



TFG – PRESENTACIÓN 2014

Sistema BBDD – Gestión del Consumo energético

El trabajo propuesto consiste en implementar un sistema de BD para dar respuesta a una necesidad planteada para la comisión europea de la energía que necesita tener controlada la gestión del consumo energético que hacen las diferentes compañías implicadas en el suministro y control de este servicio.

Yago Blanco Presas – TFG
PRESENTACIÓN 2014



Contenido Presentación

1. Creación estructura Base de datos.....	4
TFG_BBDD_Esquema.sql	4
TFG_BBDD_tablespace.sql	7
2. Inicialización Base de datos	8
Inicializacion.sql.....	8
3. Creación Procedimientos ABM.....	11
P_alta_ciudad.sql.....	12
P_alta_consumidor.sql.....	12
P_alta_contador.sql	13
P_alta_empresas.sql.....	13
P_alta_logConexion.sql.....	13
P_alta_pais.sql	14
P_alta_precioEnergia.sql	14
P_baja_ciudad.sql	14
P_baja_consumidor.sql.....	15
P_baja_contador.sql.....	15
P_baja_empresa.sql	15
P_baja_logConexion,sql	16
P_baja_pais.sql	16
P_baja_PrecioEnergia.sql.....	16
P_modificacion_ciudad.sql	17
P_modificacion_consumidor.sql	17
P_modificacion_contador.sql	17
P_modificacion_empresas.sql	18
P_modificacion_logConexion.sql	18
P_modificacion_pais.sql.....	19
P_modificacion_precioEnergia.sql.....	19
4. Creación procedimientos de Consulta/Estadística.....	20
P_Cons_Elect_Ciudad.sql	20
P_Consumo_Medio.sql	20
P_ConsumoElectrico_Mensual.sql.....	21

P_List_Cont_Antig.sql	21
P_List_Cont_Cons_Mensual.sql	22
P_Porcentaje_Lecturas.sql.....	22
P_Top10_Contadores.sql.....	23
5. LOGS EJECUCIONES PROCEDIMIENTOS	24
5.1. LOG EJECUCION PROCEDIMIENTOS ABM	24
select p_alta_ciudad('Albania');.....	24
select p_alta_consumidor('1111111F','Londres, 334 2°2º','Sevilla','Espanya',1,'ES422221443234444',1,1);.....	24
select p_alta_contador('Modelo 1',11,300,'02/01/2014','02/01/2014',11);	24
select p_alta_empresa('1111111A','Sonic S.L.','Avda Madrid, 233 2°1º','Espanya');.....	24
select p_alta_logConexion('02/01/2014','02:01:05',1,550,'OK');	25
select p_alta_pais('Canada4');.....	25
select p_alta_precioEnergia('02/01/2014','10:30:00','Espanya',350.34);.....	25
select p_baja_ciudad('Albania');.....	25
select p_baja_consumidor('1111111B');.....	26
select p_baja_contador(1,1);.....	26
select p_baja_empresa('1111111A');	26
select p_baja_logConexion('10/01/2014','10:30:45',1);.....	26
select p_baja_pais('Canada');.....	26
select p_baja_precioEnergia('02/01/2014','10:30:00','Espanya');.....	27
select p_modificacion_ciudad(1,'Madrid');	27
select p_modificacion_consumidor('1111111F','Londres, 334 2°2º','Sevilla','Espanya',1,'ES422221443234444',1,1);.....	27
select p_modificacion_contador('Modelo 10000',10,300,'02/01/2014','02/01/2014',10);.....	27
select p_modificacion_empresa('1111111A','Sonic S.L.A.','Avda Madrid, 233 2°1º','Espanya');.....	28
select p_modificacion_logConexion('10/02/2014','12:15:45',1,1000,'KO');	28
select p_modificacion_pais(2,'Canada');	28
select p_modificacion_precioEnergia('12/01/2014','00:40:05','Irlanda',12350.34);.....	28
5.2. LOG EJECUCION PROCEDIMIENTOS CONSULTA/ESTADÍSTICA	29
select * from p_Cons_Elect_Ciudad('02','Espanya');	29
select * from p_Consumo_Medio('Sevilla','2');.....	29
select * from p_Consumo_Electrico_Mensual();	29
select * from p_List_Cont_Antig(2);	30

select * from p_Top10_Contadores('Barcelona'); 30

1. Creación estructura Base de datos

TFG_BBDD_Eschema.sql

Procedemos a construir la estructura de la base de datos mediante la ejecución del script TFG_BBDD_Eschema.sql

```
[QUERY ] DROP TABLE "Países" cascade
NOTICE: eliminando además restricción «FK_Pais» en tabla "Consumidores"
[QUERY ] CREATE TABLE "Países"
(
    "IdPais" serial CONSTRAINT "PK_ID_PAIS" PRIMARY KEY USING INDEX TABLESPACE "TFG_BBDD_IDX"
NOT NULL,
    "NombrePais" character(50)
)
WITH (
    OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE creará una secuencia implícita «Países_IdPais_seq» para la columna serial
«Países.IdPais»
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «PK_ID_PAIS» para la tabla «Países»
[QUERY ] ALTER TABLE "Países"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Países"
IS 'Tabla de países'
[QUERY ] COMMENT ON COLUMN "Países"."IdPais" IS 'Identificador del país'
[QUERY ] COMMENT ON COLUMN "Países"."NombrePais" IS 'Nombre del país'
[QUERY ] DROP TABLE "Ciudades" cascade
[QUERY ] CREATE TABLE "Ciudades"
(
    "IdCiudad" serial CONSTRAINT "PK_IdCiudad" PRIMARY KEY USING INDEX TABLESPACE
"TFG_BBDD_IDX" NOT NULL,
    "NombreCiudad" character(50)
)
WITH (
    OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE creará una secuencia implícita «Ciudades_IdCiudad_seq» para la columna serial
«Ciudades.IdCiudad»
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «PK_IdCiudad» para la tabla «Ciudades»
[QUERY ] ALTER TABLE "Ciudades"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Ciudades"
IS 'Tabla de ciudades'
[QUERY ] COMMENT ON COLUMN "Ciudades"."IdCiudad" IS 'Identificador de la ciudad'
[QUERY ] COMMENT ON COLUMN "Ciudades"."NombreCiudad" IS 'Nombre de la ciudad'
[QUERY ] DROP TABLE "Contadores" cascade
NOTICE: eliminando además 2 objetos más
DETAIL: eliminando además restricción «FK_NumeroSerieContador» en tabla "LogConexion"
eliminando además restricción «FK_Consumidores» en tabla "Consumidores"
[QUERY ] CREATE TABLE "Contadores"
(
    "IdContador" serial CONSTRAINT "PK_IdContador" PRIMARY KEY USING INDEX TABLESPACE
"TFG_BBDD_IDX" NOT NULL,
    "Modelo" character(50),
    "CodigoContratoAsociado" integer NOT NULL,
    "Potencia" integer,
    "FechaUltimaInspeccion" date NOT NULL,
    "FechaInstalacion" date NOT NULL,
```

```

"NumeroSerie" integer NOT NULL,
CONSTRAINT "UN_CodigoContratoAsociado" UNIQUE ("CodigoContratoAsociado" )
USING INDEX TABLESPACE "TFG_BBDD_IDX",
CONSTRAINT "UN_NumeroSerie" UNIQUE ("NumeroSerie" )
USING INDEX TABLESPACE "TFG_BBDD_IDX"
)
WITH (
  OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE crear  una secuencia impl cita «Contadores_IdContador_seq» para la columna
serial «Contadores.IdContador»
NOTICE: CREATE TABLE / PRIMARY KEY crear  el  ndice impl cito «PK_IdContador» para la tabla
«Contadores»
NOTICE: CREATE TABLE / UNIQUE crear  el  ndice impl cito «UN_CodigoContratoAsociado» para la tabla
«Contadores»
NOTICE: CREATE TABLE / UNIQUE crear  el  ndice impl cito «UN_NumeroSerie» para la tabla «Contadores»
[QUERY ] ALTER TABLE "Contadores"
  OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Contadores"
  IS 'Tabla de contadores'
[QUERY ] COMMENT ON COLUMN "Contadores"."IdContador" IS 'Identificador del contador'
[QUERY ] COMMENT ON COLUMN "Contadores"."Modelo" IS 'Modelo del contador'
[QUERY ] COMMENT ON COLUMN "Contadores"."CodigoContratoAsociado" IS 'Codigo del contrato asociado'
[QUERY ] COMMENT ON COLUMN "Contadores"."Potencia" IS 'Potencia en Kilowatios'
[QUERY ] COMMENT ON COLUMN "Contadores"."FechaUltimaInspeccion" IS 'Fecha  ltima inspecci n'
[QUERY ] COMMENT ON COLUMN "Contadores"."FechaInstalacion" IS 'Fecha de instalaci n'
[QUERY ] COMMENT ON COLUMN "Contadores"."NumeroSerie" IS 'N mero de Serie del contador'
[QUERY ] DROP TABLE "PrecioEnergia" cascade
[QUERY ] CREATE TABLE "PrecioEnergia"
(
  "Fecha" date NOT NULL,
  "Hora" time(6) with time zone NOT NULL,
  "Pais" character(30) NOT NULL,
  "PrecioKw" real,
  CONSTRAINT "PK_PrecioEnergia" PRIMARY KEY ("Fecha" , "Hora" , "Pais" )
  USING INDEX TABLESPACE "TFG_BBDD_IDX"
)
WITH (
  OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE / PRIMARY KEY crear  el  ndice impl cito «PK_PrecioEnergia» para la tabla
«PrecioEnergia»
[QUERY ] ALTER TABLE "PrecioEnergia"
  OWNER TO postgres
[QUERY ] COMMENT ON TABLE "PrecioEnergia"
  IS 'Tabla Precios Energia'
[QUERY ] COMMENT ON COLUMN "PrecioEnergia"."Fecha" IS 'Fecha Precio Energia'
[QUERY ] COMMENT ON COLUMN "PrecioEnergia"."Hora" IS 'Hora Precio Energia'
[QUERY ] COMMENT ON COLUMN "PrecioEnergia"."Pais" IS 'Pais Precio Energia'
[QUERY ] COMMENT ON COLUMN "PrecioEnergia"."PrecioKw" IS 'Precio en Kilowatios del precio de la energia'
[QUERY ] DROP TABLE "LogConexion" cascade
[QUERY ] CREATE TABLE "LogConexion"
(
  "IdLogConexion" serial CONSTRAINT "PK_IdLogConexion" PRIMARY KEY USING INDEX TABLESPACE
  "TFG_BBDD_IDX" NOT NULL,
  "FechaConexion" date NOT NULL,
  "HoraConexion" time(6) with time zone NOT NULL,
  "NumeroSerieContador" integer NOT NULL,
  "ConsumoContador" real,
  "ResultatLectura" character(2) NOT NULL,
  CONSTRAINT "FK_NumeroSerieContador" FOREIGN KEY ("NumeroSerieContador")
    REFERENCES "Contadores" ("NumeroSerie") MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
  OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE crear  una secuencia impl cita «LogConexion_IdLogConexion_seq» para la
columna serial «LogConexion.IdLogConexion»

