones son:

- RGDIR\_EMPTY (tabla DIR (RGDIR) vacía) → Esta excepción se configura cuando la tabla DIR no contiene entradas cuando se transfiere. Esto se debe normalmente a un error del programa porque el sistema selecciona sólo aquellos números de personal para los que se ejecuta el cálculo de la nómina durante el período de evaluación.

### 0.0.0.0 *Lectura resultado cálculo nómina usando RP\_EVALUATION\_PERIODS*

Un empleado pertenece al área de cálculo de la nómina en la cual el cálculo de la nómina se realiza mensualmente. El día 15 de enero de 1994, cambia la persona jurídica asignada a este empleado (001P). Por lo tanto, se escriben dos registros de resultados del cálculo de la nómina para el mes de enero de 1994:

- Primer registro de resultados para 1/1.- 14/01/1994
- Segundo registro de resultados para 15/1.- 31/01/1994

De forma retroactiva, el sistema determina que este cambio tuvo lugar el día 20/1/1994 y no el día 15/1/1994.

La retroactividad de febrero ejecutada el 1 de enero produce los siguientes resultados:

- 1er período de 1/1 – 20/01/1994
- 2º período de 20/1 – 31/01/1994
- 3er período de 1/2 – 28/02/1994

Cuando se evalúe el período 02.1994, deberán procesarse estos cinco registros. A continuación el módulo de funciones RP\_EVALUATION\_PERIODS transfiere estos cinco registros a una tabla interna. Los dos primeros están marcados como resultados antiguos, los tres últimos como resultados nuevos.

### 0.0.0 *Indicadores de control*

## Programación en HR

Los indicadores de control son los siguientes:

Indicador de control	Significado
a	actual
p	anterior
o	antiguo

### 0.0.0 Acción

Las acciones a ejecutar son:

Opción de entrada	Acción
r	leer
u	actualizar
i	insertar
d	borrar
s	buscar
l	última búsqueda
a	archivar

### 0.0.0 Códigos de retorno

Los códigos de retorno que el sistema puede retornar es:

Código de retorno	Significado
0	Acceso con éxito (un registro encontrado)
4	Acceso con éxito parcial (varios registros encontrados)
6	Acceso sin éxito (ningún registro encontrado)

### 0.0.0 Lectura de un registro del cálculo de la nómina

Registro almacenado:

Nº sec	forbegda	feinpara	iperm	peren	indc
001	01021991	28021991	01	199102	P
002	01021991	28021991	01	199103	A

Orden incorrecta:

Nº sec	feinpara	feinpara	iperm	peren	indc
	01021991	28021991	01	199102	A

Resultado: Sy-Subrc = 8.

## Programación en HR

Orden correcta:

Nº sec	feinpara	feinpara	iperm	peren	indc
	01021991	28021991			A

Resultado:

Nº sec	feinpara	feinpara	iperm	peren	indc
002	01021991	28021991	01	199103	A

u orden correcta

Nº sec	feinpara	feinpara	iperm	peren	indc
	01021991	28021991	01	199102	

Resultado:

Nº sec	feinpara	feinpara	iperm	peren	indc
001	01021991	28021991	01	199102	P

### *0.0.0 Importar resultado cálculo nómina de cluster RX*

```
PERFORM CD_MANAGER USING  
'00003355' " Número de personal  
SPACE  
SPACE  
'19930501' "FOR periodo/fecha de inicio  
'19930531' "FOR periodo/fecha final"  
SPACE  
SPACE  
SPACE  
SPACE  
SPACE  
'A' "Registro actual  
SPACE  
'R' "Read record"  
SUBRC. "Return code
```

En el campo *SUBRC* se configura un código de retorno, que indica si se han encontrado uno, varios o ningún registro de datos.

Los resultados del cálculo de la nómina podrán importarse ahora mediante el *módulo RP-IMP-C2-Rx de RMAC*.

### *0 Comandos específicos*

#### **0.0 Módulos de funciones en HR**

Los módulos de funciones son módulos de programas que tienen una interfase definida y permiten la verificación del tipo de parámetros.

## Programación en HR

Se gestionan con la transacción SE37 y se combinan con grupos de funciones según criterios relevantes. Puede acceder a esta transacción seleccionando *Herramientas* → *Workbench ABAP/4* → *Biblioteca funciones*.

Los grupos de funciones de HR cumplen la convención para definición de nombres *Rpxx* o *Hrxx*; *xx* es un indicador que el usuario puede definir.

Puede utilizar el comando de editor *SHOW FUNCTION \** para pasar del proceso de reports a la visualización del módulo de funciones.

### 0.0 Módulos de macroinstrucciones

Al igual que los módulos de subprogramas y funciones, los módulos de macroinstrucciones son medios para la presentación de programas en forma modular. Los módulos de macroinstrucciones se utilizan con frecuencia en HR.

#### 0.0.0 Definición y llamada

Se presentan dos opciones:

- Las macros pueden definirse en reports o includes mediante el comando "definir" de ABAP/4. Si un report utiliza una macroinstrucción que está definida en un include, este include deberá integrarse en el report. Si se modifica una macroinstrucción definida con el comando "definir", cada report que utilice esa macroinstrucción será generará de nuevo automáticamente la próxima vez que se ejecute. En el include DBPNPMAC están definidas las macros ha utilizar con la base de datos lógica PNP.
- Las macroinstrucciones también pueden definirse como macroinstrucciones TRMAC. El código fuente de estos módulos se almacena en la sección de funciones de la tabla de control TRMAC. Este código se agrupa bajo un nombre específico en la clave de tabla. Según las convenciones, las dos primeras letras de la denominación han de hacer referencia a la aplicación. El resto de la denominación se puede definir libremente.

Los módulos de macroinstrucciones específicas del cliente deberán empezar por un carácter especial. Las macroinstrucciones definidas en la tabla de control TRMAC pueden utilizarse por todos los reports

Es importante: Cuando se modifica una macroinstrucción en la tabla TRMAC, los reports que utilizan esta macroinstrucción no se generan de nuevo automáticamente. Deberá generarlos de nuevo manualmente.

