

# Diseño e implementación de la base de datos de un sistema de control energético.

Presentación: TFC-BASES DE DATOS RELACIONALES

Estudiante: Andrés López Marín

ETIS

Consultor: Manel Rella Ruiz

Data lliurament: 10/06/2012



2011-2012

Segundo semestre

# Introducción

- La Comunidad Económica ha decidido abrir concurso público con el fin de recibir propuestas sobre el diseño de un BD para almacenar información del uso de la energía en el ámbito de la Comunidad.

Estos datos deberán dar soporte a las futuras aplicaciones que serán desarrolladas en una segunda fase del plan de sistemas de control energético.

# Requerimientos - 1

- **A requerimiento de la Comunidad la aplicación deberá :**

1. Guardar los datos asociados a :

- Los contadores
- Las centrales de distribución de energía
- Las centrales de producción
- Las líneas de alimentación
- Los clientes

y

- Las lecturas del consumo de los contadores

2. Guardar el histórico de la energía generada por las centrales de producción

3. Disponer de los procedimientos almacenados siguientes:

- Procedimientos ABM (alta, baja y modificación) de los datos de las distintas entidades.
- Procedimientos de consulta que darán respuesta a los siguientes requerimientos:
  - a) Dada una ciudad y una fecha como parámetros devuelva el listado de los contadores cuyo consumo supere el 80% del consumo medio de todos los contadores de la ciudad, ordenados de forma ascendente en función del porcentaje consumo eléctrico consumido en relación al consumo medio.
  - b) Listado de las 10 centrales de distribución que distribuyen más energía
  - c) Listado de las 10 líneas de comunicación más cargadas en relación con su capacidad máxima de transporte
  - d) Listado de los clientes que disponen de contadores conectados a más de una central de distribución y estas a su vez a más de una línea.
  - e) Dada una central producción y un intervalo de tiempo devuelva el consumo de los contadores que dependen de ella.
  - f) Porcentaje de lecturas realizadas en forma presencial y en forma telemática.
  - g) Listado de los contadores que tengan una determinada antigüedad.

# Requerimientos - 2

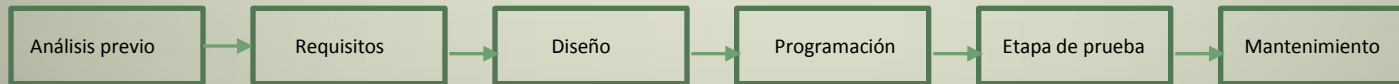
- ***Módulo Estadístico***

También a petición del cliente la aplicación deberá contemplar la implementación de un módulo estadístico que tendrá que dar respuesta, en tiempo constante , a las siguientes necesidades de información.

1. Dada una central de producción, el consumo de los contadores que dependen de la misma.
2. Dada una línea de comunicación y un año concreto, el valor medio de la energía consumida, teniendo en cuenta que este consumo depende de los contadores que se alimentan mediante esta línea. En caso de que el contador este asociado a más de una central de distribución y estas a su vez a más de una línea, consumo total del contador se ha de distribuir entre estas.
3. Línea que ha estado más cargada a nivel de energía consumida.
4. Dado un año concreto, porcentaje de líneas que superan el 50% de energía consumida
5. Dado un año concreto, número de centrales de producción que generan menos del 30% de producción.
6. Listado de los 10 contadores con mayor consumo.
7. Consumo medio de todos los clientes.

# Enfoque y metodología

- Dadas las características del proyecto la metodología a seguir durante el ciclo de vida del mismo será la que se conoce como **ciclo de vida clásico\*** o **en cascada**, el cual consta de las siguientes etapas.