```

NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «PK_IdLogConexion» para la tabla «LogConexion»

```
[QUERY ] ALTER TABLE "LogConexion"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "LogConexion"
IS 'Tabla Log Conexion'
[QUERY ] COMMENT ON COLUMN "LogConexion"."IdLogConexion" IS 'Identificador del log de conexion'
[QUERY ] COMMENT ON COLUMN "LogConexion"."FechaConexion" IS 'Fecha de conexion del log'
[QUERY ] COMMENT ON COLUMN "LogConexion"."HoraConexion" IS 'Hora del log de conexion'
[QUERY ] COMMENT ON COLUMN "LogConexion"."NumeroSerieContador" IS 'Numero de Serie del contador'
[QUERY ] COMMENT ON COLUMN "LogConexion"."ConsumoContador" IS 'Consumo del contador'
[QUERY ] COMMENT ON COLUMN "LogConexion"."ResultatLectura" IS 'Resultado de la lectura ("OK","KO")'
[QUERY ] DROP TABLE "Empresas" cascade
NOTICE: eliminando además restricción «FK_EmpresaSubministradora» en tabla "Consumidores"
[QUERY ] CREATE TABLE "Empresas"
(
    "IdEmpresa" serial CONSTRAINT "PK_Empresa" PRIMARY KEY USING INDEX TABLESPACE
"TFG_BBDD_IDX" NOT NULL,
    "NIF" character(12) NOT NULL,
    "Nombre" character(50),
    "DomicilioFiscal" character(50),
    "Pais" character(50)
)
WITH (
    OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE creará una secuencia implícita «Empresas_IdEmpresa_seq» para la columna serial
«Empresas.IdEmpresa»
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «PK_Empresa» para la tabla «Empresas»
[QUERY ] ALTER TABLE "Empresas"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Empresas"
IS 'Tabla de empresas'
[QUERY ] COMMENT ON COLUMN "Empresas"."IdEmpresa" IS 'Identificador de la empresa'
[QUERY ] COMMENT ON COLUMN "Empresas"."NIF" IS 'Identificador fiscal de la empresa'
[QUERY ] COMMENT ON COLUMN "Empresas"."Nombre" IS 'Nombre de la empresa'
[QUERY ] COMMENT ON COLUMN "Empresas"."DomicilioFiscal" IS 'Domicilio fiscal de la empresa'
[QUERY ] COMMENT ON COLUMN "Empresas"."Pais" IS 'Pais '
[QUERY ] DROP TABLE "Consumidores" cascade
[QUERY ] CREATE TABLE "Consumidores"
(
    "IdConsumidor" serial CONSTRAINT "PK_Consumidores" PRIMARY KEY USING INDEX TABLESPACE
"TFG_BBDD_IDX" NOT NULL,
    "NIF" character(12) NOT NULL,
    "Direccion" character(50),
    "Ciudad" character(50),
    "Pais" character(50),
    "CodigoContratoContador" integer NOT NULL,
    "DatosBancarios" character(20),
    "CodigoEmpresaSubministradora" integer NOT NULL,
    "CodigoPais" integer NOT NULL,
    CONSTRAINT "FK_Consumidores" FOREIGN KEY ("CodigoContratoContador")
REFERENCES "Contadores" ("CodigoContratoAsociado") MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "FK_EmpresaSubministradora" FOREIGN KEY ("CodigoEmpresaSubministradora")
REFERENCES "Empresas" ("IdEmpresa") MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT "FK_Pais" FOREIGN KEY ("CodigoPais")
REFERENCES "Paises" ("IdPais") MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITH (
    OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS"
NOTICE: CREATE TABLE creará una secuencia implícita «Consumidores_IdConsumidor_seq» para la
columna serial «Consumidores.IdConsumidor»
NOTICE: CREATE TABLE / PRIMARY KEY creará el índice implícito «PK_Consumidores» para la tabla
«Consumidores»
[QUERY ] ALTER TABLE "Consumidores"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Consumidores"
```

```

IS 'Tabla de consumidores'
[QUERY ] COMMENT ON COLUMN "Consumidores"."IdConsumidor" IS 'Identificador del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."NIF" IS 'Identificador del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."Direccion" IS 'Dirección del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."Ciudad" IS 'Ciudad del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."Pais" IS 'Pais del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."CodigoContratoContador" IS 'Codigo del contrato del contador'
[QUERY ] COMMENT ON COLUMN "Consumidores"."DatosBancarios" IS 'Datos bancarios del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."CodigoEmpresaSubministradora" IS 'Codigo de la empresa
subministradora'
[QUERY ] COMMENT ON COLUMN "Consumidores"."CodigoPais" IS 'Codigo del pais'