### 0 Módulos RMAC estándar en HR

El mismo tipo de órdenes que pueden utilizarse en diversos reports puede programarse con las siguientes herramientas para desarrollo:

- Módulos de funciones (transacción SE37)
- Módulos RMAC (Tabla TRMAC)

Los siguientes módulos RMAC se utilizan normalmente en los reports HR:

## Programación en HR

### 0.0 Definición de datos

#### 0.0.0 *RP-LOWDATE-HIGHDATE*

Definición fecha anterior y posterior

##### 0.0.0.0 *Función*

La definición de las constantes:

LOW-DATE TYPE D VALUE 18000101

y

HIGH-DATE TYPE D VALUE 99991231

Representa la primera y/o la última fecha en HR. Representan los primeros y los últimos puntos en la línea de tiempo y no constituyen puntos reales en el tiempo. Estas fechas sólo pueden utilizarse en comparaciones, no en cálculos.

##### 0.0.0.0 *Parámetros*

Ninguno

##### 0.0.0.0 *Verificación*

Ninguna

##### 0.0.0.0 *Condiciones previas*

Ninguna

##### 0.0.0.0 *Utilización*

En programas HR, como constantes de lectura para comparaciones.

RP-LOWDATE-HIGHDATE.

...

GET PERNR.

...

IF Pnnnn-ENDDA EQ HIGH-DATE. "til the 12th of never

### 0.0 Evento *INITIALIZATION*

## Programación en HR

### ***0.0.0 RP-SEL-EIN-AUS-INIT***

Sugerencia: Omisión de exempleados.

#### ***0.0.0.0 Función***

La pantalla propuesta de la orden informa al usuario del report que seleccione únicamente el personal actual. Es decir, el procesador de la base de datos no pasa a ningún empleado con un 0 en la tabla horaria (PN-BEGPS,PN-ENDPS) del campo *P0000-STAT2*, infotipo *Medidas* (0000), al report por medio del procesador de la base de datos.

#### ***0.0.0.0 Parámetros***

Ninguno.

#### ***0.0.0.0 Verificación***

Ninguna.

#### ***0.0.0.0 Condiciones previas***

Ninguna.

#### ***0.0.0.0 Utilización***

En la sección *INITIALIZATION* de los programas de la base de datos PNP

*INITIALIZATION.*  
*RP-SEL-EIN-AUS-INIT*  
*START-OF-SELECTION.*

#### ***0.0.0.0 Observaciones***

- *RP-SEL-EIN-AUS-INIT* sugiere un valor de propuesta que el usuario del report puede modificar en la pantalla de selección en la fecha de inicio del report, *mientras RP-SEL-EIN-AUS* no se puede seleccionar (como regla).
- *RP-SEL-EIN-AUS-INIT* permite un tratamiento relativamente rápido por parte del procesador de la base de datos, mientras que en *RP-SEL-EIN-AUS* esto se realiza comparativamente tarde (rendimiento).
- *RP-SEL-EIN-AUS-INIT* hace referencia al intervalo de tiempo (PN-BEGPS, PN-ENDPS), mientras que *RP-SEL-EIN-AUS* trabaja con un periodo de tiempo (PN-BEGDA, PN-ENDDA). En la mayoría de los casos estos dos intervalos son idénticos, pero no siempre.
- La selección que inicia *RP-SEL-EIN-AUS-INIT* a partir del procesador de la base de datos está *simultáneamente* enlazada con los otros criterios de selección de

## Programación en HR

la base de datos, mientras que la selección *RP-SEL-EIN-AUS* vuelve a basarse en todo el intervalo de tiempo. Con esto pueden realizarse diferentes selecciones.

### ***0.0.0 RP-INIT-WITH-SYDATUM***

Sugerencia: Período de selección = fecha actual

#### ***0.0.0 Función***

La pantalla propuesta de la orden informa al usuario del report que utilice la fecha del día (SY-DATUM,SY-DATUM) para los períodos de selección de fecha y de personas (PNPBEGDA, PNPENDDA) y (PNPBEGPS,PNPENDPS) respectivamente.

No vuelva a utilizarlo. En lugar de esto, escriba lo siguiente en la sección *INITIALIZATION* del programa:

PNPTIMED = D

(por ejemplo, véase RPLMITOO)

### ***0.0 Evento START-OF-SELECTION***

#### ***0.0.0 RP-SET-NAME-FORMAT***

Formato del report para la edición de nombre HR.

##### ***0.0.0.0 Función***

La edición de la denominación HR se realiza utilizando el módulo de funciones DENOMINACIÓN-TRATAMIENTO-RP en la selección (GET PERNR). Esta rutina edita el nombre de acuerdo con las entradas de la tabla T522N (Edición de denominación HR). Así como cada edición de denominación HR puede depender del programa, el módulo RP-SET NAME-FORMAT está directamente enlazado con la edición de denominación HR en la sección *START-OF-SELECTION*. El módulo RMAC determina el formato de report correspondiente en la tabla T522F (Formato para edición de denominación HR). Si no se ha realizado ninguna entrada en esta tabla para el report HR correspondiente, el formato de report se configura en 01 (DEFAULT).

##### ***0.0.0.0 Parámetros***

Entrada	Ninguno
Salida	Formato de report de T522F (por defecto: 01) para T522N

##### ***0.0.0.0 Verificación***

Tabla T522F (Formato para edición de la denominación HR)

## Programación en HR

### 0.0.0.0 Utilización

Este módulo RMAC se utiliza en cualquier report HR general con una cadena de caracteres del nombre de empleados en la salida, por ejemplo, en los reports estándares HR RPLMIT00, RPLTRF00, RPLTRF10 etc.

### 0.0.0 RP-FETCH-ALTER-PERID

Verificación número de identificación personal desde P0002

### 0.0.0.0 Función

Incluye los números de identificación de personal desde el infotipo *Datos de personal* (0002) (P0002-PERID) en todas las evaluaciones estándar HR generales. En algunas versiones nacionales HR (como por ejemplo, Dinamarca, Canadá, EE.UU...), el campo P0002-PERID contiene caracteres adicionales de identificación de personal imprescindibles para estos países.

La transacción PEO3 (características operativas/parámetros del sistema HR) activa o desactiva esta línea separada de salida (Característica: PERNO). Existen dos especificaciones disponibles:

PERNR	La salida muestra únicamente el número de personal.
PERID	La salida del campo P0002-PERID también se muestra en una línea aparte (primera línea).