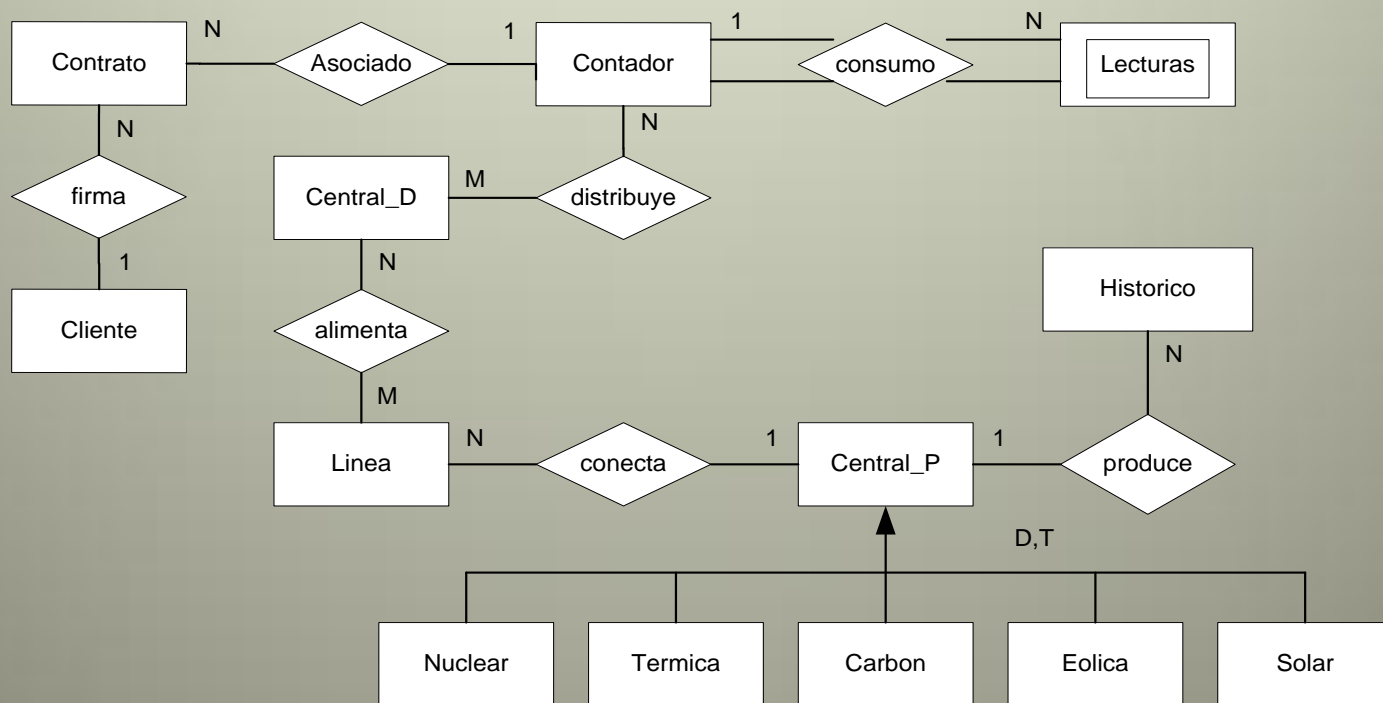


- 1) Análisis previo: En esta etapa se han de definir el software que dará soporte a nuestro proyecto (Sistemas operativos, herramientas CASE, etc.).
- 2) Análisis de requisitos: En esta etapa se han de determinar cuáles son las necesidades de información que el proyecto deberá resolver.
- 3) Diseño: En la etapa de diseño se ha de conseguir dar una solución a las necesidades determinadas durante las etapas de análisis.
- 4) Programación: En esta etapa se han de traducir los resultados de la etapa anterior a código procesable por el ordenador.
- 5) Etapa de pruebas: Como su nombre indica se trata de probar de manera planificada el funcionamiento del Sistema.
- 6) Mantenimiento: A lo largo de la vida del software se tendrán que realizar cambios y correcciones con el fin mejorar el funcionamiento del mismo, además de adaptarlo a los nuevos avances y a los posibles cambios en las necesidades de información.