```

```

-- Table: "Países"
DROP TABLE "Países" cascade;

CREATE TABLE "Países"
(
    "IdPais" serial CONSTRAINT "FK_ID_PAIS" PRIMARY KEY USING INDEX TABLESPACE "TFG_BBDD_IDX" NOT NULL, -- Identificador del pais
    "NombrePais" character(50) -- Nombre del pais
)
WITH (
    OIDS=FALSE
)
TABLESPACE "TFG_BBDD_DATOS";
ALTER TABLE "Países"
OWNER TO postgres;
COMMENT ON TABLE "Países"
IS 'Tabla de países';
COMMENT ON COLUMN "Países"."IdPais" IS 'Identificador del pais';

```

```

[QUERY ] ALTER TABLE "Consumidores"
OWNER TO postgres
[QUERY ] COMMENT ON TABLE "Consumidores"
IS 'Tabla de consumidores'
[QUERY ] COMMENT ON COLUMN "Consumidores"."IdConsumidor" IS 'Identificador del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."NIF" IS 'Identificador del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."Direccion" IS 'Dirección del consumidor'
[QUERY ] COMMENT ON COLUMN "Consumidores"."Ciudad" IS 'Ciudad del consumidor'

```

TFG_BBDD_tablespace.sql

Procedemos a construir los espacios de trabajos para los índices y para los datos mediante el script TFG_BBDD_tablespace.sql

```

[QUERY] CREATE TABLESPACE "TFG_BBDD_IDX"
OWNER postgres
LOCATION 'C:/
[QUERY ] COMMENT ON TABLESPACE "TFG_BBDD_IDX"
IS 'Tablespace índices'
[QUERY ] CREATE TABLESPACE "TFG_BBDD_DATOS"
OWNER postgres
LOCATION 'd:/
[QUERY ] COMMENT ON TABLESPACE "TFG_BBDD_DATOS"
IS 'Tablespace Datos'

```

```
-- Tablespace: TFG_BBDD_IDX
-- DROP TABLESPACE "TFG_BBDD_IDX"

CREATE TABLESPACE "TFG_BBDD_IDX"
OWNER postgres
LOCATION 'C:/';
COMMENT ON TABLESPACE "TFG_BBDD_IDX"
IS 'Tablespace indices';

-- Tablespace: TFG_BBDD_DATOS
-- DROP TABLESPACE "TFG_BBDD_DATOS"

CREATE TABLESPACE "TFG_BBDD_DATOS"
OWNER postgres
LOCATION 'd:/';
COMMENT ON TABLESPACE "TFG_BBDD_DATOS"
```

Output pane

[WARNING] CREATE TABLESPACE "TFG_BBDD_IDX"
OWNER postgres
LOCATION 'C:/'
ERROR: el tablespace «TFG_BBDD_IDX» ya existe

[QUERY] COMMENT ON TABLESPACE "TFG_BBDD_IDX"
IS 'Tablespace indices'

[WARNING] CREATE TABLESPACE "TFG_BBDD_DATOS"
OWNER postgres
LOCATION 'd:/'
ERROR: el tablespace «TFG_BBDD_DATOS» ya existe

2. Inicialización Base de datos

Inicializacion.sql

Ahora procedemos a cargar la base de datos e inicializarla mediante el script inicialización.sql

```
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Barcelona')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Albacete')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Valencia')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Tarragona')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Lerida')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Murcia')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Segovia')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Madrid')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Leon')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Valladolid')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Castellon')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Lugo')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Pontevedra')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Vigo')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Badajoz')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Huelva')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Malaga')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Sevilla')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Cadiz')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Granada')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Almeria')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Gerona')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Vizcaya')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Bilbao')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Zaragoza')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Teruel')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Huesca')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Pamplona')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Gijon')
[QUERY ] INSERT INTO "Ciudades"("NombreCiudad") VALUES ('Oviedo')
[QUERY ] INSERT INTO "Países"("NombrePais") VALUES ('Espanya')
[QUERY ] INSERT INTO "Países"("NombrePais") VALUES ('Francia')
```



```

[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Inglaterra')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Alemania')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Rumania')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Italia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('República checa')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Eslovenia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Polonia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Irlanda')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Marruecos')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Túnez')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Egipto')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Siria')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('China')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Japón')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Tailandia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Australia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('India')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Pakistán')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Rusia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Estados Unidos')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Argentina')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Brasil')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Venezuela')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Cuba')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Chile')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Paraguay')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Colombia')
[QUERY ] INSERT INTO "Países"("NombrePaís") VALUES ('Bolivia')
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 1',1,300,'01/01/2014','01/01/2011',1)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 2',2,350,'01/02/2014','01/01/2012',2)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 3',3,200,'01/03/2014','01/01/2014',3)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 4',4,400,'01/04/2014','01/01/2005',4)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 5',5,400,'01/05/2014','01/01/2002',5)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 6',6,350,'01/06/2014','01/01/2014',6)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 7',7,250,'01/07/2014','01/01/2005',7)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 8',8,100,'01/08/2014','01/01/2007',8)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 9',9,300,'01/10/2014','01/01/2009',9)
[QUERY ] INSERT INTO "Contadores"("Modelo", "CodigoContratoAsociado", "Potencia", "FechaUltimaInspeccion",
"FechaInstalacion", "NumeroSerie")
VALUES ('Modelo 10',10,375,'01/12/2014','01/01/2010',10)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('12/01/2014','00:40:05','Irlanda',200.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('22/02/2014','12:40:60','Paraguay',235.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('30/03/2014','22:30:60','Espanya',335.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('15/04/2014','22:15:60','Italia',235.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('25/05/2014','11:45:60','Cuba',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('22/06/2014','05:45:60','Chile',235.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "País", "PrecioKw") VALUES
('11/06/2014','08:30:60','Argentina',335.50)