Para verificar el *número de personal adicional* del infotipo *Datos personales* (0002), utilice el módulo de funciones HR, RP\_FETCH\_ALTERNATE-PERNR (transacción: SE37).

### 0.0.0.0 Parámetros

Entrada	1. PME04-MOLGA	Valor relacionado con el usuario de T531A
Salida	1. \$\$\$W-ALTER-PERID 2. \$RETCODE	Marcador alternativo PERID (YES/NO) Código de retorno de RE549B

### 0.0.0.0 Verificación

La transacción PEO3 (características operativas/parámetros del sistema HR)

Característica: PERNO

Var. clave de tabla: MOLGA (valor referente al usuario de T513A)

### 0.0.0.0 Utilización

Este módulo RMAC se utiliza en todos los reports HR generales que imprimen o muestran números de personal (PERNR). Sólo se utiliza con el módulo RMAC, RP-WRITE-ALTER-PERID ! Los reports estándares HR, como RPLMIT00, RPLTRF00 RPLTRF10, etc. son ejemplos típicos.

## Programación en HR

### ***0.0.0 RP-SET-KEY-DATE***

Configurar la fecha clave

#### ***0.0.0.0 Función***

Si el usuario no introduce un período de selección de datos en la pantalla de solicitud del banco de datos, las fechas de inicio y fin se configuran en LOW-DATE o HIGH-DATE, respectivamente. Si no se introduce ningún período de selección de datos, para facilitar la selección de datos del usuario, se configura la fecha del día como fecha clave. Para las evaluaciones basadas en fechas clave, deben coincidir las fechas de inicio y fin. Si no se introduce nada o si no se utiliza este módulo, la selección resultante será ilógica.

#### ***0.0.0.0 Parámetros***

Ninguno.

#### ***0.0.0.0 Verificación***

Ninguna.

#### ***0.0.0.0 Condiciones previas***

El módulo RMAC *RP-LOWDATE-HIGHDATE* tiene que estar configurado en la sección de declaración de datos de un report.

#### ***0.0.0.0 Utilización***

Este módulo RMAC se utiliza en los reports HR que realizan evaluaciones de la base de datos de una fecha clave. Puede utilizarse en combinación con el módulo RMAC *RP-STICHTAG*.

RP-LOWDATE-HIGHDATE.

...

START-OF-SELECTION.  
RP-SET-KEY-DATE.

...

END-OF-SELECTION.  
TOP-OF-PAGE.  
RP-STICHTAG.

### **0.0 Selección GET PERNR**

#### ***0.0.0 RP-SEL-EIN-AUS***

## Programación en HR

Omisión de ex empleados.

### **0.0.0.0 Función**

Muchos reports no tienen en cuenta a los ex-empleados. Para RP *SEL EIN AUS*, un *ex empleado* es un empleado con un 0 en todos los registros del infotipo *Medidas* (0000) en el campo *P0000-STAT2* y en el intervalo de tiempo (PN-BEGDA,PN-ENDDA). En *GET PERNR*, después de *RP-SEL-EIN-AUS*, estas personas ya no se incluyen en los procesamientos (declaración en lenguaje de programación ABAP: REJECT).

No vuelva a utilizar más este módulo. En su lugar, debería utilizarse *RP-SEL-EIN-AUS-INIT*.

### **0.0.0 RP-SEL-CALC**

Selección de empleados a quienes se les ha calculado la nómina.

#### **0.0.0.0 Función**

Este módulo sólo selecciona aquellos números de personal donde se ha terminado la nómina para el período de cálculo de nómina seleccionado. Por lo tanto debería utilizarse en todas las evaluaciones de nómina.

#### **0.0.0.0 Parámetros**

Ninguno.

#### **0.0.0.0 Verificación**

Ninguna.

#### **0.0.0.0 Condiciones previas**

El módulo RMAC *RP-READ-PAYROLL-DIR* tiene que estar presente al final del programa.

#### **0.0.0.0 Utilización**

El módulo RMAC se utiliza en todos los reports HR que calculan los resultados del cálculo de la nómina. Debe utilizarse en combinación con el módulo RMAC *RP-READ-PAYROLL-DIR*.

GET PERNR.  
RP-SEL-CALC.  
...  
(rutinas)  
RP-READ-PAYROLL-DIR.

### **0.0.0 RP-WRITE-ALTER-PERID**

Número de identificación de personal P0002.

## Programación en HR

### 0.0.0.0 Función

Inclusión de los números de identificación de personal desde el infotipo *Datos personales* (0002) (*P0002-PERID*) en todas las evaluaciones estándares HR generales.

En algunas versiones nacionales de HR, el campo *P0002-PERID* contiene una identificación de personal extraordinaria (por ejemplo, Dinamarca, Canadá, EE.UU.), que resulta imprescindible para estos países. El "número de personal adicional" se muestra en la evaluación estándar HR correspondiente, en una línea de salida aparte (una línea antes del número de personal HR). Esta línea de salida separada puede activarse y desactivarse por medio de la transacción PEO3 (características operativas/parámetros del sistema HR). (Característica: PERNO) Existen dos especificaciones disponibles:

PERNR	La línea de salida está limitada al número de personal
PERID	El campo PERÍODO-P0002 se visualiza en una línea separada (primera línea).

El *número de personal adicional* del infotipo *Datos personales* (0002) puede verificarse llamando el módulo de funciones HR, RP-FETCH-ALTERNATE-PERNR (transacción: SE37).

### 0.0.0.0 Parámetros

Entrada	1. \$\$\$SW-ALTER-PERID	Marcador alternativo PERÍODO (YES/NO)
	2. P0002	Infotipo de datos personales (0002)
	3. Opción de impresión	x directo en el módulo RMAC o SPACE indirecto en el report del usuario
Salida	1. PERID	Identificación de personal del infotipo de datos personales (0002)
	2. \$LENGTH	Longitud de identificación de personal

### 0.0.0.0 Verificación

La transacción PEO3 (características operativas/parámetros del sistema HR)  
Característica: PERNO  
Clave de tabla variantes: MOLGA (valor dependiente de usuario de *T513A*)

### 0.0.0.0 Utilización

El módulo RMAC se utiliza en todos los reports HR generales que imprimen los números de personal (PERNR). Sólo se utiliza en combinación con el módulo *RMAC RP-FETCH-ALTER-PERID* I. (por ejemplo, los reports estándar HR, como *RPLMIT00*, *RPLTRFO0*, *RPLTRF10*, etc.)