**\*Nota:** *En el ciclo de vida clásico o en cascada, una fase o etapa solo puede comenzar cuando ha finalizado la anterior, además en esta presentación prescindiremos de las etapas 1 y 2, ampliamente tratadas en la memoria que la acompaña.*

# Diseño-Esquema E/R

- En esta etapa se han identificado las entidades que formarán parte de las tablas principales de la BD, así como las relaciones existentes entre ellas. Todo lo cual queda reflejada en el siguiente esquema E/R.



# Diseño-justificación de la solución propuesta - 1

- **Entidad débil**

- **Lecturas** es entidad débil de **Contadores** puesto que no puede identificarse por si misma ya que en una fecha concreta se pueden realizar lecturas en diversos contadores.
- **Historico** es entidad débil de **Centrales\_P** puesto que para identificarse necesita estar asociada a esta, dado que el valor del mes y el año en el que se produce la producción puede repetirse para diferentes centrales.

- **Generalización especialización**

- Del enunciado del TFC se concluye que se ha de realizar una generalización de la entidad Centrales\_P en las entidades Nuclear, Termica, Carbon, Eolica y Solar puesto que cada una de estas subclases poseen atributos propios. Además la generalización será disjunta y total D/T.

# Diseño-justificación de la solución propuesta - 2

- **Interrelaciones**

- **firma:** se hace uso de esta interrelación para relacionar las entidades CLIENTES y CONTRATOS, con una cardinalidad 1:N justificada por el hecho de que un cliente puede firmar diversos contratos, pero un contrato solo puede estar firmado por un cliente.
- **asociado:** mediante esta interrelación se relacionan las entidades CONTRATOS y CONTADORES, cuya cardinalidad será N:1. Un contador puede históricamente haber estado asociado a varios contratos, mientras que cada contrato estará asociado únicamente a un contador.
- **consumo:** se usa esta interrelación para relacionar la entidad CONTADORES con la entidad débil LECTURAS, su cardinalidad será 1:N debido a que una lectura en una fecha pertenece a un único contador, pero un contador históricamente puede tener múltiples lecturas.
- **distribuye:** mediante esta interrelación se relacionan las entidades CONTADORES y CENTRALES\_D. La cardinalidad es N:M debido a que se entiende que una central de distribución distribuye energía a diversos contadores y un contador puede recibir energía de diversas centrales de distribución.
- **alimenta:** se hace uso de esta interrelación para relacionar las entidades CENTRALES\_D y LINEAS. Su cardinalidad será N:M puesto que una central de distribución puede estar alimentada por diversas líneas y una línea puede alimentar a diversas centrales de distribución.
- **conecta:** mediante esta interrelación se relacionan las entidades CENTRALES\_P y LINEAS, la cardinalidad será de 1:N puesto que según se desprende del enunciado de este TFC una línea solo puede conectar con una central de producción, pero una central de producción puede conectar con diversas líneas.
- **produce:** se usa esta interrelación para relacionar la entidad CENTRALES\_P con la entidad débil HISTORICO, su cardinalidad será 1:N debido a que una central de producción solo puede tener una producción total en un mes, pero en el histórico pueden existir diversas producciones mensuales para una misma central.



# Diseño-Transformación al modelo relacional

Para realizar la transformación de las entidades e interrelaciones vistas en el apartado anterior al modelo relacional, se han tenido en cuenta los siguientes aspectos .

- Una entidad en el modelo ER se transforma en una relación en el modelo relacional.
- En las interrelaciones 1:1 se ha de añadir una clave foránea en cualquiera de las relaciones que referencie a la clave primaria de la otra relación
- En las interrelaciones 1:N se ha de añadir como clave foránea a la relación de cardinalidad N, la clave primaria de la relación de cardinalidad 1.
- En las interrelaciones M:N se ha de crear una nueva relación cuya clave primaria estará formada por las claves primarias de las dos entidades interrelacionadas.
- En las entidades débiles la clave foránea de la relación identificadora forma parte de la clave primaria de la entidad débil.
- En la generalización/especialización se ha de crear una relación para la superclase y para cada una de las entidades subclase.