```

```

[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('05/07/2014','10:30:60','Brasil',235.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('10/07/2014','20:10:60','Brasil',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('12/07/2014','20:20:60','Brasil',235.45)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/07/2014','04:25:60','Venezuela',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/08/2014','05:40:60','Estados Unidos',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/08/2014','15:50:60','Irlanda',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/09/2014','23:55:60','Paraguay',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/10/2014','06:30:60','Colombia',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/11/2014','05:15:60','Rusia',335.50)
[QUERY ] INSERT INTO "PrecioEnergia"("Fecha", "Hora", "Pais", "PrecioKw") VALUES
('15/12/2014','02:58:60','Canada',335.50)
[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('11111111A','Avenida Madrid, 222 4º1º','Sevilla','Espanya',1,'ES209332171160093332',1,1)

[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('22222222B','Avenida Madrid, 222 4º1º','Barcelona','Espanya',1,'ES209332171160093333',9,1)
[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('33333333C','Avenida Madrid, 222 4º1º','Madrid','Espanya',1,'ES209332171160093334',2,1)

[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('44444444D','Avenida Madrid, 222 4º1º','Gijon','Espanya',1,'ES209332171160093335',3,1)

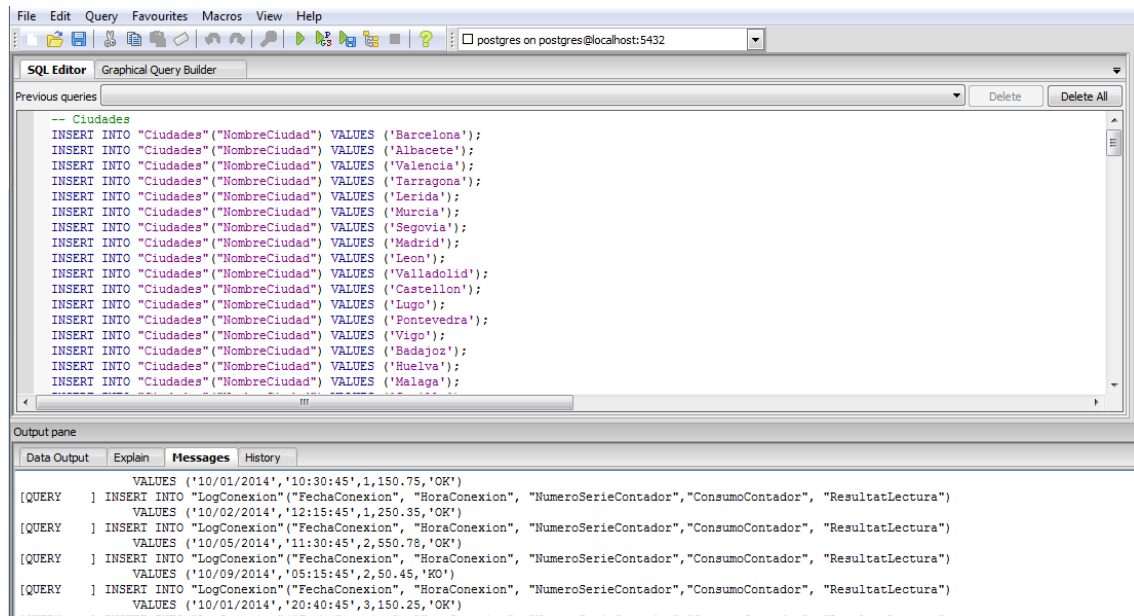
[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('55555555E','Avenida Madrid, 222 4º1º','Malaga','Espanya',1,'ES209332171160093336',3,1)

[QUERY ] INSERT INTO "Consumidores"("NIF", "Direccion", "Ciudad", "Pais",
"CodigoContratoContador", "DatosBancarios", "CodigoEmpresaSubministradora", "CodigoPais")
VALUES ('66666666F','Avenida Madrid, 222 4º1º','Valencia','Espanya',1,'ES209332171160093337',5,1)

[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('11111111A','Atos,
S.L.','Avda Londres, 321 1º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('22222222A','Sonic,
S.L.','Avda Paris, 21 2º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('33333333A','Acer,
S.L.','Avda Icaria, 32 4º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('44444444B','Yamaha,
S.L.','C/ Ignacio, 421 7º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('55555555C','Honda,
S.L.U.','Avda Meridiana, 4 4º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('66666666D','Tecnocom,
S.A.','Avda Murcia, 2 7º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('77777777E','Indra,
S.A.','Poligono Bernardo, 3 14º 3º','Espanya')
[QUERY ] INSERT INTO "Empresas"("NIF", "Nombre", "DomicilioFiscal", "Pais") VALUES ('88888888F','HP,
S.A.','Avda Madrid, 32 42º 3º','Espanya')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador", "ConsumoContador", "ResultadoLectura")
VALUES ('10/01/2014','10:30:45',1,150.75,'OK')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador", "ConsumoContador", "ResultadoLectura")
VALUES ('10/02/2014','12:15:45',1,250.35,'OK')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador", "ConsumoContador", "ResultadoLectura")
VALUES ('10/05/2014','11:30:45',2,550.78,'OK')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador", "ConsumoContador", "ResultadoLectura")
VALUES ('10/09/2014','05:15:45',2,50.45,'KO')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador", "ConsumoContador", "ResultadoLectura")






















```

```
VALUES ('10/01/2014','20:40:45',3,150.25,'OK')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador","ConsumoContador", "ResultatLectura")
VALUES ('10/12/2014','16:10:45',4,550.75,'OK')
[QUERY ] INSERT INTO "LogConexion"("FechaConexion", "HoraConexion",
"NumeroSerieContador","ConsumoContador", "ResultatLectura")
VALUES ('10/07/2014','14:20:45',5,180.90,'KO')
```



3. Creación Procedimientos ABM

Ahora procedemos a ejecutar los procedimientos ABM (Altas, bajas y modificaciones) mediante los scripts siguientes:

Nombre	Fecha modificación	Tipo	Tamaño
 p_alta_ciudad	11/05/2014 20:01	Archivo SQL	1 KB
 p_alta_ciudad	11/05/2014 20:03	Archivo SQL	2 KB
 p_alta_ciudad	11/05/2014 20:02	Archivo SQL	1 KB
 p_alta_ciudad	13/05/2014 0:21	Archivo SQL	1 KB
 p_alta_logConexion	11/05/2014 20:03	Archivo SQL	1 KB
 p_alta_pais	11/05/2014 20:03	Archivo SQL	1 KB
 p_alta_precioEnergia	11/05/2014 20:02	Archivo SQL	1 KB
 p_baja_ciudad	12/05/2014 23:36	Archivo SQL	1 KB
 p_baja_consumidor	12/05/2014 23:45	Archivo SQL	1 KB
 p_baja_contador	12/05/2014 23:56	Archivo SQL	1 KB
 p_baja_empresa	13/05/2014 0:24	Archivo SQL	1 KB
 p_baja_logConexion	13/05/2014 0:04	Archivo SQL	1 KB
 p_baja_pais	13/05/2014 0:08	Archivo SQL	1 KB
 p_baja_precioEnergia	13/05/2014 0:29	Archivo SQL	1 KB
 p_modificacion_ciudad	13/05/2014 1:27	Archivo SQL	1 KB
 p_modificacion_consumidor	13/05/2014 0:43	Archivo SQL	2 KB
 p_modificacion_contador	13/05/2014 0:52	Archivo SQL	2 KB
 p_modificacion_empresas	13/05/2014 0:59	Archivo SQL	1 KB
 p_modificacion_logConexion	13/05/2014 1:05	Archivo SQL	2 KB
 p_modificacion_pais	13/05/2014 1:12	Archivo SQL	1 KB
 p_modificacion_precioEnergia	13/05/2014 1:20	Archivo SQL	1 KB

P_alta_ciudad.sql

```
[QUERY ] create or replace function p_alta_ciudad (p_nombreCiudad varchar) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
    select count(*) into contador from "Ciudades"
        where "NombreCiudad" = p_nombreCiudad;
    if contador > 0 then
        Rsp:='ERROR: CIUDAD YA EXISTE !!!';
    ELSE
        Insert into "Ciudades" ("NombreCiudad") values(p_nombreCiudad);
        Rsp:='OK: el procedimiento ha sido un éxito';
    END IF;
    return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_alta_ciudad('Albania')
```

P_alta_consumidor.sql

```
[QUERY ] create or replace function p_alta_consumidor (pNIF character(12), pDireccion character(50), pCiudad
character(50),
                pPais character(50),pCodigoContratoContador integer,DatosBancarios character(20),
                CodigoEmpresaSubministradora integer,CodigoPais integer) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
    select count(*) into contador from "Consumidores"
        where "NIF" = pNIF;

    if contador > 0 then
        Rsp:='ERROR: CONSUMIDOR YA EXISTE !!!';
    ELSE
        insert into "Consumidores" ("NIF", "Direccion", "Ciudad", "Pais", "CodigoContratoContador", "DatosBancarios",
                "CodigoEmpresaSubministradora", "CodigoPais") values (pNIF, pDireccion, pCiudad, pPais, pCodigoContratoContador, DatosBancarios, CodigoEmpresaSubministradora, CodigoPais);
    END IF;
    return Rsp;
end;
```

```

        "CodigoEmpresaSubministradora","CodigoPais")
    values (pNIF, pDireccion, pCiudad,pPais,pCodigoContratoContador,DatosBancarios,
        CodigoEmpresaSubministradora,CodigoPais );
    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_alta_consumidor('11111111F','Londres, 334 2º2º','Sevilla','Espanya',1,'ES422221443234444',1,1)

```

P_alta_contador.sql

```

[QUERY ] create or replace function p_alta_contador(pModelo character(50), pCodigoContratoAsociado integer,
pPotencia integer, pFechaUltimaInspeccion date,
        pFechaInstalacion date,pNumeroSerie integer) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
    select count(*) into contador from "Contadores"
        where "NumeroSerie" = pNumeroSerie
        and "CodigoContratoAsociado" = pCodigoContratoAsociado;
    if contador > 0 then
        Rsp:='ERROR: CONTADOR YA EXISTE !!!';
    ELSE
        insert into "Contadores"
        ("Modelo","CodigoContratoAsociado","Potencia","FechaUltimaInspeccion","FechaInstalacion","NumeroSerie")
        values
        (pModelo,pCodigoContratoAsociado,pPotencia,pFechaUltimaInspeccion,pFechaInstalacion,pNumeroSerie);
        Rsp:='OK: el procedimiento ha sido un éxito';
    END IF;
    return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_alta_contador('Modelo 1',11,300,'02/01/2014','02/01/2014',11)