### 0.0.0 RP-MAKE-ADDRESS

Generación destino HR según reclamación regulación mailing.

### 0.0.0 Función

## Programación en HR

La interfase HR trata la dirección de un empleado para cumplir con las reglamentaciones del mailing internacional. Además de la dirección general, se dispone de una dirección breve fija en un string para propósitos especiales. Se compone de:

- Apellido, nombre de pila y título
- /
- ZIP/código postal, ciudad

En la edición de direcciones, el módulo de funciones de SAP (transacción: SE37) ADRESSAUFBEREITUNG. (SAP estándar)

### **0.0.0.0 Parámetros**

Entrada	1. Infotipo P0001	Asignación organizacional. (P0001-BUKRS para T001F)
	2. Infotipo P0002	Datos personales (P0002-ANRED para tratamiento)
	3. Infotipo P0006	Direcciones
	4. EDIT-NAME	Nombre de acuerdo con las normativas nacionales (utilizando RP-EDIT-NAME)
	5. Número de líneas en la ventana de direcciones	
Salida	1. ADRS-ZEILO	Línea de dirección 1
	...	...
	10. ADRS-ZEIL9	Línea de dirección 10
	11. ADRS-LINEK	Línea de dirección breve

### **0.0.0.0 Verificación**

Ninguno.

### **0.0.0.0 Utilización**

El módulo RMAC se utiliza en cualquier report HR en el que se necesita visualizar o imprimir la dirección completa o breve de un empleado (por ejemplo, el report de imprimir etiqueta RP *RPLADROO*).

### **0.0.0 RP-EDIT-ADDRESS**

Edición directa de HR según normativa de correo.

### **0.0.0.0 Función**

Salida de la dirección de un empleado de acuerdo con las normativas del correo internacional. Las coordenadas de la ventana de direcciones (línea y columna) pueden definirse en la tabla T001F (parámetros de la sociedad).

- ZEIVN ("línea de" en la ventana de direcciones)
- SPLVN ("columna de" en la ventana de direcciones)
- ZEIBS ("línea a" en la ventana de direcciones)
- SPLBS ("columna a" en la ventana de direcciones)

## Programación en HR

La salida de la ventana de direcciones puede determinarse también de forma individual en cada report (como parámetros de report, etc.). Además de la dirección general, se dispone de una dirección breve fija en un string para propósitos especiales.

- Apellido, nombre de pila y título
- /
- ZIP/código postal, ciudad

Al generar direcciones se utiliza el módulo HR RMAC, *RP-MAKE-ADDRESS*.

### **0.0.0.0 Parámetros**

Entrada	1. Infotipo P0001	Asignación organizacional (P0001-BUKRS para T001F)
	2. Infotipo P0002	Datos personales (P0002-ANRED para tratamiento)
	3. Infotipo P0006	Direcciones
	4. EDIT-NAME	Nombre de acuerdo con las normativas nacionales (utilizando RP-EDIT-NAME)
	5. Número de líneas en la ventana de direcciones	
	6. "Línea de" en una ventana de direcciones	
	7. "Columna de" en una ventana de direcciones	
	8. "Línea a" en una ventana de direcciones	
	9. "Columna a" en una ventana de direcciones	
Salida	1. ADRS-ZEIL0	Línea de dirección 1
	...	...
	10. ADRS-ZEIL9	Línea de dirección 10
	11. ADRS-LINEK	Línea de dirección breve

### **0.0.0.0 Verificación**

Tabla 001F (parámetros de sociedad) para coordenadas.

### **0.0.0.0 Utilización**

Este módulo RMAC se utiliza en reports HR que imprimen una totalidad de empleado o una dirección breve. (por ejemplo, los reports de carta estándar de HR *RPBDSG00*, *RPBTAR00*, *RPBLET\**, etc.)

### **0.0.0 RP-READ-ALL-TIME-ITY**

Lectura de todos los infotipos de tiempos en las tablas horarias.

## Programación en HR

### 0.0.0.0 *Función*

Todos los registros de tiempos (infotipos 2000-2999) sobre la persona en procesamiento durante un periodo de tiempo determinado se introducen en las correspondientes tablas de infotipos internas (definidas con la ayuda de la palabra clave INFOTYPES).

### 0.0.0.0 *Parámetros*

*HR-READ-ALL-TIME-ITY* ini fin

Entrada	1. Inicio	Fecha de inicio de validez del intervalo de tiempo
	2. Fin	Fecha final de la validez del intervalo de tiempo
Salida	1. PNP-SW-AUTH-SKIPPED-RECORD	0, si la verificación de autorización HR no puede guardar un registro por falta de autorización 1, si la verificación de autorización HR ha guardado al menos un registro por falta de autorización
	2. Tablas internas de infotipos de tiempos con todos los registros del periodo de tiempo relevante para el que tiene autorización el usuario	

### 0.0.0.0 *Verificación*

Ninguna.

### 0.0.0.0 *Condiciones previas*

- La fecha de inicio de validez del periodo de tiempo debe coincidir o anteponerse a la fecha de fin de validez.
- Las fechas de inicio y fin de validez constituyen especificaciones de fecha correctas (preferiblemente del tipo DATE).

### 0.0.0.0 *Utilización*

Únicamente en reports de base de datos PNP, en GET PERNR, porque el número de personal para el que se leerán los datos procede del campo *PERNR-PERNR*, mientras que el campo que se utiliza es PNP-SW-AUTH-SKIPPED-RECORD.

*DATA: BEGDA LIKE P2001-BEGDA, ENDDA LIKE P2001-ENDDA.*

*INFOTYPES: 0000, 0001, 0002,...*

*2001 MODE N, 2002 MODE N,...*

*GET PERNR.*

*BEGDA = '19900101'!*

*ENDDA = '19900131'!*

*RP-READ-ALL-TIME-ITY BEGDA ENDDA.*

*IF PNP-SW-AUTH-SKIPPED-RECORD NE '0'!*

*WRITE: / 'Autorización por falta de datos de tiempos'!*

*WRITE: / 'Por numero de personal', PERNR-PERNR. REJECT.*

*ENDIF.*

## Programación en HR

### 0.0.0.0 Observaciones

Por ejemplo, el módulo RMAC puede utilizarse cuando los infotipos de tiempos se han definido originalmente en MODE N, ya que es posible que los datos de tiempos (de LOW-DATE a HIGH-DATE) no se adaptan a la memoria intermedia. Sin embargo, ahora se leen a intervalos más breves (como ocurre, por ejemplo, en RPCALCx0 con los períodos del cálculo de la nómina).

