

**PFC – BASES DE DATOS**  
**Anexo 1 - Test del Sistema**



**GESTIÓN POSTVENTA**

## Índice

1	TEST01 – TEST_SERVICIOS.....	3
2	TEST02 – TEST_ELECTRODOMESTICOS.....	4
3	TEST03 – TEST_PROVINCIAS.....	5
4	TEST04 – TEST_POBLACIONES.....	6
5	TEST05 – TEST_ZONAS.....	7
6	TEST06 – TEST_TELEOPERADORES.....	9
7	TEST07 – TEST_TECNICOS.....	10
8	TEST08 – TEST_CLIENTES.....	13
9	TEST09 – TEST_PRODUCTOS.....	14
10	TEST10 – TEST_PETICIONES.....	15
11	TEST11 – TEST_ASIGNACION.....	16
12	TEST12 – TEST_TRABJOS.....	17
13	TEST13 – TEST_ENCUESTAS.....	18
14	TEST14 - CONSULTAS SUBSISTEMA CONFIGURACION.....	19
15	TEST15 - CONSULTAS SUBSISTEMA PERSONAL.....	25
16	TEST16 - CONSULTAS SUBSISTEMA PETICIÓN.....	33
17	TEST17 - CONSULTAS SUBSISTEMA ATENCIÓN.....	37
18	TEST18 - CONSULTAS SUBSISTEMA HISTORIAL.....	43

## 1 TEST01 – TEST\_SERVICIOS

EXEC TEST\_CONFIGURACION.TEST\_SERVICIOS;

### PRUEBAS:

#### INS\_SERVICIO

Se insertan tres servicios. (1, 2, 3)

Se generan los siguientes errores:

- Insertar servicio con una descripción existente. (4)
- Insertar servicio con descripción nula. (5)

#### DEL\_SERVICIO

Se dan de baja dos servicios. (6, 7)

Se generan los siguientes errores:

- Baja de servicio no existente. (8)
- Baja de servicio dado de baja. (9)

#### RES\_SERVICIO

Se recupera un servicio. (10)

Se generan los siguientes errores:

- Recuperar servicio no existente. (11)
- Recuperar servicio no dado de baja. (12)

#### UPD\_SERVICIO

Se modifica un servicio. (13)

Se generan los siguientes errores:

- Modificar servicio no existente. (14)
- Modificar servicio con una descripción que ya existe. (15)
- Modificar servicio con valor nulo. (16)
- Modificar servicio dado de baja. (17)

### RESULTADO

1	OK. El servicio se ha creado con éxito
2	OK. El servicio se ha creado con éxito
3	OK. El servicio se ha creado con éxito
4	ERROR: La descripción de servicio ya existe.
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "SERVICIOS", "DSSERVICIO")
6	OK. El servicio se ha dado de baja con éxito
7	OK. El servicio se ha dado de baja con éxito
8	ERROR: El identificador de servicio no existe.
9	ERROR: El servicio ya está de baja.
10	OK. El servicio se ha recuperado con éxito
11	ERROR: El identificador de servicio no existe.
12	ERROR: El servicio no está de baja.
13	OK. El servicio se ha modificado con éxito
14	ERROR: El identificador de servicio no existe.
15	ERROR: La descripción de servicio ya existe.
16	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "SERVICIOS", "DSSERVICIO") a un valor NULL
17	ERROR: El servicio está de baja.
18	

## 2 TEST02 – TEST\_ELECTRODOMESTICOS

EXEC TEST\_CONFIGURACION.TEST\_ELECTRODOMESTICOS;

### PRUEBAS:

#### INS\_ELECTRODOMESTICO

Se insertan tres electrodomésticos. (1, 2, 3)

Se generan los siguientes errores:

- Insertar electrodoméstico con una descripción existente. (4)
- Insertar electrodoméstico con descripción nula. (5)

#### DEL\_ELECTRODOMESTICO

Se dan de baja dos electrodomésticos. (6, 7)

Se generan los siguientes errores:

- Baja de electrodoméstico no existente. (8)
- Baja de electrodoméstico dado de baja. (9)

#### RES\_ELECTRODOMESTICO

Se recupera un electrodoméstico. (10)

Se generan los siguientes errores:

- Recuperar electrodoméstico no existente. (11)
- Recuperar electrodoméstico no dado de baja. (12)

#### UPD\_ELECTRODOMESTICO

Se modifica un electrodoméstico. (13)

Se generan los siguientes errores:

- Modificar electrodoméstico no existente. (14)
- Modificar electrodoméstico con una descripción que ya existe. (15)
- Modificar electrodoméstico con valor nulo. (16)
- Modificar electrodoméstico dado de baja. (17)

### RESULTADO

1	OK. El electrodoméstico se ha creado con éxito	
2	OK. El electrodoméstico se ha creado con éxito	
3	OK. El electrodoméstico se ha creado con éxito	
4	ERROR: La descripción de electrodoméstico ya existe.	
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "ELECTRODOMESTICOS", "DSELECTRODOMESTICO")	
6	OK. El electrodoméstico se ha dado de baja con éxito	
7	OK. El electrodoméstico se ha dado de baja con éxito	
8	ERROR: El identificador de electrodoméstico no existe.	
9	ERROR: El electrodoméstico ya está de baja.	
10	OK. El electrodoméstico se ha recuperado con éxito	
11	ERROR: El identificador de electrodoméstico no existe.	
12	ERROR: El electrodoméstico no está de baja.	
13	OK. El servicio se ha modificado con éxito	
14	ERROR: El identificador de electrodoméstico no existe.	
15	ERROR: La descripción de electrodoméstico ya existe.	
16	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "ELECTRODOMESTICOS", "DSELECTRODOMESTICO") a un valor NULL	
17	ERROR: El electrodoméstico está de baja.	
18		

### 3 TEST03 – TEST\_PROVINCIAS

EXEC TEST\_CONFIGURACION.TEST\_PROVINCIAS;

#### PRUEBAS:

##### INS\_PROVINCIA

Se insertan tres provincias. (1, 2, 3)

Se generan los siguientes errores:

- Insertar provincia con código existente. (4)
- Insertar provincia con nombre existente. (5)
- Insertar provincia con valor nulo. (6, 7)

##### DEL\_PROVINCIA

Se da de baja una provincia. (8)

Se generan los siguientes errores:

- Baja de provincia no existente. (9)
- Baja de provincia con poblaciones asignadas. (10)

##### UPD\_PROVINCIA

Se modifica una provincia. (11)

Se generan los siguientes errores:

- Modificar provincia no existente. (12)
- Modificar provincia con un nombre que ya existe. (13)
- Modificar provincia con valor nulo. (14)

#### RESULTADO

1	OK. La provincia se ha creado con éxito
2	OK. La provincia se ha creado con éxito
3	OK. La provincia se ha creado con éxito
4	ERROR: El código de provincia ya existe.
5	ERROR: El nombre de provincia ya existe.
6	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PROVINCIAS", "COPROVINCIA")
7	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PROVINCIAS", "DSPROVINCIA")
8	OK. La provincia se ha dado de baja con éxito
9	ERROR: El código de provincia no existe.
10	ERROR: La provincia tiene poblaciones asignadas.
11	OK. La provincia se ha modificado con éxito
12	ERROR: El código de provincia no existe.
13	ERROR: El nombre de la provincia ya existe.
14	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PROVINCIAS", "DSPROVINCIA") a un valor NULL
15	

#### 4 TEST04 – TEST\_POBLACIONES

EXEC TEST\_CONFIGURACION.TEST\_POBLACIONES;

##### PRUEBAS:

###### INS\_POBLACION

Se insertan cuatro poblaciones. (1, 2, 3, 4)

Se generan los siguientes errores:

- Insertar población con código existente. (5)
- Insertar población con nombre existente. (6)
- Insertar población con provincia no existente. (7)
- Insertar población con valor nulo. (8, 9, 10)

###### DEL\_POBLACION

Se da de baja una población. (11)

Se generan los siguientes errores:

- Baja de población no existente. (12)
- Baja de población asignada a alguna zona. (13)

###### UPD\_POBLACION

Se modifica una población. (14)

Se generan los siguientes errores:

- Modificar población no existente. (15)
- Modificar población con un nombre que ya existe. (16)
- Modificar población con una provincia no existente. (17)
- Modificar provincia con valor nulo. (18, 19)

##### RESULTADO

1	OK. La población se ha creado con éxito
2	OK. La población se ha creado con éxito
3	OK. La población se ha creado con éxito
4	OK. La población se ha creado con éxito
5	ERROR: El código postal ya existe.
6	ERROR: El nombre de la población ya existe.
7	ERROR: El código de provincia no existe.
8	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "POBLACIONES", "CODPOSTAL")
9	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "POBLACIONES", "DSPOBLACION")
10	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "POBLACIONES", "COPROVINCIA")
11	OK. La población se ha dado de baja con éxito
12	ERROR: El código postal no existe.
13	ERROR: La población está asignada a alguna zona.
14	OK. La población se ha modificado con éxito
15	ERROR: El código postal no existe.
16	ERROR: El nombre de la población ya existe.
17	ERROR: El código de provincia no existe.
18	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "POBLACIONES", "DSPOBLACION") a un valor NULL
19	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "POBLACIONES", "COPROVINCIA") a un valor NULL
20	

## 5 TEST05 – TEST\_ZONAS

EXEC TEST\_CONFIGURACION.TEST\_ZONAS;

### PRUEBAS:

#### INS\_ZONAS

Se insertan dos zonas. (1, 2)

Se generan los siguientes errores:

- Insertar zona con nombre existente. (3)
- Insertar población con valor nulo. (4)

#### DEL\_ZONA

Se da de baja una población. (5)

Se generan los siguientes errores:

- Baja de zona no existente. (6)
- Baja de zona con poblaciones asignadas. (7)

#### UPD\_ZONA

Se modifica una zona. (8)

