

ciclo: [DAM]

MÓDULO DE [BASES DE DATOS]

[Tarea Nº 06]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc71_871220036)

[2. (RA05\_d) Se han definido y utilizado guiones para automatizar tareas y (RA05\_g) Se han utilizado estructuras de control de flujo. 3](#__RefHeading___Toc85_319373017)

[Bloque anónimo PL/SQL para calcular el 'salario anual' de un empleado: 3](#__RefHeading___Toc87_319373017)

[Bloque anónimo PL/SQL [completo] 4](#__RefHeading___Toc89_319373017)

[Resultado por pantalla 5](#__RefHeading___Toc91_319373017)

[3. (RA05\_f) Se han definido procedimientos y funciones de usuario y (RA05\_j) Se han utilizado excepciones 6](#__RefHeading___Toc85_319373017_Copia_1)

[Función PL/SQL 6](#__RefHeading___Toc103_319373017)

[Función PL/SQL [completo] 8](#__RefHeading___Toc89_319373017_Copia_1)

[Crear Bloque PL/SQL para ejecutar la función getNombreApellidos 8](#__RefHeading___Toc89_319373017_Copia_1_C)

[Ejecución de la función 9](#__RefHeading___Toc91_319373017_Copia_1)

# Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Archivo JuanCarlosFilterMartinTarea6.SQL

# **(RA05\_d)** Se han definido y utilizado guiones para automatizar tareas y **(RA05\_g)** Se han utilizado estructuras de control de flujo.

## Bloque anónimo PL/SQL para calcular el 'salario anual' de un empleado:

* Tiene que pedir nombre y apellido por una variable de sustitución

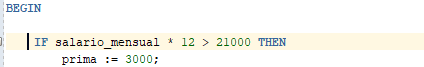


* Tiene que pedir el salario mensual en otra variable de sustitución



Este bloque PL/SQL debe:

* Calcular el salario anual : ('salario mensual' \* 12) + 'prima' y en función del salario mensual le indicamos que prima le corresponde mediante diferentes IF
  + - Si tiene más de 21000 = prima +3000



* + - Si tiene entre 12000 y 21000 (incluidos) = prima +1800



* + - Si tiene menos de 12000 = prima+1000

*(finalmente con un ELSE ya que solo quedaría esta opción)*

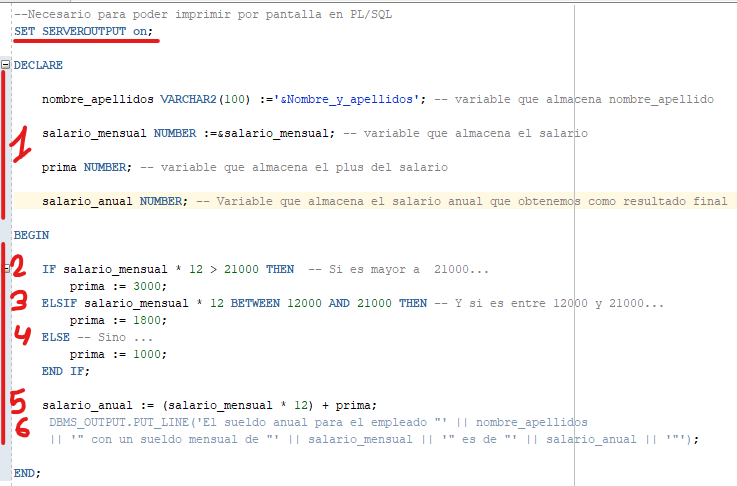


* Añadimos a la variable salario\_anual el salario mensual\*12 + la prima correspondiente y mostramos el resultado por pantalla.