### 0.0.0 RP-EDIT-NAME

Edición de nombre HR según normativa nacionales.

### 0.0.0.0 Función

Este módulo RMAC trata nombres en todas las evaluaciones estándar HR según la normativa nacional de un determinado país. La edición de denominaciones se realiza de acuerdo con las entradas de la tabla T522N (edición de nombre HR). La edición de la denominación HR también depende de:

- MOLGA (sociedad, división de personal, subdivisión de personal) de T001P
- El formato de report de la tabla 522F (por defecto: '01')
- El origen de los datos
  - Infotipo *Datos personales* (0002)
  - Infotipo *Familia/pers.refer.* (0021)
- El indicador para edición de nombre del infotipo *Datos personales* (0002) (KNZNM) (únicamente es válido si los datos provienen del infotipo *Datos personales* (0002)).

Si la edición de la denominación de la evaluación estándar HR corresponde a la que se encuentra en los datos maestros (formato: 01), la cadena de caracteres del nombre almacenada en los datos maestros se utiliza en el infotipo P0001 (*Asignación organiz./P0001-ENAME*). El módulo de función RP\_EDIT\_NAME (transacción: SE37) se utiliza para editar nombres.

### 0.0.0.0 Parámetros

Entrada	1.	Infotipo P0001	<i>Asignación organiz.</i> (P0001-ENAME)
	2.	Infotipo P0002	Datos personales o
		Infotipo P00021	Familia/pers.refer.
	3.	MOLGA	(T001P-MOLGA)
	4.	LANGU	Idioma para el modo de la dirección
			SPACE idioma de trabajo de SAP SY-LANGU
			x idioma del receptor P0002-SPRSL
	5.	\$\$FORMAT	Formato de nombre para la edición
Salida	1.	\$EDIT-NAME	String de la cadena de caracteres de la denominación editada para transferencia
	2.	\$RET-CODE	Código de retorno de edición de nombre

## Programación en HR

### **0.0.0.0 Verificación**

- Tabla T522F (Formato para edición de la denominación HR)
- Tabla T522F (Edición de nombre HR)

### **0.0.0.0 Utilización**

Este módulo RMAC se utiliza en todos los reports HR que imprimen el nombre completo de un empleado. Sólo se utiliza en combinación con el módulo *RMAC RP-FETCH-ALTER-PERID*!. (por ejemplo, reports de cartas estándares de HR *RPBDSGOO*, *RPBTAROO*, *RPBLET\** etc.)

### **0.0.0 RP-INIT-BUFFER**

Inicializa memoria intermedia para acceder a los ficheros *PCL1* AL *PCL4*.

### **0.0.0.0 Función**

Inicializa la memoria intermedia de la memoria principal para acceder a los ficheros *PCL1* a *PCL4*.

### **0.0.0.0 Parámetros**

Ninguno.

### **0.0.0.0 Verificación**

Ninguno.

### **0.0.0.0 Condiciones previas**

El siguiente incluye contiene la definición de datos para la memoria intermedia. Éstos deben incluirse en el report que escribe o lee los datos de la base de datos.

- *RPPFXD00x*
- *RPPFXD10*
- *RPPFXM00*

### **0.0.0.0 Utilización**

Inicializa la memoria intermedia de la memoria principal para el acceso a los ficheros *PCLn* (n = 1, 2, 3, 4) y le permite utilizar los ficheros import/export *RP-IMP-Cn-xx* y *RP-EXP-Cn-xx*.

## Programación en HR

### **0.0.0 RP-IMP-Cn-xx**

Módulos import

#### **0.0.0.0 Función**

Los ficheros *PCLn* (n = 1, 2, 3, 4) se utilizan mediante el comando *IMPORT*. Este comando permite leer cualquier objeto de la base de datos, como campos, strings de campos o tablas internas. La base de datos se lee utilizando una clave unívoca (*xx-KEY*).

Hay que tener en cuenta que las macroinstrucciones *RP-IMP-Cn-xx* (n = 1, 2, 3, 4, xx = cluster) pueden utilizarse para importar datos y deben utilizarse en este contexto.

Cuando las macroinstrucciones se utilizan para importar datos, los registros de datos no se leen directamente desde el fichero *PCLn*. En su lugar, el sistema verifica el directorio de la memoria intermedia para ver si la memoria principal contiene ya un registro con la misma clave. Si ése no es el caso, el registro se lee de *PCLn* a la memoria intermedia y se recupera de la memoria intermedia al report.

Si los datos se importan de forma satisfactoria, se fija el código de *retorno RP-IMP-xy-SUBRC* = 0 . Si se utiliza una memoria intermedia para leer los datos, el sistema verifica la autorización del cluster. Los programas import estándares siguen la convención para definir nombres *RPCLSTxy* (xy = nombre del cluster).

#### **0.0.0.0 Parámetros:**

Este módulo no tiene parámetros. Sin embargo, se debe especificar la *xx-Key* apropiada antes de acceder al módulo.

#### **0.0.0.0 Verificación:**

Ninguna.

#### **0.0.0.0 Condiciones previas:**

La memoria intermedia principal se debe inicializar primero utilizando el módulo de RMAC *RP-INIT-BUFFER*.

El módulo *RP-IMP-C2-RA* importa los resultados de la nómina almacenados en el fichero PCL2 (cluster RA).

### **0.0.0 RP-EXP-Cn-xx**

Módulos export

#### **0.0.0.0 Función**

Los ficheros *PCLn* se gestionan utilizando la sentencia *EXPORT*. Este comando permite almacenar cualquier objeto en la base de datos, como campos, strings de campos o tablas internas. La base de datos se lee utilizando una clave unívoca (*xx-KEY*). Hay que tener en cuenta que las macroinstrucciones *RP-EXP-Cn-xx* (n = 1, 2, 3, 4, xx = cluster) pueden utilizarse para exportar datos y deben utilizarse en este contexto.