Se generan los siguientes errores:

- Modificar zona no existente. (9)
- Modificar zona con un nombre que ya existe. (10)
- Modificar zona con valor nulo. (11)

#### ASG\_PROVINCIAZONA

Se asigna las poblaciones de una provincia. (12)

Se generan los siguientes errores:

- Asignar una provincia no existente. (13)
- Asignar una provincia a una zona no existente. (14)

#### ASG\_POBLACIONZONA

Se asigna una población. (15)

Se generan los siguientes errores:

- Asignar una población no existente. (16)
- Asignar una población a una zona no existente. (17)
- Asignar una población ya asignada a la zona. (18)

#### DSG\_PROVINCIAZONA

Se desasigna las poblaciones de una provincia. (19)

Se generan los siguientes errores:

- Desasignar una provincia no existente. (20)
- Desasignar una provincia a una zona no existente. (21)

#### DSG\_POBLACIONZONA

Se desasigna una población. (22)

Se generan los siguientes errores:

- Desasignar una población no existente. (23)
- Desasignar una población a una zona no existente. (24)

**RESULTADO**

1	OK. La zona se ha creado con éxito	
2	OK. La zona se ha creado con éxito	
3	ERROR: La descripción de la zona ya existe.	
4	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "ZONAS", "DSZONA")	
5	OK. La zona se ha dado de baja con éxito	
6	ERROR: El identificador de zona no existe.	
7	ERROR: La zona tiene poblaciones asignadas.	
8	OK. La zona se ha modificado con éxito	
9	ERROR: El identificador de zona no existe.	
10	ERROR: La descripción de zona ya existe.	
11	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "ZONAS", "DSZONA") a un valor NULL	
12	OK. La asignación se ha realizado con éxito	
13	ERROR: El código de provincia no existe.	
14	ERROR: El identificador de zona no existe.	
15	OK. La asignación se ha realizado con éxito	
16	ERROR: El código postal no existe.	
17	ERROR: El identificador de zona no existe.	
18	ERROR: La población ya está asociada a la zona.	
19	OK. La desasignación se ha realizado con éxito	
20	ERROR: El código de provincia no existe.	
21	ERROR: El identificador de zona no existe.	
22	OK. La desasignación se ha realizado con éxito	
23	ERROR: El código postal no existe.	
24	ERROR: El identificador de zona no existe.	
25		



## 6 TEST06 – TEST\_TELEOPERADORES

EXEC TEST\_PERSONAL.TEST\_TELEOPERADORES;

### PRUEBAS:

#### INS\_TELEOPERADOR

Se insertan tres teleoperadores. (1, 2, 3)

Se generan los siguientes errores:

- Insertar teleoperador con un NIF existente. (4)
- Insertar teleoperador con valores nulos. (5, 6, 7)

#### DEL\_TELEOPERADOR

Se dan de baja dos teleoperadores. (8, 9)

Se generan los siguientes errores:

- Baja de teleoperador no existente. (10)
- Baja de teleoperador dado de baja. (11)

#### RES\_TELEOPERADOR

Se recupera un teleoperador. (12)

Se generan los siguientes errores:

- Recuperar teleoperador no existente. (13)
- Recuperar teleoperador no dado de baja. (14)

#### UPD\_TELEOPERADOR

Se modifica un teleoperador. (15)

Se generan los siguientes errores:

- Modificar teleoperador no existente. (16)
- Modificar teleoperador con una descripción que ya existe. (17)
- Modificar teleoperador con valor nulo. (18, 19, 20)
- Modificar teleoperador dado de baja. (21)

### RESULTADO

1	OK. El teleoperador se ha creado con éxito
2	OK. El teleoperador se ha creado con éxito
3	OK. El teleoperador se ha creado con éxito
4	ERROR: El nif del teleoperador ya existe.
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TELEOPERADORES", "NOMBRE")
6	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TELEOPERADORES", "APELLIDOS")
7	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TELEOPERADORES", "NIF")
8	OK. El teleoperador se ha dado de baja con éxito
9	OK. El teleoperador se ha dado de baja con éxito
10	ERROR: El identificador del teleoperador no existe.
11	ERROR: El teleoperador ya está de baja.
12	OK. El teleoperador se ha recuperado con éxito
13	ERROR: El identificador del teleoperador no existe.
14	ERROR: El teleoperador no está de baja.
15	OK. El teleoperador se ha modificado con éxito
16	ERROR: El identificador del teleoperador no existe.
17	ERROR: El nif del teleoperador ya existe.
18	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "TELEOPERADORES", "NOMBRE") a un valor NULL
19	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "TELEOPERADORES", "APELLIDOS") a un valor NULL
20	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "TELEOPERADORES", "NIF") a un valor NULL
21	ERROR: El teleoperador está de baja.
22	

## 7 TEST07 – TEST\_TECNICOS

EXEC TEST\_PERSONAL.TEST\_TECNICOS;

### PRUEBAS:

#### INS\_TECNICO

Se insertan tres técnicos. (1, 2, 3)

Se generan los siguientes errores:

- Insertar técnico con un NIF existente. (4)
- Insertar técnico con valores nulos. (5, 6, 7,8)

#### DEL\_TECNICO

Se dan de baja dos técnicos. (9,10)

Se generan los siguientes errores:

- Baja de técnico no existente. (11)
- Baja de técnico dado de baja. (12)

#### RES\_TECNICO

Se recupera un técnico. (13)

Se generan los siguientes errores:

- Recuperar técnico no existente. (14)
- Recuperar técnico no dado de baja. (15)

#### UPD\_TECNICO

Se modifica un técnico. (16)

Se generan los siguientes errores:

- Modificar técnico no existente. (17)
- Modificar técnico con una descripción que ya existe. (18)
- Modificar técnico con valor nulo. (19, 20, 21, 22)
- Modificar técnico dado de baja. (23)

#### ASG\_SERVICIOTECNICO

Se asignan tres servicios. (24, 25, 26)

Se generan los siguientes errores:

- Asignar un servicio no existente a un técnico. (27)
- Asignar un servicio a un técnico no existente. (28)
- Asignar un servicio dado de baja a un técnico. (29)
- Asignar un servicio a un técnico dado de baja. (30)
- Asignar un servicio ya asignado al mismo técnico. (31)

#### DSG\_SERVICIOTECNICO

Se desasigna servicio a un técnico. (32)

Se generan los siguientes errores:

- Desasignar un servicio no existente a un técnico. (33)
- Desasignar un servicio dado de baja a un técnico. (34)
- Desasignar un servicio a un técnico no existente. (35)
- Desasignar un servicio a un técnico dado de baja. (36)

#### ASG\_ELECTRODOMESTICOTECNICO

Se asignan tres electrodomésticos. (37, 38, 39)

Se generan los siguientes errores:

- Asignar un electrodoméstico no existente a un técnico. (40)
- Asignar un electrodoméstico a un técnico no existente. (41)
- Asignar un electrodoméstico dado de baja a un técnico. (42)
- Asignar un electrodoméstico un técnico dado de baja. (43)
- Asignar un servicio ya asignado al mismo técnico. (44)

## Gestión PostVenta – Test del Sistema

### DSG\_ELECTRODOMESTICOTECNICO

Se desasigna electrodoméstico a un técnico. (45)

Se generan los siguientes errores:

- Desasignar un electrodoméstico no existente a un técnico. (46)
- Desasignar un electrodoméstico dado de baja a un técnico. (47)
- Desasignar un electrodoméstico a un técnico no existente. (48)
- Desasignar un electrodoméstico a un técnico dado de baja. (49)

### ASG\_ZONATECNICO

Se asignan tres zonas. (50, 51, 52)

Se generan los siguientes errores:

- Asignar una zona no existente a un técnico. (53)
- Asignar una zona a un técnico no existente. (54)
- Asignar una zona un técnico dado de baja. (55)
- Asignar una zona ya asignada al mismo técnico. (56)

### DSG\_ZONATECNICO

Se desasigna zona a un técnico. (57)

Se generan los siguientes errores:

- Desasignar una zona no existente a un técnico. (58)
- Desasignar una zona a un técnico no existente. (59)
- Desasignar una zona a un técnico dado de baja. (60)

## RESULTADO

1	OK. El técnico se ha creado con éxito
2	OK. El técnico se ha creado con éxito
3	OK. El técnico se ha creado con éxito
4	ERROR: El nif del técnico ya existe.
5	ERROR: Se ha intentado almacenar un valor nulo ("BD","TECNICOS","NOMBRE")
6	ERROR: Se ha intentado almacenar un valor nulo ("BD","TECNICOS","APELLIDOS")
7	ERROR: Se ha intentado almacenar un valor nulo ("BD","TECNICOS","NIF")
8	ERROR: Se ha intentado almacenar un valor nulo ("BD","TECNICOS","TELMOVIL")
9	OK. El técnico se ha dado de baja con éxito
10	OK. El técnico se ha dado de baja con éxito
11	ERROR: El identificador del técnico no existe.
12	ERROR: El técnico ya está de baja.
13	OK. El técnico se ha recuperado con éxito
14	ERROR: El identificador del técnico no existe.
15	ERROR: El técnico no está de baja.
16	OK. El técnico se ha modificado con éxito
17	ERROR: El identificador del técnico no existe.
18	ERROR: El nif del técnico ya existe.
19	ERROR: Se ha intentado actualizar a un valor nulo ("BD","TECNICOS","NOMBRE") a un valor NULL
20	ERROR: Se ha intentado actualizar a un valor nulo ("BD","TECNICOS","APELLIDOS") a un valor NULL
21	ERROR: Se ha intentado actualizar a un valor nulo ("BD","TECNICOS","NIF") a un valor NULL
22	ERROR: Se ha intentado actualizar a un valor nulo ("BD","TECNICOS","TELMOVIL") a un valor NULL
23	ERROR: El técnico está de baja.
24	OK. La asignación se ha realizado con éxito
25	OK. La asignación se ha realizado con éxito
26	OK. La asignación se ha realizado con éxito
27	ERROR: El identificador de servicio no existe.
28	ERROR: El identificador de técnico no existe.
29	ERROR: El servicio está de baja.
30	ERROR: El técnico está de baja.
31	ERROR: El servicio ya está asociado al técnico.
32	OK. La desasignación se ha realizado con éxito
33	ERROR: El identificador de servicio no existe.
34	ERROR: El servicio está de baja.
35	ERROR: El identificador de técnico no existe.
36	ERROR: El técnico está de baja.
37	OK. La asignación se ha realizado con éxito
38	OK. La asignación se ha realizado con éxito
39	OK. La asignación se ha realizado con éxito
40	ERROR: El identificador de electrodoméstico no existe.
41	ERROR: El identificador de técnico no existe.