# Diseño-Transformación de entidades-1

La aplicación de lo visto en la anterior transparencia nos dará como resultado las siguientes transformaciones.

Claves primarias subrayadas con línea continua, claves alternativas subrayadas con línea discontinua, valores nulos en cursiva.

## clientes

id\_cliente, nombre, direccion, localidad, país

## contadores

id\_contador, direccion, localidad, pais, modelo, any\_fabricacion, fabricante, consumo\_acumulado  
{direccion, localidad, pais} clave alternativa

## contratos

id\_contratos, fecha\_alta, *fecha\_baja*, potencia\_contratada, cliente, contador  
donde {cliente} es clave foránea de clientes (id\_cliente)  
donde {contador} es clave foránea de contadores (id\_contador)

## lecturas

contador, fecha, modalidad, lectura\_total, consumo\_mensual  
donde {contador} es clave foránea de contadores(id\_contador)

## centrales\_d

direccion, localidad, pais, suministro\_maximo

## centrales\_p

codigo\_central, direccion, localidad, pais, fecha\_ultima\_inspeccion, maxima\_produccion\_mes  
{direccion,localidad,pais}clave alternativa

Continua.....

# Diseño-Transformación de entidades-2

Continua.....

## **nuclear**

codigo- central\_n, energia\_funcionamiento, kg\_residuos\_radiactivos

donde {codigo\_central\_n} referencia a centrales\_p

## **termica**

codigo- central\_t, kg\_CO2

donde {codigo\_central\_t} referencia a centrales\_p

## **carbon**

codigo- central\_c, kg\_CO2

donde {codigo\_central\_c} referencia a centrales\_p

## **eolica**

codigo- central\_e, numero\_molinos

donde {codigo\_central\_e} referencia a centrales\_p

## **solar**

codigo- central\_s, numero\_paneles

donde {codigo\_central\_s} referencia a centrales\_p

## **lineas**

codigo\_linea, capacidad, central\_p

donde {central\_p} es clave foranea de centrales\_p (codigo\_central)

## **historico**

central\_p, mes, anyo, produccion

donde {central\_p} es clave foranea de centrales\_p (codigo\_central)

# Diseño-interrelaciones y módulo estadístico

## Interrelaciones

### **alimentacion**

direccion, localidad, pais, id\_linea

donde {direccion, localidad, pais} es clave foranea de centrales\_d (direccion, localidad, pais)

donde {id\_linea} es clave foranea de lineas (codigo\_linea)

### **distribuye**

contador, direccion, localidad, pais

donde {contador} es clave foranea de contadores (id\_contador)

donde {direccion, localidad, pais} es clave foranea de centrales\_d (direccion, localidad, pais)

## Diseño lógico del módulo estadístico

### **me\_Central\_Consumo\_contadores**

codigo\_central\_p, consumo\_acumulado

### **me\_valor\_medio\_consumo**

id\_linea, anyo, valor\_medio\_consumo

### **me\_Linea\_mas\_cargada**

identificador, id\_linea, max\_total\_consumo

### **me\_Porcentaje\_lineas\_mas\_50**

anyo, porcentaje\_mas\_50

### **me\_Numero\_centrales\_menos\_30**

anyo, num\_centrales

### **me\_Top\_contadores\_mas\_consumo**

id\_contador

### **me\_consumo\_medio\_clientes**

consumo\_medio

# Codificación

- En las siguientes transparencias se muestran algunos ejemplos del código SQL resultante de transformar el modelo relacional, esta codificación podrá ser consultada en su totalidad en los archivos SQL que acompañan a esta presentación.
- Ejemplo de codificación de la tabla principal *TB\_CENTRALES\_P*

```
CREATE TABLE tb_Centrales_P
(
    codigo_central VARCHAR2(10) NOT NULL,
    tipo_central VARCHAR2(10) NOT NULL,
    direccion VARCHAR2(50) NOT NULL,
    localidad VARCHAR2(50) NOT NULL,
    pais VARCHAR2(25) NOT NULL,
    energia_funcionamiento NUMBER,
    kg_residuos_radiactivos NUMBER,
    kg_CO2 NUMBER,
    numero_molinos NUMBER,
    numero_paneles NUMBER,
    fecha_ultima_inspeccion DATE NOT NULL,
    maxima_produccion_mes NUMBER(16) DEFAULT (0),
    CONSTRAINT pk_centrales_P PRIMARY KEY(codigo_central),
    CONSTRAINT ck_tipo_central CHECK (tipo_central IN('NUCLEAR', 'TERMICA', 'CARBON', 'EOLICA', 'SOLAR'))
);
```