```

P_alta_empresas.sql

```

[QUERY ] create or replace function p_alta_empresa (pNIF character(12), pNombre character(50),pDomicilioFiscal
character(50),pPais character(50)) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
    select count(*) into contador from "Empresas"
        where "NIF" = pNIF;
    if contador > 0 then
        Rsp:='ERROR: EMPRESA YA EXISTE !!!';
    else
        Insert into "Empresas" ("NIF", "Nombre", "DomicilioFiscal","Pais")
        values(pNIF, pNombre, pDomicilioFiscal, pPais);
        Rsp:='OK: el procedimiento ha sido un éxito';
    end if;
    return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_alta_empresa('11111111A','Sonic S.L.','Avda Madrid, 233 2º1º','Espanya')

```

P_alta_logConexion.sql

```

[QUERY ] create or replace function p_alta_logConexion (pFechaConexion date, pHoraConexion time(6) with time
zone, pNumeroSerieContador integer,
        pConsumoContador real,pResultatLectura character(2)) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
    select count(*) into contador from "LogConexion"
        where "FechaConexion" = pFechaConexion

```

```

and "HoraConexion" = pHoraConexion
and "NumeroSerieContador" = pNumeroSerieContador;

if contador > 0 then
  Rsp:= 'ERROR: LOG DE CONEXION YA EXISTE !!!';
ELSE
  insert into "LogConexion"
("FechaConexion", "HoraConexion", "NumeroSerieContador", "ConsumoContador", "ResultatLectura")
  values (pFechaConexion, pHoraConexion,
pNumeroSerieContador, pConsumoContador, pResultatLectura);
  Rsp:= 'OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_alta_logConexion('02/01/2014','02:01:05',1,550,'OK')

```

P_alta_pais.sql

```

[QUERY ] create or replace function p_alta_pais (p_nombrePais varchar) RETURNS text AS $$
declare contador numeric;
  Rsp TEXT ;
begin
  select count(*) into contador from "Paises"
  where "NombrePais" = p_nombrePais;
  if contador > 0 then
    Rsp:= 'ERROR: PAIS YA EXISTE !!!';
  else
    Insert into "Paises" ("NombrePais") values(p_nombrePais);
    Rsp:= 'OK: el procedimiento ha sido un éxito';
  end if;
  return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_alta_pais('Canada4')

```

P_alta_precioEnergia.sql

```

[QUERY ] create or replace function p_alta_precioEnergia (p_fecha date, p_hora time(6) with time zone, p_pais
character(30), p_precio real) RETURNS text AS $$
declare contador numeric;
  Rsp TEXT ;
begin
  select count(*) into contador from "PrecioEnergia"
  where "Fecha" = p_fecha
  and "Hora" = p_hora
  and "Pais" = p_pais;
  if contador > 0 then
    Rsp:= 'ERROR: PRECIO YA EXISTE !!!';
  ELSE
    Insert into "PrecioEnergia" ("Fecha", "Hora", "Pais", "PrecioKw") values(p_fecha, p_hora, p_pais, p_precio);
    Rsp:= 'OK: el procedimiento ha sido un éxito';
  END IF;
  return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_alta_precioEnergia('02/01/2014','10:30:00','Espanya',350.34)

```

P_baja_ciudad.sql

```

[QUERY ] create or replace function p_baja_ciudad (p_nombreCiudad varchar) RETURNS text AS $$
declare contador numeric;
  Rsp TEXT ;
begin
  select count(*) into contador from "Ciudades"
  where "NombreCiudad" = p_nombreCiudad;

```

```

if contador = 0 then
    Rsp:='ERROR: CIUDAD NO EXISTE !!!';
ELSE
    DELETE FROM "Ciudades" WHERE "NombreCiudad" = p_nombreCiudad;
    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_baja_ciudad('Albania')

```

P_baja_consumidor.sql

```

[QUERY ] create or replace function p_baja_consumidor (pNIF character(12)) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "Consumidores"
    where "NIF" = pNIF;

if contador = 0 then
    Rsp:='ERROR: CONSUMIDOR NO EXISTE !!!';
ELSE
    DELETE FROM "Consumidores" WHERE "NIF" = pNIF;
    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_baja_consumidor('11111111B')

```

P_baja_contador.sql

```

[QUERY ] create or replace function p_baja_contador(pCodigoContratoAsociado integer, pNumeroSerie integer)
RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "Contadores"
    where "NumeroSerie" = pNumeroSerie
    and "CodigoContratoAsociado" = pCodigoContratoAsociado;
if contador = 0 then
    Rsp:='ERROR: CONTADOR NO EXISTE !!!';
ELSE
    DELETE FROM "Contadores" WHERE "CodigoContratoAsociado" = pCodigoContratoAsociado
        AND "NumeroSerie" = pNumeroSerie;
    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql

```

P_baja_empresa.sql

```

[QUERY ] create or replace function p_baja_empresa (pNIF character(12)) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "Empresas"
    where "NIF" = pNIF;
if contador = 0 then
    Rsp:='ERROR: EMPRESA NO EXISTE !!!';
else
    DELETE FROM "Empresas" WHERE "NIF" = pNIF;
    Rsp:='OK: el procedimiento ha sido un éxito';
end if;

```



```

        return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_baja_empresa('11111111A')

```

P_baja_logConexion.sql

```

[QUERY ] create or replace function p_baja_logConexion (pFechaConexion date, pHoraConexion time(6) with time
zone, pNumeroSerieContador integer) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
        select count(*) into contador from "LogConexion"
        where "FechaConexion" = pFechaConexion
        and "HoraConexion" = pHoraConexion
        and "NumeroSerieContador" = pNumeroSerieContador;

        if contador = 0 then
            Rsp:=ERROR: LOG DE CONEXION NO EXISTE !!!;
        ELSE
            DELETE FROM "LogConexion" WHERE "FechaConexion" = pFechaConexion AND
            "HoraConexion" = pHoraConexion AND
            "NumeroSerieContador" = pNumeroSerieContador;

            Rsp:=OK: el procedimiento ha sido un éxito;
        END IF;
        return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_baja_logConexion('10/01/2014','10:30:45',1)

```

P_baja_pais.sql

```

[QUERY ] create or replace function p_baja_pais (p_nombrePais varchar) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
        select count(*) into contador from "Paises"
        where "NombrePais" = p_nombrePais;
        if contador = 0 then
            Rsp:=ERROR: PAIS NO EXISTE !!!;
        else
            DELETE FROM "Paises" WHERE "NombrePais" = p_nombrePais;
            Rsp:=OK: el procedimiento ha sido un éxito;
        end if;
        return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_baja_pais('Canada')

```

P_baja_PrecioEnergia.sql

```

[QUERY ] create or replace function p_baja_precioEnergia (p_fecha date, p_hora time(6) with time zone, p_pais
character(30)) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
    begin
        select count(*) into contador from "PrecioEnergia"
        where "Fecha" = p_fecha
        and "Hora" = p_hora
        and "Pais"=p_pais;
        if contador = 0 then
            Rsp:=ERROR: PRECIO NO EXISTE !!!;
        ELSE
            DELETE FROM "PrecioEnergia" WHERE "Fecha" = p_fecha
            AND "Hora" = p_hora

```



```

AND "Pais" = p_pais;

    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_baja_precioEnergia('02/01/2014','10:30:00','Espanya')