## Gestión PostVenta – Test del Sistema

42	ERROR: El electrodoméstico está de baja.
43	ERROR: El técnico está de baja.
44	ERROR: El electrodoméstico ya está asociado al técnico.
45	OK. La desasignación se ha realizado con éxito
46	ERROR: El identificador de electrodoméstico no existe.
47	ERROR: El electrodoméstico está de baja.
48	ERROR: El identificador de técnico no existe.
49	ERROR: El técnico está de baja.
50	OK. La asignación se ha realizado con éxito
51	OK. La asignación se ha realizado con éxito
52	OK. La asignación se ha realizado con éxito
53	ERROR: El identificador de zona no existe.
54	ERROR: El identificador de técnico no existe.
55	ERROR: El técnico está de baja.
56	ERROR: La zona ya está asociada al técnico.
57	OK. La asignación se ha realizado con éxito
58	ERROR: El identificador de zona no existe.
59	ERROR: El identificador de técnico no existe.
60	ERROR: El técnico está de baja.
61	

## 8 TEST08 – TEST\_CLIENTES

EXEC TEST\_PETICION.TEST\_CLIENTES;

### PRUEBAS:

#### INS\_CLIENTE

Se insertan tres clientes. (1, 2, 3)

Se generan los siguientes errores:

- Insertar cliente con un NIF existente. (4)
- Insertar cliente con valores nulos. (5, 6, 7, 8, 9, 10, 11, 12)
- Insertar cliente con un código postal no existente. (13)

#### UPD\_CLIENTE

Se modifica un cliente. (14)

Se generan los siguientes errores:

- Modificar cliente no existente. (15)
- Modificar cliente con un NIF existente. (16)
- Modificar cliente con valor nulo. (17, 18, 19, 20, 21, 22, 23, 24)
- Insertar cliente con un código postal no existente. (25)

### RESULTADO

1	OK. El cliente se ha creado con éxito
2	OK. El cliente se ha creado con éxito
3	OK. El cliente se ha creado con éxito
4	ERROR: El nif del cliente ya existe.
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "NOMBRE")
6	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "APELLIDOS")
7	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "NIF")
8	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "DIRECCION")
9	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "CODPOSTAL")
10	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "TELFIXO")
11	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "TELMOVIL")
12	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "CLIENTES", "EMAIL")
13	ERROR: El código postal no existe
14	OK. El cliente se ha modificado con éxito
15	ERROR: El identificador del cliente no existe.
16	ERROR: El nif del cliente ya existe.
17	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "NOMBRE") a un valor NULL
18	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "APELLIDOS") a un valor NULL
19	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "NIF") a un valor NULL
20	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "DIRECCION") a un valor NULL
21	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "CODPOSTAL") a un valor NULL
22	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "TELFIXO") a un valor NULL
23	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "TELMOVIL") a un valor NULL
24	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "CLIENTES", "EMAIL") a un valor NULL
25	ERROR: El código postal no existe
26	

## 9 TEST09 – TEST\_PRODUCTOS

EXEC TEST\_PETICION.TEST\_PRODUCTOS;

### PRUEBAS:

#### INS\_PRODUCTO

Se insertan dos productos. (1, 2)

Se generan los siguientes errores:

- Insertar producto con un número de serie existente. (3)
- Insertar producto con valores nulos. (4, 5, 6, 7, 8, 9, 10)
- Insertar producto con un electrodoméstico no existente. (11)
- Insertar producto con un electrodoméstico dado de baja. (12)
- Insertar producto con un código postal no existente. (13)
- Insertar producto con un cliente no existente. (14)

#### UPD\_PRODUCTO

Se modifica un producto. (15)

Se generan los siguientes errores:

- Modificar producto no existente. (16)
- Modificar producto con número de serie existente. (17)
- Modificar producto con valor nulo. (18, 19, 20, 21, 22, 23, 24)
- Modificar producto con un electrodoméstico no existente. (25)
- Modificar producto con un electrodoméstico dado de baja. (26)
- Modificar producto con un código postal no existente. (27)
- Modificar producto con un cliente no existente. (28)

### RESULTADO

1	OK. El producto se ha creado con éxito
2	OK. El producto se ha creado con éxito
3	ERROR: El número de serie del producto ya existe.
4	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "MARCA")
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "MODELO")
6	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "NUMSERIE")
7	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "FXCOMPRA")
8	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "DIRECCION")
9	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "CODPOSTAL")
10	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "PRODUCTOS", "IDCLIENTE")
11	ERROR: El identificador de electrodoméstico no existe.
12	ERROR: El electrodoméstico está de baja.
13	ERROR: El código postal no existe.
14	ERROR: El código postal no existe.
15	OK. El producto se ha modificado con éxito
16	ERROR: El identificador del producto no existe.
17	ERROR: El número de serie del producto ya existe.
18	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "MARCA") a un valor NULL
19	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "MODELO") a un valor NULL
20	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "NUMSERIE") a un valor NULL
21	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "FXCOMPRA") a un valor NULL
22	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "DIRECCION") a un valor NULL
23	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "CODPOSTAL") a un valor NULL
24	ERROR: Se ha intentado actualizar a un valor nulo ("BD", "PRODUCTOS", "IDCLIENTE") a un valor NULL
25	ERROR: El identificador de electrodoméstico no existe.
26	ERROR: El electrodoméstico está de baja.
27	ERROR: El código postal no existe.
28	ERROR: El identificador de cliente no existe
29	

## 10 TEST10 – TEST\_PETICIONES

EXEC TEST\_PETICION.TEST\_PETICIONES;

### PRUEBAS:

#### INS\_PETICION

Se insertan tres peticiones. (1, 2, 3, 4)

Se generan los siguientes errores:

- Insertar petición con un producto no existente. (5)
- Insertar petición con un servicio no existente. (6)
- Insertar petición con un servicio dado de baja. (7)
- Insertar petición con un teleoperador no existente. (8)
- Insertar petición con un teleoperador dado de baja. (9)

#### CANCEL\_PETICION

Se cancela una petición. (10)

Se generan los siguientes errores:

- Anulamos petición no existente. (11)
- Anulamos petición con estado distinto de PENDIENTE. (12)

### RESULTADO

1	OK. La petición se ha creado con éxito
2	OK. La petición se ha creado con éxito
3	OK. La petición se ha creado con éxito
4	OK. La petición se ha creado con éxito
5	ERROR: El identificador de producto no existe.
6	ERROR: El identificador de servicio no existe.
7	ERROR: El servicio está de baja.
8	ERROR: El identificador de teleoperador no existe.
9	ERROR: El teleoperador está de baja.
10	OK. La petición se ha cancelado con éxito
11	ERROR: El identificador de la petición no existe.
12	ERROR: La petición no se encuentra en estado PENDIENTE.
13	

## 11 TEST11 – TEST\_ASIGNACION

EXEC TEST\_ATENCION.TEST\_ASIGNACIONES;

### PRUEBAS:

#### ASG\_TECNICO

Se asignan tres técnicos. (1, 2, 7)

Se generan los siguientes errores:

- Asignar técnico a una petición no existente. (3)
- Asignar técnico a una petición en estado distinto de PENDIENTE. (4)
- Asignar técnico no existente a una petición. (5)
- Asignar técnico dado de baja a una petición. (6)

#### DSG\_TECNICO

Se desasigna una petición. (8)

Se generan los siguientes errores:

- Desasignar técnico de una petición no existente. (9)
- Desasignar técnico de una petición en estado distinto de ASIGNADA. (10)

### RESULTADO

1	OK. La asignación se ha realizado con éxito
2	OK. La asignación se ha realizado con éxito
3	ERROR: El identificador de petición no existe.
4	ERROR: El petición no se encuentra en estado PENDIENTE.
5	ERROR: El identificador de técnico no existe.
6	ERROR: El técnico está de baja.
7	OK. La asignación se ha realizado con éxito
8	OK. La desasignación se ha realizado con éxito
9	ERROR: El identificador de petición no existe.
10	ERROR: El petición no se encuentra en estado ASIGNADA.
11	



## 12 TEST12 – TEST\_TRABJOS

EXEC TEST\_ATENCION.TEST\_TRABAJOS;

### PRUEBAS:

#### INS\_TRABAJO

Se insertan tres trabajos. (1, 2, 3)

Se generan los siguientes errores:

- Insertar trabajo con valores nulos. (4, 5, 6)
- Insertar trabajo con una petición no existente. (7)
- Insertar trabajo con una petición en estado distinto de ASIGNADA. (8)
- Insertar trabajo con una petición asignada a otro técnico. (9)
- Insertar trabajo con un técnico no existente. (10)
- Insertar trabajo con un técnico dado de baja. (11)
- Insertar trabajo con un valor de importe no valido. (12)
- Insertar trabajo con un valor de garantía no valido. (13)

#### CLOSE\_PETICION

Se cierra una petición. (14)

Se generan los siguientes errores:

- Cerramos petición no existente. (15)
- Cerramos petición con estado distinto de ASIGNADA. (16)
- Cerramos petición sin trabajos asociados. (17)