## Programación en HR

Cuando se utilizan macroinstrucciones para exportar datos, los registros de datos no se registran directamente en la base de datos. En vez de eso, se registran en una memoria intermedia de la memoria principal. Al final del programa, los registros almacenados en la memoria intermedia se graban en la base de datos *PCLn* correspondiente.

Con la rutina *Update*, también puede resultar obligatorio grabar los registros almacenados de una memoria intermedia.

### **0.0.0.0 Parámetros**

Este módulo no tiene parámetros. Sin embargo, se debe especificar la *xx-Key* apropiada antes de acceder al módulo.

### **0.0.0.0 Verificación**

Ninguna.

### **0.0.0.0 Condiciones previas:**

La memoria intermedia principal se debe inicializar primero utilizando el módulo de *RMAC RP-INIT-BUFFER*.

El módulo *RP-EXP-C2-RA* registra los resultados de la nómina de la memoria a la memoria intermedia principal (fichero *PCL2*, cluster *RA*).

### **0.0.0 RP-REF-Cn-xx**

Cómo borrar resultados nómina de memoria.

### **0.0.0.0 Función**

Esta función borra los datos almacenados en la memoria del cluster *xx*, fichero *PCLn*. La estructura se guarda en la memoria.

Los datos no se borran de la memoria intermedia de la memoria principal ni de la base de datos.

### **0.0.0.0 Parámetros**

Este módulo no tiene parámetros. Sin embargo, se debe especificar la *xx-Key* apropiada, antes de acceder al módulo.

### **0.0.0.0 Verificación**

Ninguna.

### **0.0.0.0 Condiciones previas**

Ninguna.

## Programación en HR

El módulo RP-REF-C2-RA borra los resultados de la nómina almacenados en la memoria del cluster RA, fichero PCL2.

### 0.0 Evento TOP-OF-PAGE

#### 0.0.0 *RP-STICHTAG*

Salida de fecha clave en la lista *TOP-OF-PAGE*.

##### 0.0.0.0 *Función*

Visualiza la fecha clave (PN/ENDDA) en la línea superior de cada página de una lista HR estándar.

##### 0.0.0.0 *Parámetros*

Ninguno.

##### 0.0.0.0 *Verificación*

Ninguna.

##### 0.0.0.0 *Condiciones previas*

Ninguna.

##### 0.0.0.0 *Utilización*

En cualquier programa de la base de datos lógica PNP, en TOP-OF-PAGE

```
TOP-OF-PAGE.  
IF PN/BEGDA EQ PN/ENDDA.  
RP-STICHTAG.  
ELSE.  
RP-ZEITRAUM.  
ENDIF.
```

#### 0.0.0 *RP-ZEITRAUM*

Salida tiempo en lista *TOP-OF-PAGE*.

##### 0.0.0.0 *Función*

Muestra el período de tiempo (PN/BEGDA,PN/ENDDA) en la línea superior de cada página en una lista HR estándar.

##### 0.0.0.0 *Parámetros*

Ninguno.

## Programación en HR

### *0.0.0.0 Verificación*

Ninguna.

### *0.0.0.0 Condiciones previas*

Ninguna.

### *0.0.0.0 Utilización*

En cualquier programa de la base de datos lógica *PNP* en *TOP-OF-PAGE*.

```
...  
TOP-OF-PAGE.  
IF PN/BEGDA EQ PN/ENDDA.  
RP-STICHTAG.  
ELSE.  
RP-ZEITRAUM.  
ENDIF.
```

## **0.0 En cualquier punto**

### *0.0.0 Obtención de datos*

#### *0.0.0.0 RP-READ-INFOTYPE*

Lectura infotipo persona en un período de tiempo.

##### *0.0.0.0.0 Función*

Se introducen todos los registros de infotipos de una persona durante un período de tiempo determinado en una tabla de infotipos interna (definida mediante la declaración del lenguaje de programación INFOTYPES).

##### *0.0.0.0.0 Parámetros*

Entrada	1.	Número de personal de la persona solicitada
	2.	Número del infotipo solicitado
	3.	Nombre de la tabla de infotipos interna
	4.	Fecha de inicio de validez del intervalo de tiempo
	5.	Fecha de fin de validez del intervalo de tiempo
Salida	1.	PNP-SW-FOUND = 0, si no existe ningún registro en el set de datos que sea adecuado
		PNP-SW-FOUND = 1, si existe como mínimo un registro en el set de datos que sea adecuado
	2.	PNP-SW-AUTH-SKIPPED-RECORD = 0, si la verificación de autorización de HR no

## Programación en HR

		ha retenido ningún registro por falta de autorización
		PNP-SW-AUTH-SKIPPED-RECORD = 1, si la verificación de autorización HR ha retenido, como mínimo, un registro por falta de autorización
	3.	Tabla de infotipos interna, con todos los registros que coincidan con aquéllos para los que el usuario tiene autorización (también puede darse el caso de que esta tabla esté vacía).

### 7.6.1.1.3 Verificación

Ninguna.

### 7.6.1.1.4 Condición previa

- Es preciso que la fecha de inicio de validez de la tabla horaria sea anterior o coincida con la fecha de fin de validez.
- El inicio y el fin de validez constituyen datos de fecha correctos (preferiblemente del tipo DATE).
- La tabla de infotipos debe coincidir con el número de infotipo.
- El programa que utiliza el módulo RMAC debe contener el elemento de *INCLUDE, DBFNPCOM* (esto es lo que ocurre automáticamente con todos los reports LDP PNP).

### 7.6.1.1.5 Utilización

- En cualquier punto de cualquier programa (inclusive la base de datos PNP).
- También puede utilizarse en módulos de funciones.



```

INFOTYPES: 0001.
RP-LOWDATE-HIGHDATE.
DATA: PERNR LIKE P0001-PERNR.
DATA: BEGDA LIKE P0001-BEGDA, ENDDA LIKE P0001-ENDDA.
PERNR = '12345678'.
BEGDA = LOW-DATE + 15.
ENDDA = HIGH-DATE - 5.
RP-READ-INFOTYPE PERNR 0001 P0001 BEGDA ENDDA.
IF PNP-SW-AUT-SKIPPED-RECORD EQ '1'.
WRITE: / 'Insufficient authorization'. STOP.
ENDIF.
IF PNP-SW-FOUND EQ '0'.
WRITE: / 'Infotype 0001 lacking'. STOP.
ENDIF.

