

Un paquete se compone de dos partes obligatorias: la **Especificación** y el **Cuerpo**.

1. Especificación (SPEC)

La especificación actúa como una **interfaz pública** o una "tabla de contenidos". Define qué procedimientos, funciones o variables son accesibles desde fuera del paquete (es decir, desde SQL o desde otros bloques PL/SQL). Solo contiene las *declaraciones* de los componentes.

2. Cuerpo (BODY)

El cuerpo contiene la **implementación** (el código real) de todos los procedimientos y funciones declarados en la especificación. También puede contener elementos privados (variables, funciones auxiliares) que solo son visibles dentro del propio cuerpo.

Guía Paso a Paso: Creación de un Package

Vamos a crear un paquete simple llamado `gestion_empleados` que agrupará la lógica de bonos y creación de personal.

Paso 1: Crear la Especificación (SPEC)

Definimos los componentes públicos que otros usuarios pueden ver y usar.

```
CREATE OR REPLACE PACKAGE gestion_empleados
```

```
AS
```

```
-- 1. Constante pública
```

```
TASA_MAX_BONO CONSTANT NUMBER := 0.20;
```

```
-- 2. Declaración de Función Pública
```

```
-- Calcula el bono, requiere el salario y devuelve un NUMBER.
```

```
FUNCTION calcular_bono (
```

```
    p_salario IN empleados.salario%TYPE
```

```
)
```

```
    RETURN NUMBER;
```

```
-- 3. Declaración de Procedimiento Público
```

```
-- Crea un empleado y devuelve el nuevo ID.
```

```
PROCEDURE crear_empleado (
```

```
    p_nombre IN empleados.nombre%TYPE,
```

```
    p_salario IN empleados.salario%TYPE,
```

```
p_jefe_id IN empleados.jefe_id%TYPE,  
p_nuevo_id OUT empleados.emp_id%TYPE  
);
```

```
END gestion_empleados;
```

```
/
```

Paso 2: Crear el Cuerpo (BODY)

Implementamos la lógica para la función y el procedimiento declarados.

```
CREATE OR REPLACE PACKAGE BODY gestion_empleados
```

```
AS
```

```
-- Función Privada (No declarada en el SPEC, solo visible aquí)
```

```
FUNCTION validar_salario_minimo (
```

```
    p_salario IN empleados.salario%TYPE
```

```
)
```

```
RETURN BOOLEAN
```

```
IS
```

```
BEGIN
```

```
-- Ejemplo: El salario debe ser mayor a 1000
```

```
    RETURN p_salario > 1000;
```

```
END validar_salario_minimo;
```

```
-- 1. Implementación de la Función Pública (calcular_bono)
```

```
FUNCTION calcular_bono (
```

```
    p_salario IN empleados.salario%TYPE
```

```
)
```

```
RETURN NUMBER
```

```
IS
```

```
BEGIN
```

```
    IF p_salario > 8000 THEN
```

```

    RETURN p_salario * TASA_MAX_BONO; -- Usando la constante pública
ELSE
    RETURN p_salario * 0.10;
END IF;
END calcular_bono;

```

-- 2. Implementación del Procedimiento Público (crear_empleado)

```

PROCEDURE crear_empleado (
    p_nombre IN empleados.nombre%TYPE,
    p_salario IN empleados.salario%TYPE,
    p_jefe_id IN empleados.jefe_id%TYPE,
    p_nuevo_id OUT empleados.emp_id%TYPE
)
AS
    v_next_id NUMBER;
BEGIN
    -- Primero, validamos usando la función PRIVADA
    IF NOT validar_salario_minimo(p_salario) THEN
        RAISE_APPLICATION_ERROR(-20002, 'Salario propuesto es muy bajo.');
    END IF;

    SELECT NVL(MAX(emp_id), 0) + 1 INTO v_next_id FROM empleados;

    INSERT INTO empleados (emp_id, nombre, salario, jefe_id)
    VALUES (v_next_id, p_nombre, p_salario, p_jefe_id);

    p_nuevo_id := v_next_id;
    COMMIT;
END crear_empleado;

```

```
END gestion_empleados;
```

```
/
```

3. Uso del Package

Todos los componentes públicos se invocan usando la sintaxis
NOMBRE_PAQUETE.NOMBRE_COMPONENTE.

Uso de la Función y la Constante

-- 1. Usar la función en un SELECT

```
SELECT nombre, salario,  
       gestion_empleados.calcular_bono(salario) AS bono_calculado  
  FROM empleados  
 WHERE salario > 5000;
```

-- 2. Usar la constante pública (dentro de un bloque PL/SQL)

```
DECLARE  
    v_max_tasa NUMBER := gestion_empleados.TASA_MAX_BONO;  
BEGIN  
    DBMS_OUTPUT.PUT_LINE('La tasa máxima de bono es: ' || v_max_tasa);  
END;  
/
```

Uso del Procedimiento

```
DECLARE
```

```
    v_nuevo_id NUMBER;  
BEGIN  
    gestion_empleados.crear_empleado(  
        p_nombre => 'Maria Nueva',  
        p_salario => 9000,  
        p_jefe_id => 1,  
        p_nuevo_id => v_nuevo_id  
    );
```

```
DBMS_OUTPUT.PUT_LINE('Nuevo empleado ID: ' || v_nuevo_id);  
END;
```

```
/
```

Beneficios de los Packages

Los paquetes no son solo para organizar; ofrecen ventajas de rendimiento y diseño:

Beneficio	Descripción	Impacto
Modularidad	Agrupa lógica relacionada (ej. todo lo de RH va en un paquete).	Mantenimiento y legibilidad más sencillos.
Variables de Sesión	Las variables declaradas en la especificación mantienen su valor durante toda la sesión (Globales privadas).	Útil para contadores o variables de estado que persisten entre llamadas.
Privacidad	Los elementos en el cuerpo pero no en la especificación son invisibles.	Encapsulación de lógica interna y seguridad.
Compilación Parcial	Si solo cambias el cuerpo , Oracle no obliga a recompilar todos los objetos que dependen de la especificación .	Menos interrupciones en entornos de producción.
Carga de Memoria	Cuando llamas a una función de un paquete, Oracle carga el paquete completo en memoria una sola vez.	Mejora el rendimiento porque las llamadas posteriores son más rápidas.