### RESULTADO

1	OK. El trabajo se ha creado con éxito
2	OK. El trabajo se ha creado con éxito
3	OK. El trabajo se ha creado con éxito
4	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TRABAJOS", "DSTRABAJO")
5	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TRABAJOS", "IMPORTE")
6	ERROR: Se ha intentado almacenar un valor nulo. ("BD", "TRABAJOS", "GARANTIA")
7	ERROR: El identificador de petición no existe.
8	ERROR: La petición no se encuentra en estado ASIGNADA.
9	ERROR: La petición está asignada a otro técnico.
10	ERROR: El identificador de técnico no existe.
11	ERROR: El técnico está de baja.
12	ERROR: El valor de importe debe ser mayor que cero.
13	ERROR: El valor de garantía debe ser 0 o 1.
14	OK. El cierre se ha realizado con éxito
15	ERROR: El identificador de petición no existe.
16	ERROR: El petición no se encuentra en estado ASIGNADA.
17	ERROR: La petición no tiene trabajos.
18	

### 13 TEST13 – TEST\_ENCUESTAS

EXEC TEST\_ATENCION.TEST\_ENCUESTAS;

#### PRUEBAS:

INS\_ENCUESTA

Se inserta una encuesta. (1)

Se generan los siguientes errores:

- Insertar encuesta con una petición no existente. (2)
- Insertar encuesta con una petición en estado distinto de CERRADA. (3)
- Insertar encuesta con un teleoperador no existente. (4)
- Insertar trabajo con un teleoperador dado de baja. (5)
- Insertar trabajo con un valor de valoración no valido. (6)

#### RESULTADO

1	OK. La encuesta se ha creado con éxito
2	ERROR: El identificador de petición no existe.
3	ERROR: La petición no se encuentra en estado CERRADA.
4	ERROR: El identificador de teleoperador no existe.
5	ERROR: El teleoperador está de baja.
6	ERROR: El valor de valoración debe ser entre 0 y 9.
7	

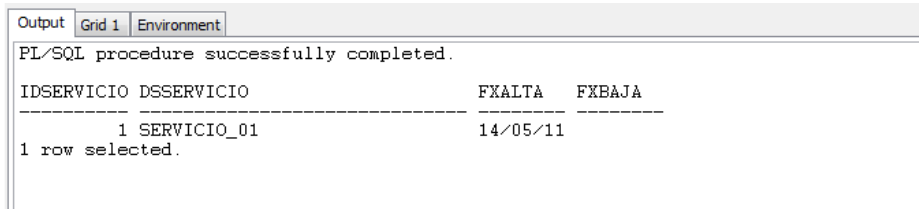
## 14 TEST14 - CONSULTAS SUBSISTEMA CONFIGURACION

### GET\_SERVICIO

Consultamos un servicio:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_SERVICIO(1, :datos);  
END  
PRINT datos;
```

Resultado:



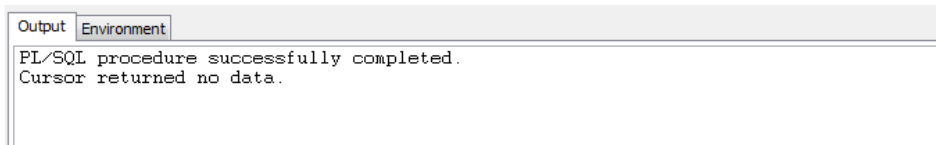
Output Grid 1 Environment

```
PL/SQL procedure successfully completed.  
IDSERVICIO DSSERVICIO                FXALTA  FXBAJA  
-----  
1 SERVICIO_01                        14/05/11  
1 row selected.
```

Consultamos un servicio no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_SERVICIO(88, :datos);  
END  
PRINT datos;
```

Resultado:



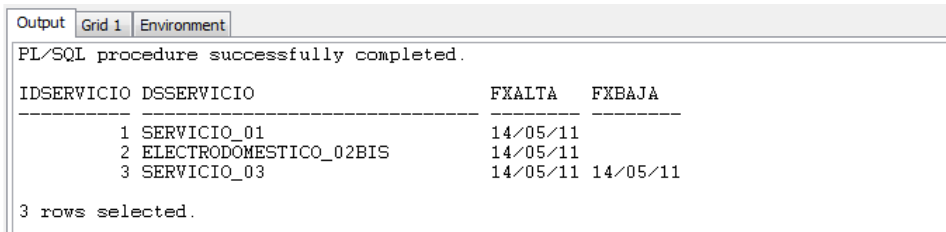
Output Environment

```
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos los servicios:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_SERVICIO(null, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.  
IDSERVICIO DSSERVICIO                FXALTA  FXBAJA  
-----  
1 SERVICIO_01                        14/05/11  
2 ELECTRODOMESTICO_02BIS            14/05/11  
3 SERVICIO_03                        14/05/11 14/05/11  
3 rows selected.
```

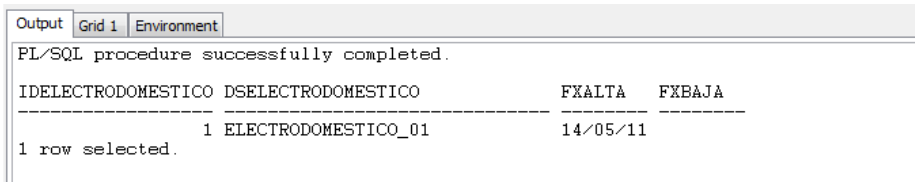
## Gestión PostVenta – Test del Sistema

### GET\_ELECTRODOMESTICO

Consultamos un electrodoméstico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ELECTRODOMESTICO(1, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

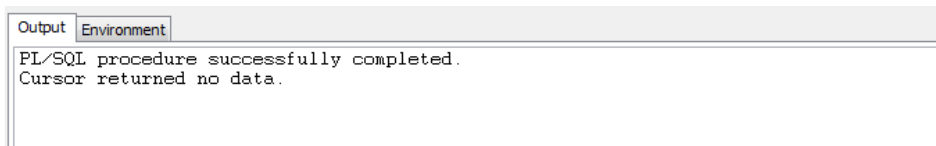
IDELECTRODOMESTICO	DSELECTRODOMESTICO	FXALTA	FXBAJA
1	ELECTRODOMESTICO_01	14/05/11	

1 row selected.

Consultamos un electrodoméstico no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ELECTRODOMESTICO(88, :datos);  
END  
PRINT datos;
```

Resultado:



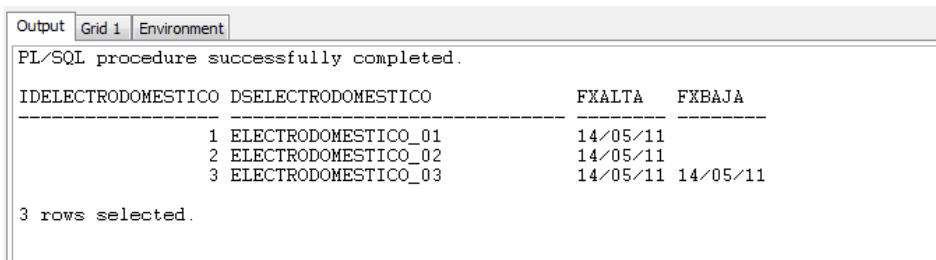
Output Environment

```
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos los electrodomésticos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ELECTRODOMESTICO(null, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

IDELECTRODOMESTICO	DSELECTRODOMESTICO	FXALTA	FXBAJA
1	ELECTRODOMESTICO_01	14/05/11	
2	ELECTRODOMESTICO_02	14/05/11	
3	ELECTRODOMESTICO_03	14/05/11	14/05/11

3 rows selected.

## Gestión PostVenta – Test del Sistema

### GET\_PROVINCIA

Consultamos una provincia:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_PROVINCIA('01', :datos);  
END  
PRINT datos;
```

Resultado:

```
Output | Grid 1 | Environment  
-----  
PL/SQL procedure successfully completed.  
COPROVINCIA DSPROVINCIA  
-----  
01          PROVINCIA_01BIS  
1 row selected.
```

Consultamos una provincia no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_PROVINCIA('88', :datos);  
END  
PRINT datos;
```

Resultado:

```
Output | Environment  
-----  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las provincias:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_PROVINCIA(null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output | Grid 1 | Environment  
-----  
PL/SQL procedure successfully completed.  
COPROVINCIA DSPROVINCIA  
-----  
01          PROVINCIA_01BIS  
03          PROVINCIA_03  
2 rows selected.
```

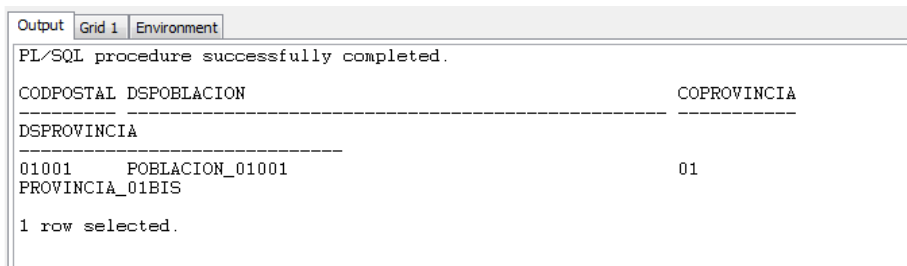
## Gestión PostVenta – Test del Sistema

### GET\_POBLACION

Consultamos una población:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACION('01001', :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

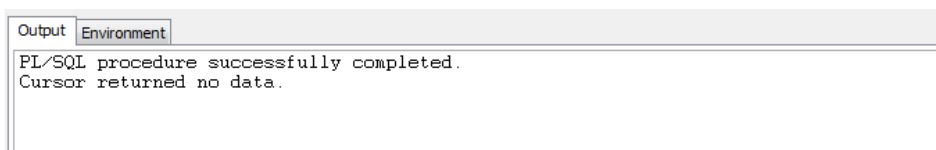
CODPOSTAL	DSPROVINCIA	DSPROVINCIA	COPROVINCIA
01001	POBLACION_01001		01
	PROVINCIA_01BIS		

1 row selected.