```

### 0.0.0.0.0 Observaciones

En la base de datos PNP, los infotipos deberían leerse por lo general utilizando GET PERNR. La excepción es utilizar RP-READ-INFOTYPE.

## Programación en HR

### 0.0.0.0 *RP-READ-T001P*

Lectura de la tabla *T001P*.

#### 0.0.0.0 *Función*

Mediante la clave completa, se accede a la tabla T001P. El acceso se graba en la memoria intermedia. El resultado se visualiza en el título de la tabla (interna) T001P.

#### 0.0.0.0 *Parámetros*

*RP-READ-T001P WERKS BTRTL message*

Entrada	1.	División de personal
	2.	Subdivisión de personal
	3.	Mensaje: reacción si no existe ninguna entrada:
		X el report finaliza con el mensaje D001: <i>Ninguna entrada en la clave de tabla T001P...</i>
		SPACE SY-SUBRC se fija en 4.
Salida	1.	Título en la tabla (interna) T001P
	2.	SY-SUBRC 0 se ha leído con éxito
		4 el registro no existe

#### 0.0.0.0 *Utilización*

En cualquier punto de cualquier programa (inclusive la base de datos PNP).

```
PROVIDE *FROM P0001...  
RP-READ-T001P P0001-WERKS P0001-BTRTL SPACE.  
IF SY-SUBRC NE 0.  
own error handling  
ENDIF.  
ENDPROVIDE.
```

### 0.0.0.0 *RP-READ-T5DOP*

Lectura de la tabla *T5DOP*.

#### 0.0.0.0.0 *Función*

Se accede a la tabla *T5DOP* escribiendo la clave completa. El acceso se graba en la memoria intermedia. El resultado se visualiza en el título de la tabla (interna) *T5DOP*.

#### 0.0.0.0.0 *Parámetros*

*RP-READ-T5DOP WERKS BTRTL message*

Entrada	1.	División de personal
	2.	Subdivisión de personal
	3.	Mensaje: reacción si no existe ninguna entrada:

## Programación en HR

		X el report finaliza con el mensaje D001: <i>Ninguna entrada en la clave de tabla T5DOP...</i>
		SPACE SY-SUBRC se fija en 4.
Salida	1.	Título en la tabla (interna) T5DOP
	2.	SY-SUBRC 0 se ha leído con éxito 4 el registro no existe

### *0.0.0.0 Utilización*

En cualquier punto de cualquier programa (inclusive la base de datos PNP).

```

PROVIDE * FROM P0001...
RP-READ-T5DOP P0001-WERKS P0001-BTRTL SPACE.
IF SY-SUBRC NE 0.
own error handling
ENDIF.
ENDPROVIDE.
    
```

### **0.0.0.0 RP-READ-PAYROLL-DIR**

Rutinas CD para leer cluster lectura y verificación nómina de empleado.

### *0.0.0.0 Función*

Este módulo contiene las siguientes rutinas:

- READ-PAYROLL-DIRECTORY
- IN\_PERIOD-IN-RGDIR

IN\_PERIOD-IN-RGDIR

Entrada	1. PERNR-PERNR	Número de personal
	2. PN/PERMO	Modificador de período
	3. PN/PAPER	Período utilizando YYYYPP
Salida	1. RP-IMP-CD-SUBRC	0 registros encontrados 4 registros no encontrados
	2. CD de clusters	Estructuras y tablas del CD de cluster
	3. PNP-SW-FOUND	0 Nóminas de empleado no realizadas dentro del período 1 Nómina de empleado realizada dentro del período
	4. RP-SEL-CALC-RRDAT	Cálculos retroactivos para empleado

Para ello, se llama la rutina READ-PAYROLL-DIR.

READ-PAYROLL-DIR

Entrada	1. PERNR-PERNR	Número de personal
---------	----------------	--------------------

## Programación en HR

Salida	1. RP-IMP-CD-SUBRC	0 registros encontrados 4 registros no encontrados
	2. CD de cluster	Estructuras y tablas del CD de cluster

*0.0.0.0.0 Parámetros*

Ninguno.

*0.0.0.0.0 Verificación*

Ninguna.

*0.0.0.0.0 Condiciones previas*

El miembro *INCLUDE RPC2CDOO* tiene que estar presente.

*0.0.0.0.0 Utilización*

La rutina *IN\_PERIOD-IN-RGDIR* puede llamarse mediante *GET PERNR*. Esto se utiliza cuando usted no desea omitir ningún número de personal para el que no se ha realizado la nómina. Véase: Módulo *RMAC RP-SEL-CALC*.

*GET PERNR.*

*PERFORM IN\_PERIOD-IN-RGDIR.*

*IF PNP-SW-FOUND EQ '1'. "Encontrado*

*... "Si*

*ELSE.*

*... "No*

*ENDIF.*

*...*

*(Rutinas)*

*RP-READ-PAYROLL-DIR.*

***0.0.0 Procesamiento de datos***

***0.0.0.0 RP-PROVIDE-FROM-FRST***

Suministro 1ª entrada en período tiempo

*0.0.0.0.0 Función*

La primera entrada del período de tiempo (o subtipo) que desee realizar se toma de una tabla de infotipos interna (definida mediante la ayuda de la declaración del lenguaje de programación INFOTYPES) y se sitúa en la cabecera de la tabla.

*0.0.0.0.0 Parámetros*

RP-PROVIDE-FROM-FRST infytab subty beg end

Entrada	1.	Nombre de la tabla interna
	2.	Subtipo deseado o SPACE si no se ha especificado ningún subtipo
	3.	Fecha de inicio de validez del intervalo de tiempo

## Programación en HR

	4.	Fecha de fin de validez del intervalo de tiempo
Salida	1.	PNP-SW-FOUND: si no existe ninguna entrada con la que coincida en la tabla de infotipos de un período de tiempo determinado, posee el valor 0. De lo contrario, adopta el valor 1.
	2.	La entrada de la cabecera de la tabla que coincide si PNP-SW-OUND = 1 o la entrada de la cabecera de la tabla borrada, si PNP-SW-FOUND = 0

### *0.0.0.0.0 Verificación*

Ninguno.