```

P_modificacion_ciudad.sql

```

[QUERY ] create or replace function p_modificacion_ciudad (pIdCiudad integer, p_nombreCiudad varchar) RETURNS
text AS $$
    declare contador numeric;
    Rsp TEXT ;
begin
    select count(*) into contador from "Ciudades"
    where "IdCiudad" = pIdCiudad;
    if contador = 0 then
        Rsp:='ERROR: CIUDAD NO EXISTE !!!';
    ELSE
        UPDATE "Ciudades"
        SET "NombreCiudad" = p_nombreCiudad
        WHERE "IdCiudad" = pIdCiudad;
        Rsp:='OK: el procedimiento ha sido un éxito';
    END IF;
    return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_ciudad(1,'Madrid')

```

P_modificacion_consumidor.sql

```

[QUERY ] create or replace function p_modificacion_consumidor (pNIF character(12), pDireccion character(50),
pCiudad character(50),
pPais character(50),pCodigoContratoContador integer,DatosBancarios character(20),
CodigoEmpresaSubministradora integer,CodigoPais integer) RETURNS text AS $$
    declare contador numeric;
    Rsp TEXT ;
begin
    select count(*) into contador from "Consumidores"
    where "NIF" = pNIF;

    if contador = 0 then
        Rsp:='ERROR: CONSUMIDOR NO EXISTE !!!';
    ELSE
        UPDATE "Consumidores"
        SET "Direccion" = pDireccion,
        "Ciudad" = pCiudad,
        "Pais" = pPais,
        "CodigoContratoContador" = pCodigoContratoContador,
        "DatosBancarios" = DatosBancarios,
        "CodigoEmpresaSubministradora" = CodigoEmpresaSubministradora,
        "CodigoPais" = CodigoPais
        WHERE "NIF" = pNIF;
        Rsp:='OK: el procedimiento ha sido un éxito';
    END IF;
    return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_consumidor('11111111F','Londres, 334
2º2º','Sevilla','Espanya',1,'ES42221443234444',1,1)

```

P_modificacion_contador.sql

```

[QUERY ] create or replace function p_modificacion_contador(pModelo character(50), pCodigoContratoAsociado
integer, pPotencia integer, pFechaUltimaInspeccion date,

```

```

        pFechaInstalacion date,pNumeroSerie integer) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "Contadores"
    where "NumeroSerie" = pNumeroSerie
    and "CodigoContratoAsociado" = pCodigoContratoAsociado;
IF contador = 0 then
    Rsp:='ERROR: CONTADOR NO EXISTE !!!';
ELSE
    UPDATE "Contadores"
    SET "Modelo" = pModelo ,
        "CodigoContratoAsociado" = pCodigoContratoAsociado,
        "Potencia" =pPotencia,
        "FechaUltimaInspeccion"=pFechaUltimaInspeccion,
        "FechaInstalacion"=pFechaInstalacion,
        "NumeroSerie" = pNumeroSerie
    WHERE "NumeroSerie" = pNumeroSerie
    AND "CodigoContratoAsociado" = pCodigoContratoAsociado;

    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_contador('Modelo 10000',10,300,'02/01/2014','02/01/2014',10)

```

P_modificacion_empresas.sql

```

[QUERY ] create or replace function p_modificacion_empresa (pNIF character(12), pNombre
character(50),pDomicilioFiscal character(50),pPais character(50)) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "Empresas"
    where "NIF" = pNIF;
if contador = 0 then
    Rsp:='ERROR: EMPRESA NO EXISTE !!!';
else
    UPDATE "Empresas"
    SET "NIF" = pNIF,
        "Nombre" = pNombre,
        "DomicilioFiscal" = pDomicilioFiscal,
        "Pais" = pPais
    WHERE "NIF" = pNIF;
    Rsp:='OK: el procedimiento ha sido un éxito';
end if;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_empresa('11111111A','Sonic S.L.A.','Avda Madrid, 233 2º1º','Espanya')

```

P_modificacion_logConexion.sql

```

[QUERY ] create or replace function p_modificacion_logConexion (pFechaConexion date, pHoraConexion time(6)
with time zone, pNumeroSerieContador integer,
        pConsumoContador real,pResultatLectura character(2)) RETURNS text AS $$
declare contador numeric;
    Rsp TEXT ;
begin
select count(*) into contador from "LogConexion"
    where "FechaConexion" = pFechaConexion
    and "HoraConexion" = pHoraConexion
    and "NumeroSerieContador" = pNumeroSerieContador;

if contador = 0 then
    Rsp:='ERROR: LOG DE CONEXION NO EXISTE !!!';
ELSE

```

```

UPDATE "LogConexion"
SET "FechaConexion"=pFechaConexion,
    "HoraConexion"=pHoraConexion,
    "NumeroSerieContador"=pNumeroSerieContador,
    "ConsumoContador"=pConsumoContador,
    "ResultatLectura"=pResultatLectura
WHERE "FechaConexion" = pFechaConexion
AND "HoraConexion" = pHoraConexion
AND "NumeroSerieContador" = pNumeroSerieContador;

    Rsp:='OK: el procedimiento ha sido un éxito';
END IF;
return Rsp;
end;
$$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_logConexion('10/02/2014','12:15:45',1,1000,'KO')

```

P_modificacion_pais.sql

```

[QUERY ] create or replace function p_modificacion_pais (pIdPais integer,p_nombrePais varchar) RETURNS text AS
$$
    declare contador numeric;
        Rsp TEXT ;
    begin
        select count(*) into contador from "Países"
        where "IdPais" = pIdPais;
        if contador = 0 then
            Rsp:='ERROR: PAIS NO EXISTE !!!';
        else
            UPDATE "Países"
            SET "NombrePais"= p_nombrePais
            WHERE "IdPais" = pIdPais;
            Rsp:='OK: el procedimiento ha sido un éxito';
        end if;
        return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_pais(2,'Canada')

```

P_modificacion_precioEnergia.sql








```

[QUERY ] create or replace function p_modificacion_precioEnergia (p_fecha date, p_hora time(6) with time zone,
p_pais character(30), p_precio real) RETURNS text AS $$
    declare contador numeric;
        Rsp TEXT ;
    begin
        select count(*) into contador from "PrecioEnergia"
        where "Fecha" = p_fecha
        and "Hora" = p_hora
        and "Pais"=p_pais;
        if contador = 0 then
            Rsp:='ERROR: PRECIO NO EXISTE !!!';
        ELSE
            UPDATE "PrecioEnergia"
            SET "Fecha"=p_fecha,
                "Hora" =p_hora,
                "Pais" = p_pais,
                "PrecioKw" = p_precio
            WHERE "Fecha" = p_fecha
            AND "Hora" = p_hora
            AND "Pais"=p_pais;
            Rsp:='OK: el procedimiento ha sido un éxito';
        END IF;
        return Rsp;
    end;
    $$ LANGUAGE plpgsql
[QUERY ] select p_modificacion_precioEnergia('12/01/2014','00:40:05','Irlanda',12350.34)

```

4. Creación procedimientos de Consulta/Estadística

A continuación crearemos los procedimientos de consulta y estadística mediante los scripts que indicamos en la siguiente pantalla:

Nombre	Fecha modificación	Tipo	Tamaño
 p_Cons_Elect_Ciudad	15/05/2014 1:36	Archivo SQL	3 KB
 p_Consumo_Medio	15/05/2014 1:34	Archivo SQL	2 KB
 p_ConsumoElectrico_Mensual	13/05/2014 2:45	Archivo SQL	2 KB
 p_List_Cont_Antig	15/05/2014 1:37	Archivo SQL	2 KB
 p_List_Cont_Cons_Mensual	15/05/2014 1:38	Archivo SQL	3 KB
 p_Porcentaje_Lecturas	15/05/2014 1:37	Archivo SQL	3 KB
 p_Top10_Contadores	15/05/2014 0:33	Archivo SQL	2 KB