Consultamos una población no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACION('88888', :datos);  
END  
PRINT datos;
```

Resultado:



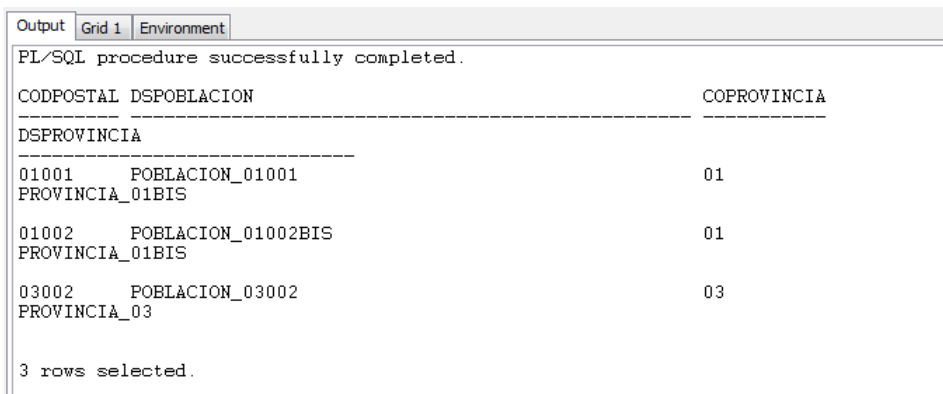
Output Environment

```
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las poblaciones:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACION(null, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

CODPOSTAL	DSPROVINCIA	DSPROVINCIA	COPROVINCIA
01001	POBLACION_01001		01
	PROVINCIA_01BIS		
01002	POBLACION_01002BIS		01
	PROVINCIA_01BIS		
03002	POBLACION_03002		03
	PROVINCIA_03		

3 rows selected.

## Gestión PostVenta – Test del Sistema

### GET\_ZONA

Consultamos una zona:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ZONA(1, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
  
IDZONA DSZONA  
-----  
1 ZONA_01  
1 row selected.
```

Consultamos una zona no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ZONA(88, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las zonas:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_ZONA(null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
  
IDZONA DSZONA  
-----  
1 ZONA_01  
2 ZONA_02BIS  
2 rows selected.
```

## Gestión PostVenta – Test del Sistema

### GET\_POBLACIONESZONA

Consultamos las poblaciones de una zona:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACIONESZONA(1, :datos);  
END  
PRINT datos;
```

Resultado:

IDZONA	DSZONA	CODPOSTAL
DSPOBLACION		COPROVINCIA
DSPROVINCIA		
2	ZONA_02BIS	01001
POBLACION_01001		01
PROVINCIA_01BIS		

1 row selected.

Consultamos las poblaciones de una zona no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACIONESZONA(88, :datos);  
END  
PRINT datos;
```

Resultado:

Output	Environment
PL/SQL procedure successfully completed. Cursor returned no data.	

Consultamos las poblaciones de todas las zonas:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.CONFIGURACION.GET_POBLACIONESZONA(null, :datos);  
END  
PRINT datos;
```

Resultado:

IDZONA	DSZONA	CODPOSTAL
DSPOBLACION		COPROVINCIA
DSPROVINCIA		
2	ZONA_02BIS	01001
POBLACION_01001		01
PROVINCIA_01BIS		

1 row selected.



## 15 TEST15 - CONSULTAS SUBSISTEMA PERSONAL

### GET\_TELEOPERADOR

Consultamos un teleoperador:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TELEOPERADOR(1, :datos);  
END  
PRINT datos;
```

Resultado:

IDTELEOPERADOR	NOMBRE	APELLIDOS	NIF	FXALTA
OPERADOR_01	1 TELE_01		11111111A	14/05/11

1 row selected.

Consultamos un teleoperador no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TELEOPERADOR(88, :datos);  
END  
PRINT datos;
```

Resultado:

Output	Environment
PL/SQL procedure successfully completed. Cursor returned no data.	

Consultamos todos los teleoperadores:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TELEOPERADOR(null, :datos);  
END  
PRINT datos;
```

Resultado:

IDTELEOPERADOR	NOMBRE	APELLIDOS	NIF	FXALTA
OPERADOR_01	1 TELE_01		11111111A	14/05/11
OPERADOR_02BIS	2 TELE_02BIS		22222222Z	14/05/11
OPERADOR_03	3 TELE_03		33333333C	14/05/11

3 rows selected.

## Gestión PostVenta – Test del Sistema

### GET\_TECNICO

Consultamos un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TECNICO(1, :datos);  
END  
PRINT datos;
```

Resultado:

Output			
Grid 1	Environment		
PL/SQL procedure successfully completed.			
IDTECNICO	NOMBRE		
-----			
APELLIDOS			NIF
-----			
TELMOVIL	FXALTA	FXBAJA	
-----			
1	TEC_01		
NICO_01			11111111A
611111111	14/05/11		
1 row selected.			

Consultamos un técnico no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TECNICO(88, :datos);  
END  
PRINT datos;
```

Resultado:

Output	
Environment	
PL/SQL procedure successfully completed.	
Cursor returned no data.	

Consultamos todos los técnicos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_TECNICO(null, :datos);  
END  
PRINT datos;
```

Resultado:

Output			
Grid 1	Environment		
PL/SQL procedure successfully completed.			
IDTECNICO	NOMBRE		
-----			
APELLIDOS			NIF
-----			
TELMOVIL	FXALTA	FXBAJA	
-----			
1	TEC_01		
NICO_01			11111111A
611111111	14/05/11		
2	TEC_02BIS		
NICO_02BIS			2222222Z
623232323	14/05/11		
3	TEC_03		
NICO_03			3333333C
633333333	14/05/11	14/05/11	
3 rows selected.			

## Gestión PostVenta – Test del Sistema

### GET\_SERVICIOSTECNICO

Consultamos un servicio de un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_SERVICIOSTECNICO(1, 1, :datos);  
END  
PRINT datos;
```

Resultado:

IDTECNICO	NOMBRE	IDSERVICIO
1	TEC_01	
NICO_01		1
SERVICIO_01		

1 row selected.

Consultamos los técnicos asignados a un servicio:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_SERVICIOSTECNICO(1, null, :datos);  
END  
PRINT datos;
```

Resultado:

IDTECNICO	NOMBRE	IDSERVICIO
1	TEC_01	
NICO_01		1
SERVICIO_01		
2	TEC_02BIS	
NICO_02BIS		1
SERVICIO_01		

2 rows selected.

Consultamos los servicios asignados a un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_SERVICIOSTECNICO(null, 1, :datos);  
END  
PRINT datos;
```

Resultado:

IDTECNICO	NOMBRE	IDSERVICIO
1	TEC_01	
NICO_01		1
SERVICIO_01		

1 row selected.

## Gestión PostVenta – Test del Sistema

Consultamos obteniendo un cursor vacío:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_SERVICIOSTECNICO(88, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos los servicios de todos los técnicos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_SERVICIOSTECNICO(null, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
-----  
APELIDOS                                IDSERVICIO  
-----  
DSSERVICIO  
-----  
1 TEC_01  
NICO_01                                1  
SERVICIO_01  
2 TEC_02BIS  
NICO_02BIS                             1  
SERVICIO_01  
  
2 rows selected.
```

## Gestión PostVenta – Test del Sistema

### GET\_ELECTRODOMESTICOSTECNICO

Consultamos un electrodoméstico de un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ELECTRODOMESTICOTECNICO(1, 1, :datos);  
END  
PRINT datos;
```

Resultado:

Output		
Grid 1	Environment	
PL/SQL procedure successfully completed.		
IDTECNICO NOMBRE		
-----		
APELLIDOS		IDELECTRODOMESTICO
-----		-----
DSELECTRODOMESTICO		
-----		
1 TEC_01		
NICO_01		1
ELECTRODOMESTICO_01		
1 row selected.		

Consultamos los técnicos asignados a un electrodoméstico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ELECTRODOMESTICOTECNICO(1, null, :datos);  
END  
PRINT datos;
```

Resultado:

Output		
Grid 1	Environment	
PL/SQL procedure successfully completed.		
APELLIDOS		IDELECTRODOMESTICO
-----		-----
DSELECTRODOMESTICO		
-----		
1 TEC_01		
NICO_01		1
ELECTRODOMESTICO_01		
2 TEC_02BIS		
NICO_02BIS		1
ELECTRODOMESTICO_01		
2 rows selected.		

Consultamos los electrodomésticos asignados a un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ELECTRODOMESTICOSTECNICO(null, 1, :datos);  
END  
PRINT datos;
```

Resultado:

Output		
Grid 1	Environment	
PL/SQL procedure successfully completed.		
IDTECNICO NOMBRE		
-----		
APELLIDOS		IDELECTRODOMESTICO
-----		-----
DSELECTRODOMESTICO		
-----		
1 TEC_01		
NICO_01		1
ELECTRODOMESTICO_01		
1 row selected.		