# Codificación SQL-M.estadístico

- Ejemplo de codificación de algunas de las tablas de módulo estadístico

```
CREATE TABLE me_Central_Consumo_contadores
(
    codigo_central_p VARCHAR2(10)NOT NULL,
    consumo_acumulado NUMBER(16,2),
    CONSTRAINT pk_me_central PRIMARY KEY (codigo_central_p)
);
CREATE TABLE me_Valor_medio_consumo
(
    id_linea NUMBER(10)NOT NULL,
    anyo NUMBER(4)NOT NULL,
    valor_medio_consumo NUMBER(10,2),
    CONSTRAINT pk_me_valor_medio PRIMARY KEY (id_linea,anyo)
);
CREATE TABLE me_Linea_mas_cargada
(
    identificador NUMBER(1) NOT NULL,
    id_linea NUMBER(10) NOT NULL,
    max_total_consumo NUMBER(16,2),
    CONSTRAINT pk_me_linea_cargada PRIMARY KEY (identificador)
);
```

# Codificación SQL-Tabla logs

- Tal y como nos solicitaba el cliente, se ha creado también la tabla de **logs** que contendrá, un identificador, autonumérico (*para cuyo mantenimiento se ha generado la correspondiente secuencia*), la fecha y la hora de la ejecución, los parámetros de entrada y de salida si los hubiera, el resultado de la ejecución de los procedimientos (parámetro RSP). Su estructura es la siguiente.

```
CREATE TABLE tb_Logs
(
    id_log NUMBER NOT NULL,
    fecha_proceso VARCHAR2(20) NOT NULL,
    nombre_procedimiento VARCHAR2(30) NOT NULL,
    parametros_in CLOB,
    parametros_out CLOB,
    parametro_RSP CLOB NOT NULL,
    CONSTRAINT pk_logs PRIMARY KEY (id_log)
);
```

```
CREATE SEQUENCE seq_id_logs
CACHE 10;
```

# Especificación de procedimientos

- Como se ha visto en el inicio de esta presentación el cliente nos solicita la creación de una serie de procedimientos almacenados, el código de los cuales puede ser consultado en los ficheros SQL que se adjuntan.

A continuación se muestra la especificación de uno de estos procedimientos cuyo objetivo es el de facilitar el trabajo a los futuros desarrolladores de las aplicaciones de gestión.



# Especificación de procedimientos-2

## Procedimientos ABM

### *pr\_alta\_central\_d*

Procedimiento que nos permite dar de alta una central de distribución a partir de los parámetros de entrada. Devolverá el parámetro RSP que contendrá el resultado de la ejecución del procedimiento.

PARAMETROS IN:     **dir** (variable de tipo Varchar2 que contiene la dirección-calle, avenida, polígono, etc.- de la central)  
                      **loc** (variable de tipo varchar2, contiene la localidad (Barcelona, Madrid, etc) de la central)  
                      **ps** (Variable de tipo varchar2, contiene el país donde está situada la central)  
                      **sumi\_max** (Variable de tipo NUMBER, contiene el máximo de energía permitada por la central)  
                      **RSP** (Variable de tipo CLOB que se corresponde con el parámetro de retorno).

RETURN:             Parámetro **RSP** que contendrá información sobre el resultado de la ejecución del procedimiento.