P_Cons_Elect_Ciudad.sql

```
[QUERY ] DROP TABLE TB_CONS_ELECT_CIUADAD
[QUERY ] CREATE TABLE TB_CONS_ELECT_CIUADAD (
    ciudad character(50),
    consumo real)
[QUERY ] CREATE OR REPLACE FUNCTION p_Cons_Elect_Ciudad(pMes character(2), pPais character(50))
    RETURNS TABLE (ciudad character(50), consumo real) AS
    $$
    DELETE FROM TB_CONS_ELECT_CIUADAD;
    INSERT INTO TB_CONS_ELECT_CIUADAD
        SELECT cons."Ciudad", sum(log."ConsumoContador")
            FROM "Consumidores" cons,
            "Contadores" cont,
            "LogConexion" log
            WHERE substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) = $1
            AND log."NumeroSerieContador" = cont."NumeroSerie"
            AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
            AND cons."Pais" = $2
            GROUP BY cons."Ciudad"
            ORDER BY 1;

    SELECT cons."Ciudad", sum(log."ConsumoContador")
        FROM "Consumidores" cons,
        "Contadores" cont,
        "LogConexion" log
        WHERE substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) = $1
        AND log."NumeroSerieContador" = cont."NumeroSerie"
        AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
        AND cons."Pais" = $2
        GROUP BY cons."Ciudad"
        ORDER BY 1;

    $$ LANGUAGE SQL
[QUERY ] select * from p_Cons_Elect_Ciudad('02','Espanya')
```

P_Consumo_Medio.sql

```
[QUERY ] DROP TABLE TB_CONSUMO_MEDIO
[QUERY ] CREATE TABLE TB_CONSUMO_MEDIO
(
    NIF character(12),
    Direccion character(50),
    mes character(2),
```

```

ciudad character(50),
ConsumoMedio real
)
[QUERY ] CREATE OR REPLACE FUNCTION p_Consumo_Medio(pCiudad character(50), pMes character(2))
RETURNS TABLE (NIF character(12),
Direccion character(50),
mes character(2),
ciudad character(50),
ConsumoMedio real) AS
$$
DELETE FROM TB_CONSUMO_MEDIO;
INSERT INTO TB_CONSUMO_MEDIO
SELECT cons."NIF", cons."Direccion", cons."Ciudad",
substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) as Mes, log."ConsumoContador"
FROM "LogConexion" log,
"Contadores" cont,
"Consumidores" cons
WHERE substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) = $2
AND log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."Ciudad" = $1;

SELECT cons."NIF", cons."Direccion", cons."Ciudad",
substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) as Mes, log."ConsumoContador"
FROM "LogConexion" log,
"Contadores" cont,
"Consumidores" cons
WHERE substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) = $2
AND log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."Ciudad" = $1;

$$ LANGUAGE SQL
[QUERY ] select * from p_Consumo_Medio('Barcelona','2')

```

P_ConsumoElectrico_Mensual.sql

```

[QUERY ] DROP TABLE TB_CONSUMO_ELECTRICO_MENSUAL
[QUERY ] CREATE TABLE TB_CONSUMO_ELECTRICO_MENSUAL (
"Modelo" character(12),
"NumeroSerieContador" integer,
"FechaConexion" text,
"ConsumoContador" real)
[QUERY ] CREATE OR REPLACE FUNCTION p_Consumo_Electrico_Mensual() RETURNS table (Modelo
character(12), NumeroSerieContador integer, FechaConexion text, consumo real) AS
$$
INSERT INTO TB_CONSUMO_ELECTRICO_MENSUAL
SELECT cont."Modelo", cont."NumeroSerie", substr(to_char("FechaConexion",'yyyy-mm-dd'),6,2)
mes, sum(log."ConsumoContador")
FROM "Contadores" cont,
"LogConexion" log
WHERE cont."NumeroSerie" = log."NumeroSerieContador"
GROUP BY cont."Modelo",
cont."NumeroSerie", substr(to_char("FechaConexion",'yyyy-mm-dd'),6,2)
order by 1,2,3;

SELECT cont."Modelo", cont."NumeroSerie", substr(to_char("FechaConexion",'yyyy-mm-dd'),6,2)
mes, sum(log."ConsumoContador")
FROM "Contadores" cont,
"LogConexion" log
WHERE cont."NumeroSerie" = log."NumeroSerieContador"
GROUP BY cont."Modelo",
cont."NumeroSerie", substr(to_char("FechaConexion",'yyyy-mm-dd'),6,2)
order by 1,2,3;

$$ LANGUAGE SQL
[QUERY ] select * from p_Consumo_Electrico_Mensual()

```

P_List_Cont_Antig.sql

```

[QUERY] DROP TABLE TB_LIST_CONT_ANTIG
[QUERY ] CREATE TABLE TB_LIST_CONT_ANTIG (
    Modelo character(12),
    NumeroSerie integer,
    FechaInstalacion date
)
[QUERY ] CREATE OR REPLACE FUNCTION p_List_Cont_Antig(Antiguedad integer)
RETURNS TABLE (Modelo character(12), NumeroSerie integer, FechaInstalacion date) AS
$$
DELETE FROM TB_LIST_CONT_ANTIG;
INSERT INTO TB_LIST_CONT_ANTIG
SELECT cont."Modelo", cont."NumeroSerie", cont."FechaInstalacion"
FROM "Contadores" cont
WHERE extract(year from now()) - extract(year from cont."FechaInstalacion") = $1;

SELECT cont."Modelo", cont."NumeroSerie", cont."FechaInstalacion"
FROM "Contadores" cont
WHERE extract(year from now()) - extract(year from cont."FechaInstalacion") = $1;

$$ LANGUAGE SQL
[QUERY ] select * from p_List_Cont_Antig(2)

```

P_List_Cont_Cons_Mensual.sql

```

[QUERY ] CREATE OR REPLACE FUNCTION p_List_Cont_Cons_Mensual() RETURNS table (Modelo
character(12), NumeroSerieContador integer, FechaConexion text, consumo real) AS
$$
SELECT cont."Modelo", cont."NumeroSerie", substr(to_char("FechaConexion", 'yyyy-mm-dd'), 6, 2)
mes, sum(log."ConsumoContador")
FROM "Contadores" cont,
"LogConexion" log
WHERE cont."NumeroSerie" = log."NumeroSerieContador"
GROUP BY cont."Modelo",
cont."NumeroSerie", substr(to_char("FechaConexion", 'yyyy-mm-dd'), 6, 2)
order by 1, 2, 3;

SELECT cont."Modelo", cont."NumeroSerie", substr(to_char("FechaConexion", 'yyyy-mm-dd'), 6, 2)
mes, sum(log."ConsumoContador")
FROM "Contadores" cont,
"LogConexion" log
WHERE cont."NumeroSerie" = log."NumeroSerieContador"
GROUP BY cont."Modelo",
cont."NumeroSerie", substr(to_char("FechaConexion", 'yyyy-mm-dd'), 6, 2)
order by 1, 2, 3;

$$ LANGUAGE SQL
[QUERY ] select * from p_Consumo_Electrico_Mensual()

```

P_Porcentaje_Lecturas.sql

```

[QUERY ] DROP TABLE TB_PORCENTAJE_LLECTURAS
[QUERY ] CREATE TABLE TB_PORCENTAJE_LLECTURAS (
    porcentaje numeric)
[QUERY ] CREATE OR REPLACE FUNCTION p_Porcentaje_Lecturas(pEmpresa character(50), pMes character(2))
RETURNS TABLE (porcentaje numeric) AS
$$
DELETE FROM TB_PORCENTAJE_LLECTURAS;
INSERT INTO TB_PORCENTAJE_LLECTURAS
SELECT (count(1)*0.1)*100 AS PORCENTAJE
FROM "Empresas" emp,
"Consumidores" cons,
"Contadores" cont,
"LogConexion" log
WHERE substr(to_char(log."FechaConexion", 'yyyy-mm-dd'), 6, 2) = $2
AND log."ResultadoLectura" = 'OK'
AND log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."CodigoEmpresaSubministradora" = emp."IdEmpresa"
AND emp."Nombre" = $1;

SELECT (count(1)*0.1)*100 AS PORCENTAJE