### Bloque anónimo PL/SQL [completo]

Se va a mostrar todo el bloque PL/SQL indicando que hace cada línea:



* En primer lugar tenemos que activar el servicio para poder mostrar por pantalla

**SET SERVEROUTPUT on**

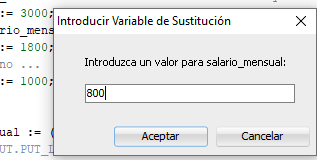
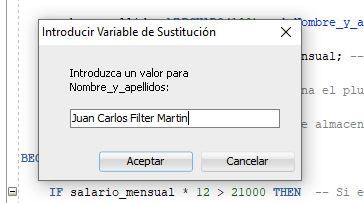
1. Declaramos las variables de sustitución y las variables necesarias ‘prima’ y ‘salario\_anual’
2. Condición IF que entra si el salario mensual \* 12 es mayor a 21000
3. Entra si el salario mensual \* 12 está entre 12000 y 21000 (incluidos)
4. Por ultimo si nada de lo anterior se cumple entra en el ELSE que sería si es menor a 12000
5. Se almacena en salario anual el resultado de salario mensual\*12 + prima correspondiente.
6. Mostrar por pantalla con la concatenación indicada.

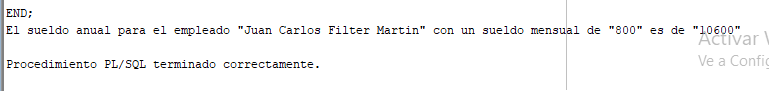
### Resultado por pantalla

**Prueba 1:**

nombre\_apellidos: Juan Carlos Filter Martin

salario\_mensual: 800

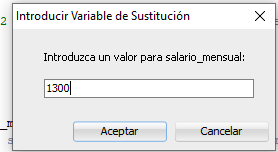
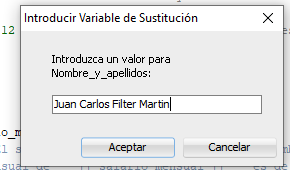


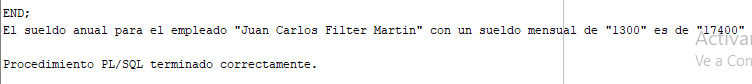


**Prueba 2:**

nombre\_apellidos: Juan Carlos Filter Martin

salario\_mensual: 1300

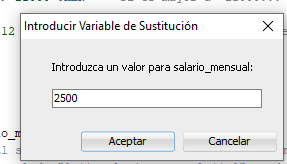
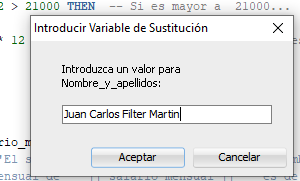


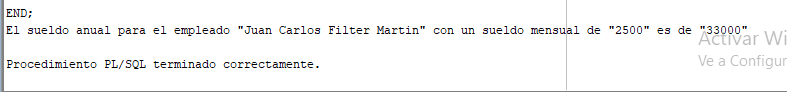


**Prueba 3:**

nombre\_apellidos: Juan Carlos Filter Martin

salario\_mensual: 2500





# **(RA05\_f)** Se han definido procedimientos y funciones de usuario y **(RA05\_j)** Se han utilizado excepciones

## Función PL/SQL

Se va a crear una función llamada 'getNombreApellidos' que va a consultar en la tabla ‘Employees’ del esquema HR para devolver nombre y apellido de un empleado pasándole el id de empleado.

* Función llamada getNombreApellidos

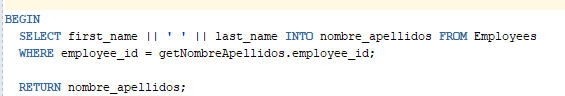


* Parámetro de entrada el id de empleado y devuelve como salida "Nombre Apellidos" es la concatenación de los campos 'first\_name' y 'last\_name'.



Esta función PL/SQL hace lo siguiente:

* Realizamos una consulta a la tabla ‘Employees’ con el id del empleado pasado por parámetro y recogemos el nombre y apellido que se va a concatenar y guardar en la variable nombre\_apellido
* Y va a ser devuelto mediante un return a nombre y apellidos que es la salida de la función

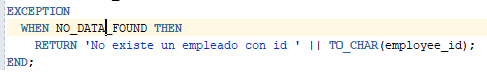


* Si NO existe un empleado con dicho id *(WHEN NO\_DATA\_FOUND THEN)* ...