## Gestión PostVenta – Test del Sistema

Consultamos obteniendo un cursor vacío:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ELECTRODOMESTICOSTECNICO(88, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos los electrodomésticos de todos los técnicos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ELECTRODOMESTICOSTECNICO(null, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
APELIDOS ----- IDELECTRODOMESTICO  
DSELECTRODOMESTICO  
-----  
1 TEC_01  
NICO_01 ----- 1  
ELECTRODOMESTICO_01  
2 TEC_02BIS  
NICO_02BIS ----- 1  
ELECTRODOMESTICO_01  
  
2 rows selected.
```

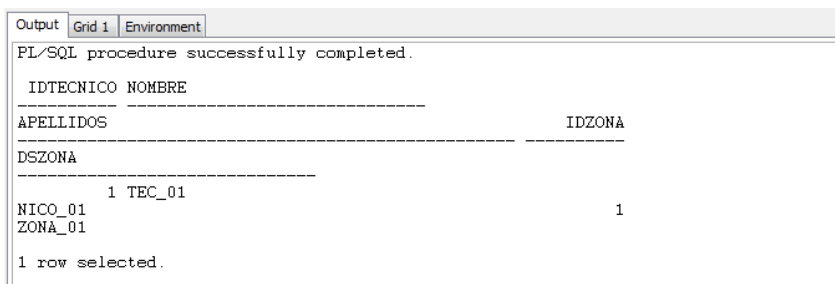
## Gestión PostVenta – Test del Sistema

### GET\_ZONASTECNICO

Consultamos una zona de un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ZONASTECNICO(1, 1, :datos);  
END  
PRINT datos;
```

Resultado:

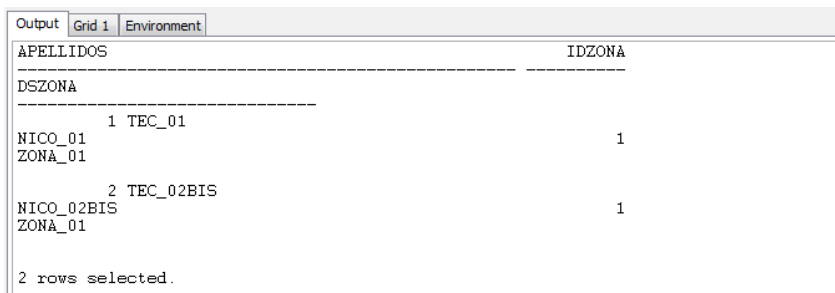


```
Output | Grid 1 | Environment  
-----  
PL/SQL procedure successfully completed.  
  
  IDTECNICO NOMBRE  
-----  
APELLIDOS                                     IDZONA  
-----  
DSZONA  
-----  
      1 TEC_01  
NICO_01                                     1  
ZONA_01  
  
1 row selected.
```

Consultamos los técnicos asignados a una zona:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ZONASTECNICO(1, null, :datos);  
END  
PRINT datos;
```

Resultado:

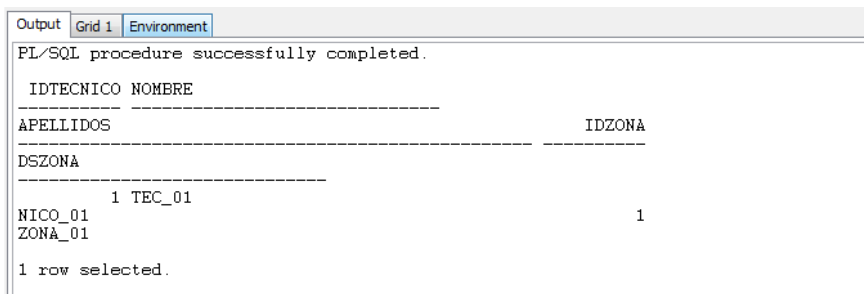


```
Output | Grid 1 | Environment  
-----  
APELLIDOS                                     IDZONA  
-----  
DSZONA  
-----  
      1 TEC_01  
NICO_01                                     1  
ZONA_01  
  
      2 TEC_02BIS  
NICO_02BIS                                  1  
ZONA_01  
  
2 rows selected.
```

Consultamos las zonas asignadas a un técnico:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ZONASTECNICO(null, 1, :datos);  
END  
PRINT datos;
```

Resultado:



```
Output | Grid 1 | Environment  
-----  
PL/SQL procedure successfully completed.  
  
  IDTECNICO NOMBRE  
-----  
APELLIDOS                                     IDZONA  
-----  
DSZONA  
-----  
      1 TEC_01  
NICO_01                                     1  
ZONA_01  
  
1 row selected.
```

## Gestión PostVenta – Test del Sistema

Consultamos obteniendo un cursor vacío:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ZONASTECNICO(88, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las zonas de todos los técnicos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_ZONASTECNICO(null, null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
-----  
APELLIDOS                                IDZONA  
-----  
DSZONA  
-----  
1 TEC_01  
NICO_01                                    1  
ZONA_01  
-----  
2 TEC_02BIS  
NICO_02BIS                                1  
ZONA_01  
-----  
2 rows selected.
```



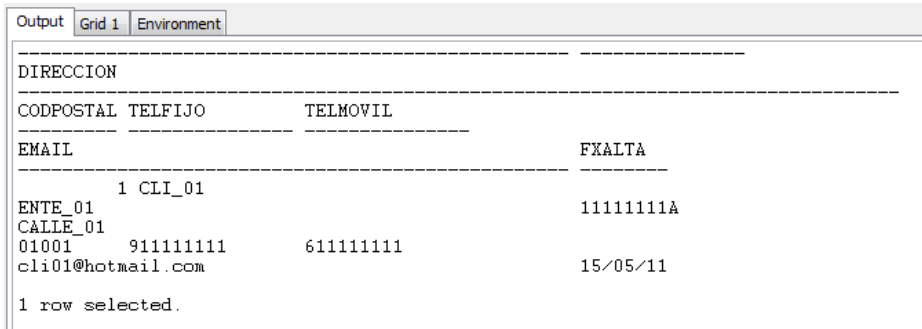
## 16 TEST16 - CONSULTAS SUBSISTEMA PETICIÓN

### GET\_CLIENTE

Consultamos un cliente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_CLIENTE(1, :datos);  
END  
PRINT datos;
```

Resultado:



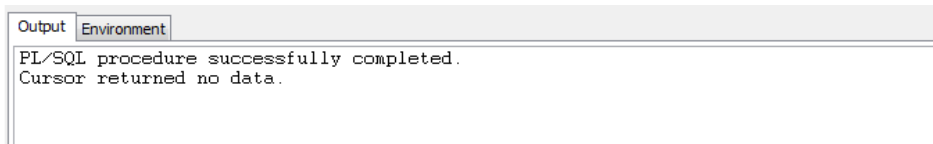
The screenshot shows the output window of SQL Developer with the following content:

```
Output Grid 1 Environment  
-----  
DIRECCION  
-----  
CODPOSTAL TELFIJO          TELMOVIL  
-----  
EMAIL                      FXALTA  
-----  
1 CLI_01  
ENTE_01                    11111111A  
CALLE_01  
01001  911111111          611111111  
cli01@hotmail.com          15/05/11  
  
1 row selected.
```

Consultamos un cliente no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_CLIENTE(88, :datos);  
END  
PRINT datos;
```

Resultado:



The screenshot shows the output window of SQL Developer with the following content:

```
Output Environment  
-----  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos los servicios:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_CLIENTE(null, :datos);  
END  
PRINT datos;
```

# Gestión PostVenta – Test del Sistema

Resultado:

```
Output Grid 1 Environment
PL/SQL procedure successfully completed.
  IDCLIENTE NOMBRE
-----
 APELLIDOS                                NIF
-----
 DIRECCION
-----
 CODPOSTAL TELFIJO      TELMOVIL
-----
 EMAIL                                FXALTA
-----
      1 CLI_01                                11111111A
ENTE_01
CALLE_01
01001  911111111      611111111
cli01@hotmail.com      15/05/11
      2 CLI_02BIS                             22222222Z
ENTE_02BIS
CALLE_02BIS
01001  923232323      623232323
cli_02@hotmail.com      15/05/11
      3 CLI_03
```

```
Output Grid 1 Environment
  IDCLIENTE NOMBRE
-----
 APELLIDOS                                NIF
-----
 DIRECCION
-----
 CODPOSTAL TELFIJO      TELMOVIL
-----
 EMAIL                                FXALTA
-----
ENTE_03                                33333333C
CALLE_03
01001  933333333      633333333
cli03@hotmail.com      15/05/11

3 rows selected.
```

## Gestión PostVenta – Test del Sistema

### GET\_PRODUCTO

Consultamos un producto:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PRODUCTO(1, :datos);  
END  
PRINT datos;
```

Resultado:

MODELO	NUMSERIE	FXCOMPRA
DIRECCION		
CODPOSTAL IDCLIENTE		
1	1 MARCA_01	
MODELO_01	SERIE_01	01/01/11
DIRECCION_01		
01001		1

1 row selected.

Consultamos un producto no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PRODUCTO(88, :datos);  
END  
PRINT datos;
```

Resultado:

Output	Environment
PL/SQL procedure successfully completed. Cursor returned no data.	

Consultamos todos los productos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PRODUCTO(null, :datos);  
END  
PRINT datos;
```

Resultado:

IDPRODUCTO	IDELECTRODOMESTICO	MARCA
MODELO		
NUMSERIE		
FXCOMPRA		
DIRECCION		
CODPOSTAL IDCLIENTE		
1	1 MARCA_01	
MODELO_01	SERIE_01	01/01/11
DIRECCION_01		
01001		1
2	1 MARCA_02BIS	
MODELO_02BIS	SERIE_02BIS	20/02/11
DIRECCION_02BIS		
01001		1

2 rows selected.