```

```

FROM "Empresas" emp,
     "Consumidores" cons,
     "Contadores" cont,
     "LogConexion" log
WHERE substr(to_char(log."FechaConexion",'yyyy-mm-dd'),6,2) = $2
AND log."ResultatLectura" = 'OK'
AND log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."CodigoEmpresaSubministradora" = emp."IdEmpresa"
AND emp."Nombre" = $1;

$$ LANGUAGE SQL
[QUERY ] select * from p_Porcentaje_Lecturas('Sonic, S.L.', '10')

```

P_Top10_Contadores.sql

```

[QUERY ] DROP TABLE TB_TOP_CONTADORES
[QUERY ] CREATE TABLE TB_TOP_CONTADORES (
    Modelo character(50),
    NumeroSerie integer,
    ConsumoContador real)
[QUERY ] CREATE OR REPLACE FUNCTION p_Top10_Contadores(pCiudad character(50))
RETURNS TABLE (Modelo character(50), NumeroSerie integer, ConsumoContador real) AS
$$
DELETE FROM TB_TOP_CONTADORES;
INSERT INTO TB_TOP_CONTADORES
SELECT cont."Modelo", cont."NumeroSerie", log."ConsumoContador"
FROM "Consumidores" cons,
     "Contadores" cont,
     "LogConexion" log
WHERE log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."Ciudad" = 'Barcelona'
ORDER BY log."ConsumoContador"
DESC LIMIT 10;

SELECT cont."Modelo", cont."NumeroSerie", log."ConsumoContador"
FROM "Consumidores" cons,
     "Contadores" cont,
     "LogConexion" log
WHERE log."NumeroSerieContador" = cont."NumeroSerie"
AND cont."CodigoContratoAsociado" = cons."CodigoContratoContador"
AND cons."Ciudad" = 'Barcelona'
ORDER BY log."ConsumoContador"
DESC LIMIT 10;
$$ LANGUAGE SQL
[QUERY ] select * from p_Top10_Contadores('Barcelona')

```

5. LOGS EJECUCIONES PROCEDIMIENTOS

5.1. LOG EJECUCION PROCEDIMIENTOS ABM

Procederemos a ejecutar los siguientes procedimientos ABM

```
select p_alta_ciudad('Albania');
```

Output pane	
Data Output Explain Messages History	
	p_alta_ciudad text
1	OK: el procedimiento ha sido un éxito

```
select p_alta_consumidor('1111111F','Londres, 334  
2º2º','Sevilla','Espanya',1,'ES422221443234444',1,1);
```

Output pane	
Data Output Explain Messages History	
	p_alta_consumidor text
1	ERROR: CONSUMIDOR YA EXISTE !!!

```
select p_alta_contador('Modelo 1',11,300,'02/01/2014','02/01/2014',11);
```

Output pane	
Data Output Explain Messages History	
	p_alta_contador text
1	ERROR: CONTADOR YA EXISTE !!!

```
select p_alta_empresa('1111111A','Sonic S.L.','Avda Madrid, 233  
2º1º','Espanya');
```


Output pane	
Data Output Explain Messages History	
	p_alta_empresa text
1	OK: el procedimiento ha sido un éxito

`select p_alta_logConexion('02/01/2014','02:01:05',1,550,'OK');`

Output pane	
Data Output Explain Messages History	
	p_alta_logconexion text
1	ERROR: LOG DE CONEXION YA EXISTE !!!

`select p_alta_pais('Canada4');`

Output pane	
Data Output Explain Messages History	
	p_alta_pais text
1	ERROR: PAIS YA EXISTE !!!

`select p_alta_precioEnergia('02/01/2014','10:30:00','Espanya',350.34);`

Output pane	
Data Output Explain Messages History	
	p_alta_precioenergia text
1	OK: el procedimiento ha sido un éxito

`select p_baja_ciudad('Albania');`

Output pane	
Data Output Explain Messages History	
	p_alta_precioenergia text
1	OK: el procedimiento ha sido un éxito

```
select p_baja_consumidor('1111111B');
```

Output pane	
Data Output Explain Messages History	
p_baja_consumidor text	
1	ERROR: CONSUMIDOR NO EXISTE !!!

```
select p_baja_contador(1,1);
```

Output pane	
Data Output Explain Messages History	
p_baja_contador text	
1	ERROR: CONTADOR NO EXISTE !!!

```
select p_baja_empresa('1111111A');
```

Output pane	
Data Output Explain Messages History	
p_baja_empresa text	
1	OK: el procedimiento ha sido un éxito

```
select p_baja_logConexion('10/01/2014','10:30:45',1);
```

Output pane	
Data Output Explain Messages History	
p_baja_logconexion text	
1	ERROR: LOG DE CONEXION NO EXISTE !!!

```
select p_baja_pais('Canada');
```

Output pane	
Data Output	Explain Messages History
	p_baja_pais text
1	ERROR: PAIS NO EXISTE !!!

```
select p_baja_precioEnergia('02/01/2014','10:30:00','Espanya');
```

Output pane	
Data Output	Explain Messages History
	p_baja_precioenergia text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_ciudad(1,'Madrid');
```

Output pane	
Data Output	Explain Messages History
	p_modificacion_ciudad text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_consumidor('1111111F','Londres, 334  
2º2º','Sevilla','Espanya',1,'ES422221443234444',1,1);
```

Output pane	
Data Output	Explain Messages History
	p_modificacion_consumidor text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_contador('Modelo  
10000',10,300,'02/01/2014','02/01/2014',10);
```

Output pane	
Data Output Explain Messages History	
	p_modificacion_contador text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_empresa('1111111A','Sonic S.L.A.','Avda Madrid, 233  
2º1º','Espanya');
```

Output pane	
Data Output Explain Messages History	
	p_modificacion_empresa text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_logConexion('10/02/2014','12:15:45',1,1000,'KO');
```

Output pane	
Data Output Explain Messages History	
	p_modificacion_logconexion text
1	OK: el procedimiento ha sido un éxito

```
select p_modificacion_pais(2,'Canada');
```

Output pane	
Data Output Explain Messages History	
	p_modificacion_pais text
1	ERROR: PAIS NO EXISTE !!!

```
select  
p_modificacion_precioEnergia('12/01/2014','00:40:05','Irlanda',12350.34);
```

Output pane	
Data Output	Explain Messages History
	p_modificacion_precioenergia text
1	OK: el procedimiento ha sido un éxito

Como se puede apreciar en algunos logs se procede a dar el alta, la baja o la modificación con éxito y en algunos otros casos se retorna un log de no haberse podido ejecutar el alta/baja o modificación.

5.2. LOG EJECUCION PROCEDIMIENTOS CONSULTA/ESTADÍSTICA

`select * from p_Cons_Elect_Ciudad('02','Espanya');`

Output pane		
Data Output	Explain	Messages History
	ciudad bpchar	consumo real
1	Sevilla	1000

`select * from p_Consumo_Medio('Sevilla','2');`

Output pane					
Data Output	Explain	Messages	History		
	NIF character(12)	Direccion character(50)	Ciudad character(50)	mes text	ConsumoContador real
1	11111111F	Londres, 334 2°2°	Sevilla	02	1000

`select * from p_Consumo_Electrico_Mensual();`

Output pane				
Data Output	Explain	Messages	History	
	modelo bpchar	numeroseriecontador integer	fechaconexion text	consumo real
1	Modelo 1	1	01	550
2	Modelo 1	1	02	1000
3	Modelo 2	2	05	550.78
4	Modelo 2	2	09	50.45
5	Modelo 3	3	01	150.25
6	Modelo 4	4	12	550.75
7	Modelo 5	5	07	180.9

```
select * from p_List_Cont_Antig(2);
```

Output pane

	modelo bpchar	numeroserie integer	fechainstalacion date
1	Modelo 2	2	2012-01-01

```
select * from p_Top10_Contadores('Barcelona');
```

Output pane

	Modelo character(50)	NumeroSerie integer	ConsumoContador real
1	Modelo 1	1	1000