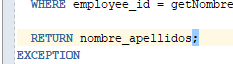
va a devolver una excepción como cadena de texto

**“No existe un empleado con id “ID”**

**CONVERSION EXPLÍCITA A VARCHAR2**

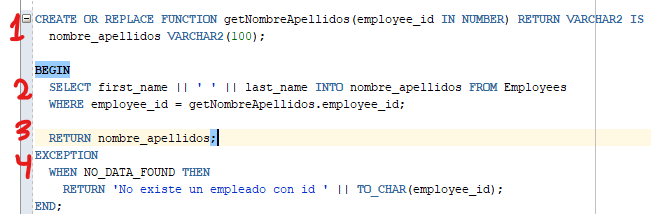
****

* Si existe un empleado con el id entonces devolvemos la cadena nombre\_apellido mediante el return y este va a ser enviado cuando se llame a la función





### Función PL/SQL [completo]

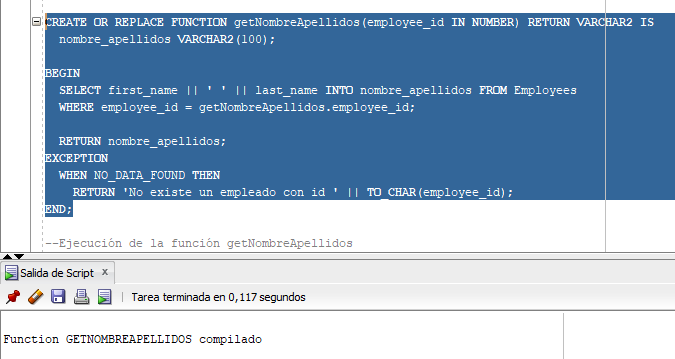
1. Función getNombreApellidos que recoge por parámetros el id y retorna la concatenación de nombre\_apellido.
2. Consulta a la tabla Employees con el id del empleado pasado por parámetro (mediante el where se compara ambos id) y recogemos el nombre y apellido que se va a concatenar y guardar en la variable nombre\_apellido.
3. Devolvemos la concatenación del nombre y apellido.
4. Si no existe el id mandará un mensaje de error con el texto indicado.

### Crear Bloque PL/SQL para ejecutar la función getNombreApellidos

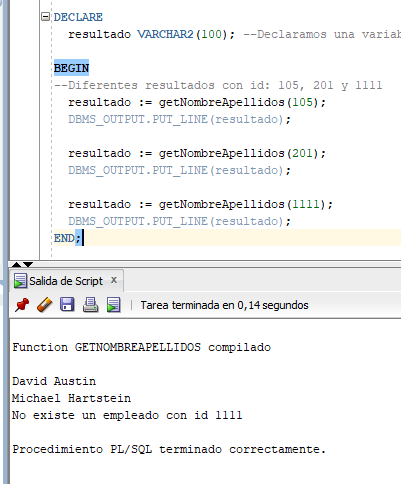
1. Se declara una variable que va a obtener el resultado de la función
2. Se va a llamar a la función, indicándole por parámetros diferentes id y almacenándolo en resultado.

### Ejecución de la función

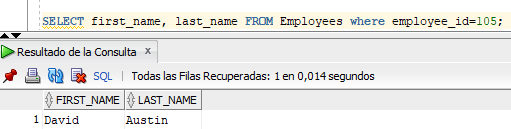
1. Compilamos la función



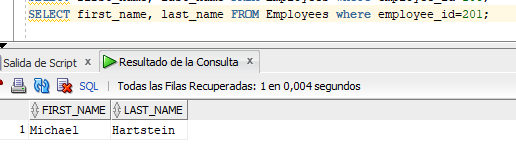
1. Ejecutamos el bloque PL/SQL para llamar a la función con los id: 105, 201 y 1111



El ID 105 : corresponde a David Austin



El ID 201 : corresponde a Michael Hartstien



El ID 1111 : No corresponde a ningún empleado

