

Un paquete consta de los siguientes elementos:

- Especificación o cabecera: contiene las declaraciones públicas (es decir, accesibles desde cualquier parte de la aplicación) de sus programas, tipos, constantes, variables, cursores, excepciones, etc.
- Cuerpo: contiene los detalles de implementación y declaraciones privadas, es decir, accesibles solamente desde los objetos del paquete.

La sintaxis de la cabecera es la siguiente:

```
create [or replace] package nombre_paquete as
  <declaraciones públicas>
  <especificaciones de subprogramas>
end nombre_paquete;
```

La sintaxis del cuerpo sería la siguiente:

```
create [or replace] package body nombre_paquete as
  <declaraciones privadas>
  <código de subprogramas>
  [begin
    <instrucciones iniciales>]
end nombre_paquete;
```

Como podéis observar la cabecera se compila independientemente del cuerpo. Os dejo un ejemplo de paquete para que lo veáis más claro.

```
/* Cabecera */
create or replace package busar_emple as
  TYPE t_reg_emple is RECORD
    (num_empleado emple.emp_no%TYPE,
     apellido emple.apellido%TYPE,
```

```

        salario emple.salario%TYPE,
        departamento emple.dept_no%TYPE);
    procedure ver_por_numero(v_cod
emple.emp_no%TYPE);
    procedure ver_por_apellido(v_ape
emple.apellido%TYPE);
    function datos (v_cod emple.emp_no%TYPE)
        return t_reg_emple;
end buscar_emple;

```

/* Cuerpo */

```

create or replace package body buscar_emple as
    vg_emple t_reg_emple;
    procedure ver_emple; /* procedimiento privado*/
    procedure ver_por_numero (v_cod
emple.emp_no%TYPE)
    is
    begin
        select emp_no, apellido, salario, dept_no into
vg_emple from emple where emp_no=v_cod;
        ver_emple;
    end ver_por_numero;
    procedure ver_por_apellido (v_ape
emple.apellido%TYPE)
    is
    begin
        select emp_no,apellido,salario,dept_no into
vg_emple from emple where apellido=v_apellido;
        ver_emple;
    end ver_por_apellido;
    function datos (v_cod emple.emp_no%TYPE)
        return t_reg_emple

```

```

is
begin
    select emp_no,apellido,salario,dept_no into
vg_emple from emple where emp_no=v_cod;
    procedure ver_emple
    is
    begin
        DBMS_OUTPUT.PUT_LINE(vg_emple.num_emple
ado || '*' || vg_emple.apellido || '*' || vg_emple.salario ||
'*'|| vg_emple.departamento);
    end ver_emple;
end buscar_emple;

```

Como podéis ver este paquete nos permite buscar un empleado de tres formas distintas y visualizar sus datos.

Utilización de los objetos definidos en los paquetes

Podemos utilizar los objetos definidos en los paquetes básicamente de dos maneras distintas:

- Desde el mismo paquete: esto quiere decir que cualquier objeto puede ser utilizado dentro del paquete por otro objeto declarado en el mismo. Para utilizar un objeto dentro del paquete tan sólo tendríamos que llamarlo. La llamada sería algo así: `v_emple := buscar_emple.datos(v_n_ape);` (como veis no utilizamos el `execute` ya que nos encontramos dentro del paquete).
- Desde fuera del paquete: Podemos utilizar los objetos de un paquete siempre y cuando haya sido declarado en la especificación del mismo.

Para llamar a un objeto o procedimiento desde fuera del paquete utilizaríamos la siguiente notación: execute
nombre_paquete.nombre_procedimiento(lista de parametros);

Declaración de cursores en paquetes

En los paquetes también podemos [introducir cursores](#), para ello debemos declararlo en la cabecera del paquete indicando su nombre, los parámetros y tipo devuelto. Para que lo veáis más claro os dejo un ejemplo a continuación:

```
CREATE or REPLACE PACKAGE empleados AS
```

```
.....
```

```
CURSOR a RETURN empleado%ROWTYPE;
```

```
...
```

```
END empleados;
```

```
CREATE or REPLACE PACKAGE BODY empleados AS
```

```
....
```

```
CURSOR a RETURN empleado%ROWTYPE
```

```
SELECT * FROM empleado WHERE salario < 10000;
```

```
....
```

```
END empleados;
```