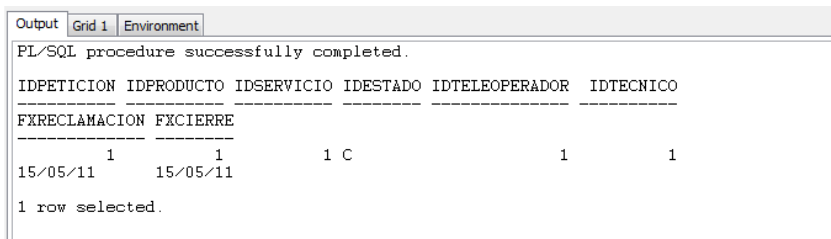
## Gestión PostVenta – Test del Sistema

### GET\_PETICION

Consultamos una petición:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICION(1, :datos);  
END  
PRINT datos;
```

Resultado:

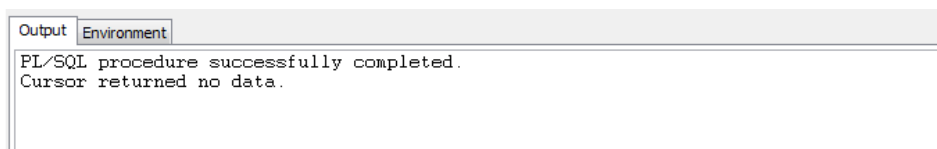


```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDPETICION IDPRODUCTO IDSERVICIO IDESTADO IDTELEOPERADOR IDTECNICO  
-----  
FXRECLAMACION FXCIERRE  
-----  
15/05/11 1 1 1 C 1 1  
1 row selected.
```

Consultamos una petición no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICION(88, :datos);  
END  
PRINT datos;
```

Resultado:

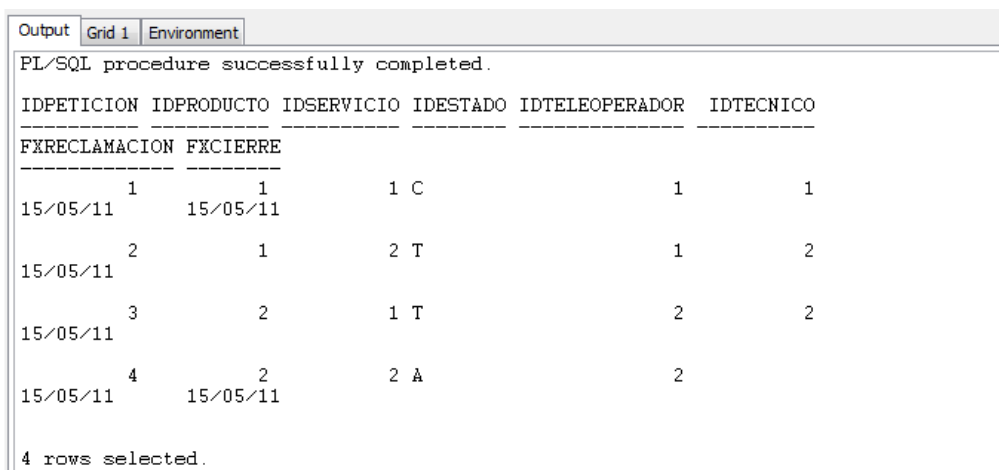


```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las peticiones:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICION(null, :datos);  
END  
PRINT datos;
```

Resultado:



```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDPETICION IDPRODUCTO IDSERVICIO IDESTADO IDTELEOPERADOR IDTECNICO  
-----  
FXRECLAMACION FXCIERRE  
-----  
15/05/11 1 1 1 C 1 1  
15/05/11 2 1 2 T 1 2  
15/05/11 3 2 1 T 2 2  
15/05/11 4 2 2 A 2  
4 rows selected.
```

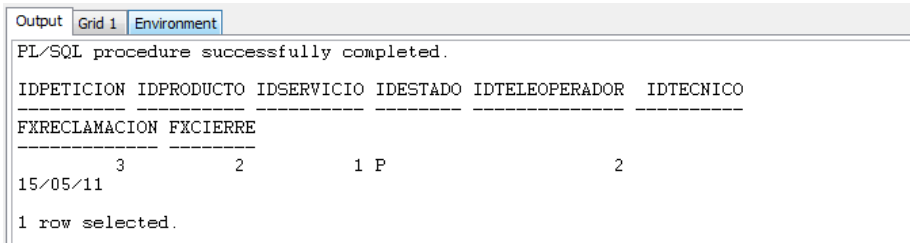
## 17 TEST17 - CONSULTAS SUBSISTEMA ATENCIÓN

### GET\_PETICIONESERVICIOZONA

Consultamos peticiones de un servicio y una zona:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.ATENCION.GET_PETICIONESERVICIOZONA(1, 2, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

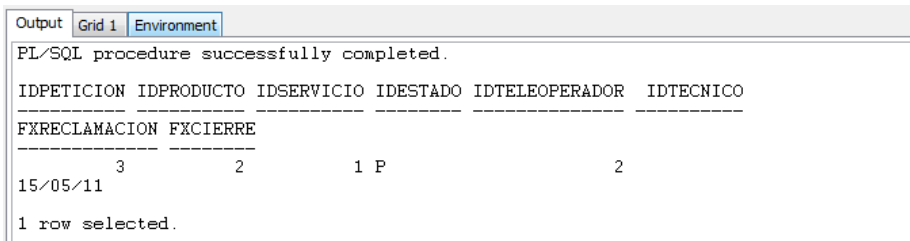
IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
FXRECLAMACION	FXCIERRE				
15/05/11	3	2	1 P		2

1 row selected.

Consultamos las peticiones de un servicio:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_PETICIONESERVICIOZONA (1, null, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

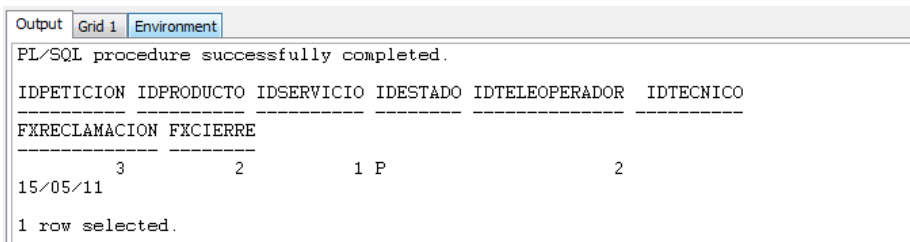
IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
FXRECLAMACION	FXCIERRE				
15/05/11	3	2	1 P		2

1 row selected.

Consultamos las peticiones de una zona:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_PETICIONESERVICIOZONA (null, 2, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
FXRECLAMACION	FXCIERRE				
15/05/11	3	2	1 P		2

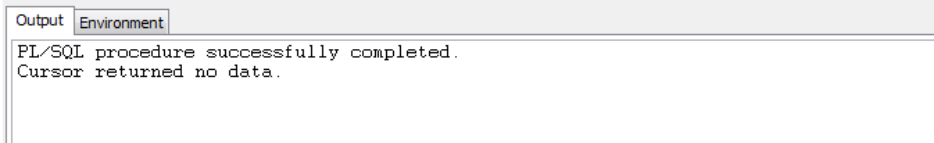
1 row selected.

## Gestión PostVenta – Test del Sistema

Consultamos obteniendo un cursor vacío:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_PETICIONESSERVICIOZONA (88, null, :datos);  
END  
PRINT datos;
```

Resultado:

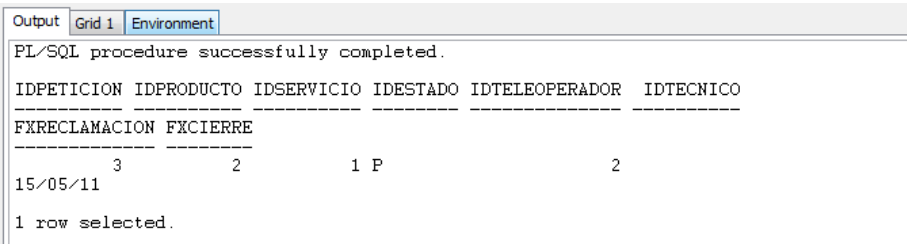


```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las peticiones:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PERSONAL.GET_PETICIONESSERVICIOZONA (null, null, :datos);  
END  
PRINT datos;
```

Resultado:



```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDPETICION IDPRODUCTO IDSERVICIO IDESTADO IDTELEOPERADOR IDTECNICO  
-----  
FXRECLAMACION FXCIERRE  
-----  
15/05/11 3 2 1 P 2  
1 row selected.
```

## Gestión PostVenta – Test del Sistema

### GET\_PETICIONESTECNICO

Consultamos una petición:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICIONESTECNICO(1, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDPETICION IDPRODUCTO IDSERVICIO IDESTADO IDTELEOPERADOR IDTECNICO  
-----  
FXRECLAMACION FXCIERRE  
-----  
3 2 2 P 2  
15/05/11  
1 row selected.
```

Consultamos peticiones para un técnico no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICIONTECNICO(88, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todas las peticiones:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_PETICIONTECNICO(null, :datos);  
END  
PRINT datos;
```

Resultado:

```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDPETICION IDPRODUCTO IDSERVICIO IDESTADO IDTELEOPERADOR IDTECNICO  
-----  
FXRECLAMACION FXCIERRE  
-----  
3 2 2 P 2  
15/05/11  
1 row selected.
```

## Gestión PostVenta – Test del Sistema

### GET\_TECNICOASIGNADO

Consultamos el técnico asignado a una petición:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_TECNICOASIGNADO(2, :datos);  
END  
PRINT datos;
```

Resultado:

IDTECNICO	NOMBRE	APELLIDOS	NIF	TELMOVIL	FXALTA	FXBAJA
2	TEC_02BIS		22222222Z			
NICO_02BIS				623232323	14/05/11	

1 row selected.

Consultamos técnico asignado para una petición no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_TECNICOASIGNADO (88, :datos);  
END  
PRINT datos;
```

Resultado:

Output	Environment
PL/SQL procedure successfully completed. Cursor returned no data.	



## Gestión PostVenta – Test del Sistema

### GET\_TRABAJO

Consultamos un trabajo:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_TRABAJO(1, :datos);  
END  
PRINT datos;
```

Resultado:

IDTRABAJO	IDPETICION	IDTECNICO	FXTRABAJO
DSTRABAJO			
-----			
IMPORTE	GARANTIA		
-----			
1	1	1	15/05/11
TRABAJO_01			
1000	0		

1 row selected.