### *0.0.0.0.0 Condiciones previas*

- Es preciso que la fecha de inicio de validez del período de tiempo sea anterior o coincida con la fecha de fin de validez.
- El inicio y el fin de la validez constituyen fechas correctas (preferiblemente del tipo DATE).
- La tabla de infotipos se clasifica en orden ascendente (si no se hace así, la primera entrada que coincida de entre las que se encontraron en la tabla no tiene por qué ser necesariamente la primera en orden cronológico).

### *0.0.0.0.0 Utilización*

Se utiliza en aquellos programas que utilizan la base de datos PNP y en los que desea leer el primer registro de tiempos de un período (o de un subtipo) de una tabla de infotipos (que se ha completado anteriormente utilizando, por ejemplo, GET PERNR o RP-READ-INFOTYPE). Sólo merece la pena utilizarlo si la tabla de infotipos (o subtipos, en el caso de que se hayan especificado) contiene el enlace temporal 1 ó 2.

```
RP-PROVIDE-FROM-FRST P0021 '1' PN/BEGDA PN/ENDDA.
IF PNP-SW-FOUND EQ '1'. " Se ha encontrado
```

```
...
ENDIF
```

*o*

```
RP-PROVIDE-FROM-FRST P0001 SPACE PN-BEGDA PN-ENDDA.
IF PNP-SW-FOUND EQ '0'.
```

```
WRITE: / 'Error '.
```

```
REJECT.
```

```
ENDIF.
```

### *0.0.0.0.0 Observaciones*

El módulo (no implementado) *PROVIDE-FROM-INITIAL* es un caso especial de *PROVIDE-FROM-FIRST*.

```
PROVIDE-FROM-INITIAL infytab subty beg end =
RP-PROVIDE-FROM-FRST infytab subty beg beg
```

### ***0.0.0.0 RP-PROVIDE-FROM-LAST***

Suministra última entrada de período de tiempo.

## Programación en HR

### 0.0.0.0.0 *Función*

La última entrada del período de tiempo (o subtipo) que usted desee se toma de una tabla de infotipo interna (definida mediante la declaración del lenguaje de programación INFOTYPES) y se sitúa en la cabecera de la tabla.

### 0.0.0.0.0 *Parámetros*

*RP-PROVIDE-FROM-LAST infytab subty beg end*

Entrada	1.	Nombre de la tabla interna
	2.	Subtipo deseado o SPACE, si no se ha especificado ningún subtipo
	3.	Fecha de inicio de validez del intervalo de tiempo
	4.	Fecha de fin de validez del intervalo de tiempo
Salida	1.	PNP-SW-FOUND: si en la tabla de infotipos de un período de tiempo determinado no existe ninguna entrada con la que coincida, posee el valor 0. De lo contrario, adopta el valor 1.
	2.	La entrada de la cabecera de la tabla que coincide si PNP-SW-FOUND = 1 o la entrada de la cabecera de la tabla borrada, si PNP-SW-FOUND = 0

### 0.0.0.0.0 *Verificación*

Ninguna.

### 0.0.0.0.0 *Condiciones previas*

- Es preciso que la fecha de inicio de validez del período de tiempo sea anterior o coincida con la fecha de fin de validez.
- El inicio y el fin de la validez constituyen fechas correctas (preferiblemente del tipo DATE).
- La tabla de infotipos se clasifica en orden ascendente (de lo contrario, la última entrada que coincida con las que existen en la tabla no tiene por qué ser la última en orden cronológico).

### 0.0.0.0.0 *Utilización*

Se utiliza en aquellos programas que utilizan la base de datos PNP y en los que desea leer el último registro de tiempos de un período (o de un subtipo) de una tabla de infotipos (que se ha completado anteriormente utilizando, por ejemplo, GET PERNR o RP-READ-INFOTYPE). Sólo merece la pena utilizarlo si la tabla de infotipos (o subtipo, en caso de que se hayan especificado) contiene el enlace temporal 1 ó 2.

```
RP-PROVIDE-FROM-LAST P0021 '1' PN/BEGDA PN/ENDDA.
IF PNP-SW-FOUND EQ '1'. "spouse data?"
```

```
...
ENDIF.
```

o

```
RP-PROVIDE-FROM-LAST P0001 SPACE PN/BEGDA PN/ENDDA.
IF PNP-SW-FOUND EQ '0'.
```

```
WRITE: / 'Error: Organizational assignment lacking'.
REJECT.
ENDIF.
```

## Programación en HR

### 0.0.0.0.0 Observaciones

El módulo (no implementado) PROVIDE-FROM-FINAL es un caso especial de PROVIDE-FROM-LAST:

```
PROVIDE-FROM-FINAL infytyab subty beg end =  
RP-PROVIDE-FROM-LAST infytyab subty beg beg
```

### 0.0.0 Modificación de datos

#### 0.0.0.0 RP-UPDATE

##### 0.0.0.0.0 Función

Interfases entre los reports que actualizan datos del fichero PREL y tareas de actualización. Los registros sólo pueden actualizarse, pero no añadirse o borrarse.

##### 0.0.0.0.0 Parámetros

*RP-UPDATE old-table new-table*

Entrada	1.	Nombre de la tabla interna con registros de infotipo antes de la actualización.
	2.	Nombre de la tabla interna con registros de infotipo después de la actualización.

##### 0.0.0.0.0 Verificación

Ninguna.

##### 0.0.0.0.0 Condiciones previas

- Todos los registros que se actualicen también se incluyen en una tabla con su forma original.
- Las claves no se modifican, es decir, ni se añaden ni se borran datos.

##### 0.0.0.0.0 Utilización

En programas de actualización de datos.

*RP-UPDATE OLD-P0003 P0003.*

## 0 Utilidades en HR

Utilidades que tenemos en HR

### 0.0 Utilidades generales

Report	Significado
--------	-------------

## Programación en HR

RPUACG00	Generación de código para autorización de datos maestros de HR
RPUAUD00	Auditoría de infotipo

### 0.0 Utilidades de programación

Report	Significado
RPINCL10	Búsqueda de string en reports

### 0.0 Utilidades de cluster

Report	Meaning
RPCLSTyy	Visualizar clusters en PCLx (yy = Relid)
RPUPxD00	Borrar cluster de PCLx (registros individuales)
RPUPxD10	Borrar cluster de PCLx (varios registros)