### EXCEPCIONES:

**ERROR: PRACD001:** La central de distribución ya ha sido dada de alta

**ERROR: PRACD002:** La dirección completa de la central no puede tener valores nulos

**ERROR: PRACD003:** El valor del suministro máximo de la central no puede ser nulo

**ERROR:** Capturamos cualquier otra excepción lanzada por el SGBD.

# Especificación de procedimientos-3

## Procedimientos de consulta

### *pr\_consulta\_g*

Procedimiento que nos permitirá listar el identificador, la dirección, localidad, país y año de fabricación de los contadores que tengan una determinada antigüedad.

PARAMETROS IN:                   **antigüedad**(Variable de tipo NUMBER, contiene los años de antigüedad que queremos consultar.)

**CURSOR\_R** (Variable de tipo SYS\_REFCURSOR, que se corresponde con el parámetro de retorno que contendrá los datos solicitados)

RETURN:                           **CURSOR\_R** (Contendrá el listado de los contadores que tengan un determinado número de años de antigüedad.).

**RSP** contendrá información sobre el resultado de la ejecución del procedimiento.

### EXCEPCIONES:

**ERROR:** Capturamos cualquier otra excepción lanzada por el SGBD.

- **La forma de llamar y utilizar estos procedimientos es la siguiente:**

*Nombre\_procedimiento(parametro\_in\_1, parametro\_in\_2,....., parametro\_out\_1, ....., RSP);*

# Trigger

- Se ha implementado un único *disparador* que actuará al hacer inserciones o modificaciones en la tabla TB\_LECTURAS.
- El disparador llama a los procedimientos PR\_ALTA\_HISTORICO o PR\_MODIFICA\_HISTORICO dependiendo de la acción que se este ejecutando.

Estos dos procedimientos, junto con el procedimiento PR\_MODIFICA\_LECTURA, son los encargados de mantener actualizado el módulo estadístico.

Se ha utilizado el procedimiento PR\_MODIFICA\_LECTURA para actualizar el apartado 4 del módulo estadístico, puesto que al tener que consultar los datos de la tabla que había provocado el disparo del trigger, el SGBD Oracle lanzaba el error asociado a los **triggers** de TABLAS MUTANTES.

El código SQL del disparador es el que podemos ver en la siguiente transparencia.

# Trigger-codificación

```
/*
*Este trigger se disparará al realizar una inserción o una modificación en el campo
*consumo_mes de la tabla TB_LECTURAS
*/

CREATE OR REPLACE
TRIGGER TR_ACTUALIZA_HISTORICO BEFORE INSERT OR UPDATE OF CONSUMO_MES ON
TB_LECTURAS
FOR EACH ROW
DECLARE
    RSP CLOB;
BEGIN
    IF INSERTING THEN
        pr_alta_historico(:new.contador,:new.fecha,:new.consumo_mes, RSP);
    END IF;

    IF UPDATING THEN
        IF(:new.consumo_mes <> :old.consumo_mes)THEN
            pr_modifica_historico(:new.contador, :new.fecha,
            :new.consumo_mes, :old.consumo_mes, RSP);
        END IF;
    END IF;
END;
```

# Orden de ejecución de ficheros

- Los ficheros SQL que se entregan junto con esta presentación tienen el siguiente orden de ejecución.
  - ❖ tablas principales.sql
  - ❖ tablas m\_estadistico.sql
  - ❖ Procedimientos de alta.
  - ❖ Procedimientos de modificación
  - ❖ Procedimientos de baja
  - ❖ Procedimientos de consulta
  - ❖ Trigger
  - ❖ prueba\_altas.sql (modificado)
  - ❖ prueba\_modificaciones.sql
  - ❖ prueba\_bajas.sql
- **Observaciones:** Los juegos de prueba de los procedimientos de consulta (**pruebas\_consulta.sql**), de comprobación de errores (**prueba\_de\_errores.sql**) y de comprobación del módulo estadístico (**prueba\_modulo\_estadistico.sql**), se puede ejecutar en cualquier momento una vez cargado el fichero **prueba\_altas.sql**.