Consultamos un trabajo no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_TRABAJO(88, :datos);  
END  
PRINT datos;
```

Resultado:

Output Environment			
PL/SQL procedure successfully completed.			
Cursor returned no data.			

Consultamos todos los trabajos:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_TRABAJO(null, :datos);  
END  
PRINT datos;
```

Resultado:

IDTRABAJO	IDPETICION	IDTECNICO	FXTRABAJO
DSTRABAJO			
-----			
IMPORTE	GARANTIA		
-----			
1	1	1	15/05/11
TRABAJO_01			
1000	0		
2	1	1	15/05/11
TRABAJO_02			
100	1		
3	2	2	15/05/11
TRABAJO_03			
500	1		

3 rows selected.

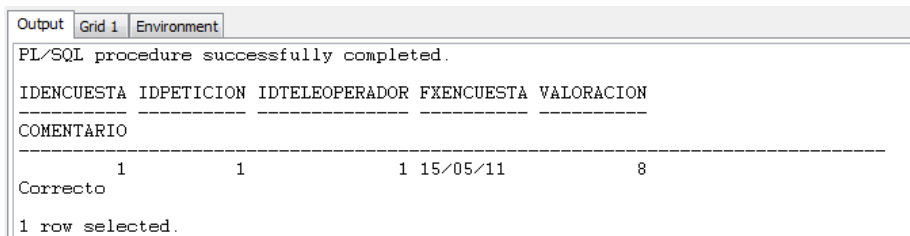
## Gestión PostVenta – Test del Sistema

### GET\_ENCUESTA

Consultamos una encuesta:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_ENCUESTA(1, :datos);  
END  
PRINT datos;
```

Resultado:

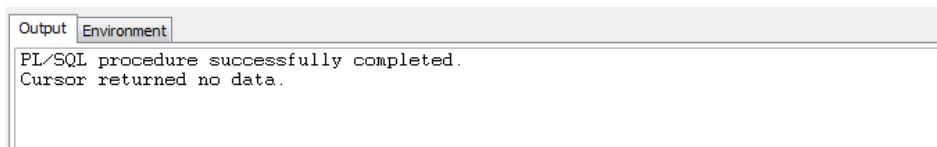


```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDENCUESTA IDPETICION IDTELEOPERADOR FXENCUESTA VALORACION  
-----  
COMENTARIO  
-----  
1 1 1 15/05/11 8  
Correcto  
1 row selected.
```

Consultamos una encuesta no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_ENCUESTA(88, :datos);  
END  
PRINT datos;
```

Resultado:

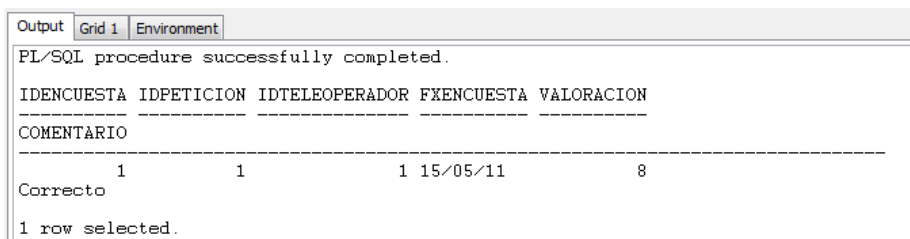


```
Output Environment  
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

Consultamos todos las encuestas:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.PETICION.GET_ENCUESTA(null, :datos);  
END  
PRINT datos;
```

Resultado:



```
Output Grid 1 Environment  
PL/SQL procedure successfully completed.  
IDENCUESTA IDPETICION IDTELEOPERADOR FXENCUESTA VALORACION  
-----  
COMENTARIO  
-----  
1 1 1 15/05/11 8  
Correcto  
1 row selected.
```

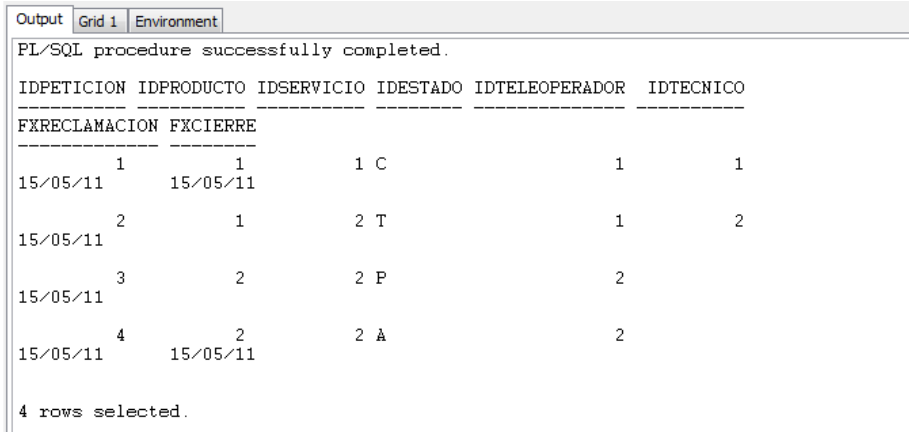
## 18 TEST18 - CONSULTAS SUBSISTEMA HISTORIAL

### GET\_HCLIENTE

Consultamos las peticiones de un cliente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HCLIENTE(1, :datos);  
END  
PRINT datos;
```

Resultado:



Output Grid 1 Environment

```
PL/SQL procedure successfully completed.
```

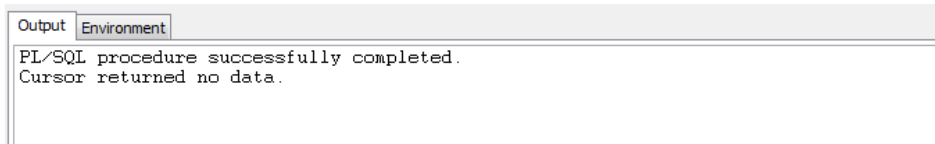
IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
FXRECLAMACION	FXCIERRE				
15/05/11	1	1	1 C	1	1
15/05/11	2	1	2 T	1	2
15/05/11	3	2	2 P	2	
15/05/11	4	2	2 A	2	

4 rows selected.

Consultamos las peticiones de un cliente no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HCLIENTE (88, :datos);  
END  
PRINT datos;
```

Resultado:



Output Environment

```
PL/SQL procedure successfully completed.  
Cursor returned no data.
```

## Gestión PostVenta – Test del Sistema

### GET\_HPRODUCTO

Consultamos las peticiones de un producto:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HPRODUCTO(1, :datos);  
END  
PRINT datos;
```

Resultado:

IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
FXRECLAMACION	FXCIERRE				
15/05/11	1	1	1 C	1	1
15/05/11	2	1	2 T	1	2

2 rows selected.

Consultamos las peticiones de un producto no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HPRODUCTO (88, :datos);  
END  
PRINT datos;
```

Resultado:

IDPETICION	IDPRODUCTO	IDSERVICIO	IDESTADO	IDTELEOPERADOR	IDTECNICO
------------	------------	------------	----------	----------------	-----------

PL/SQL procedure successfully completed.  
Cursor returned no data.

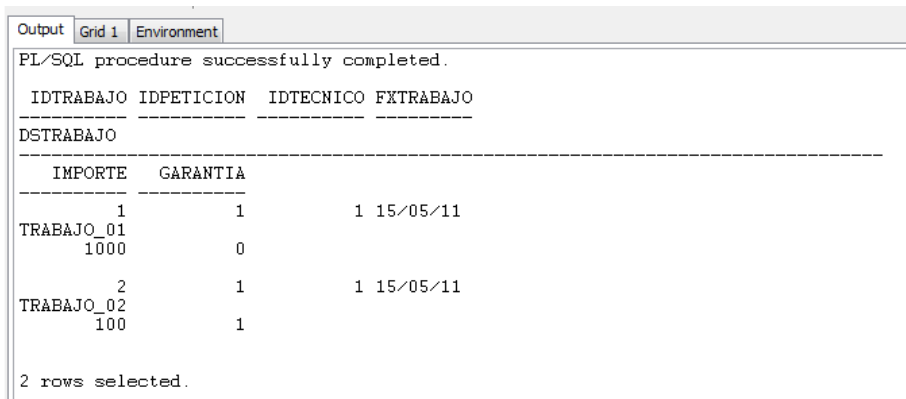
## Gestión PostVenta – Test del Sistema

### GET\_HTECNICO

Consultamos las peticiones de un producto:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HTECNICO(1, :datos);  
END  
PRINT datos;
```

Resultado:



The screenshot shows the output window of SQL Developer. The title bar includes 'Output', 'Grid 1', and 'Environment'. The main content area displays the following text:

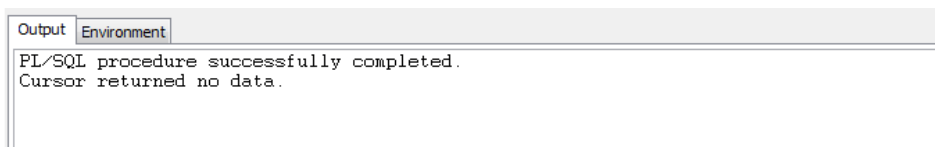
```
PL/SQL procedure successfully completed.  
  
  IDTRABAJO  IDPETICION  IDTECNICO  EXTRABAJO  
-----  
DSTRABAJO  
-----  
  IMPORTE    GARANTIA  
-----  
          1          1          1 15/05/11  
TRABAJO_01  
    1000          0  
          2          1          1 15/05/11  
TRABAJO_02  
    100          1
```

At the bottom of the output, it states: 2 rows selected.

Consultamos las peticiones de un producto no existente:

```
VARIABLE datos REFCURSOR;  
BEGIN  
    BD.HISTORIAL.GET_HTECNICO (88, :datos);  
END  
PRINT datos;
```

Resultado:



The screenshot shows the output window of SQL Developer. The title bar includes 'Output' and 'Environment'. The main content area displays the following text:

```
PL/SQL procedure successfully completed.  
Cursor returned no data.
```