

DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS DE CONTROL ENERGÉTICO.



Trabajo de Fin de Carrera
Estudiante: EDUARD MONZONIS HIERRO.

emonzonis@uoc.edu

Titulación: Ingeniería Técnica Informática de Gestión.
Consultor: Ismael Pérez Laguna

INTRODUCCION

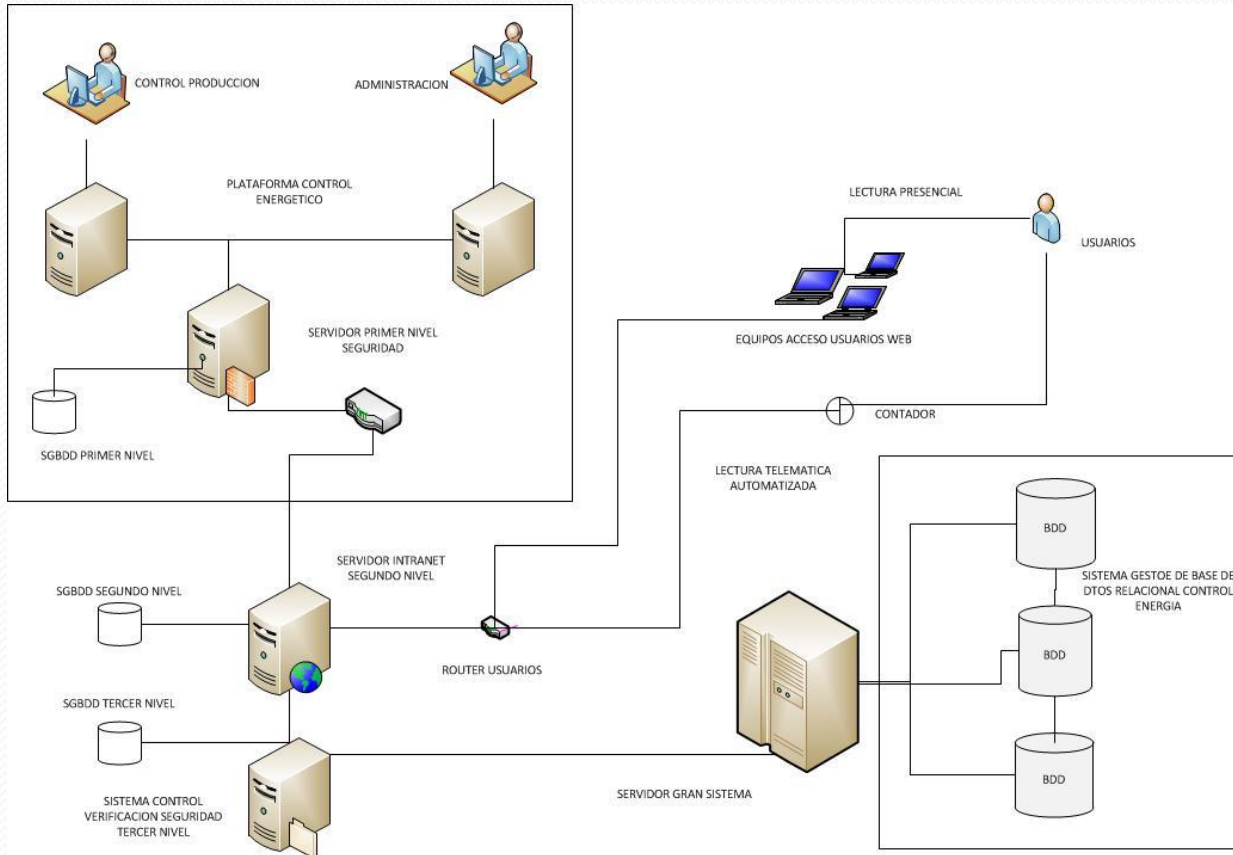
- **Objetivo principal** : Consolidar y ampliar los conocimientos adquiridos a lo largo de la carrera de Ingeniería Técnica Informática de Gestión en el área de las bases de datos relacionales.
- **Proyecto:** Diseño y desarrollo de una BD que permita la generación de datos estadísticos sobre la energía para controlar el uso de energía en el marco de la Comunidad Económica Europea
- **Presentación:** Expone a modo de síntesis el proyecto de Bases de Datos Relacional desarrollada bajo la Plataforma ORACLE 10g con el lenguaje declarativo PL/SQL como herramienta de implementación .El desarrollo se ha seguido la metodología en cascada en la fases de análisis hasta culminar con el producto final siempre orientado a la funcionalidad y la accesibilidad.
- **La memoria:** Se adjunta memoria con todos los detalles del proyecto.

DOCUMENTACIÓN ENTREGADA

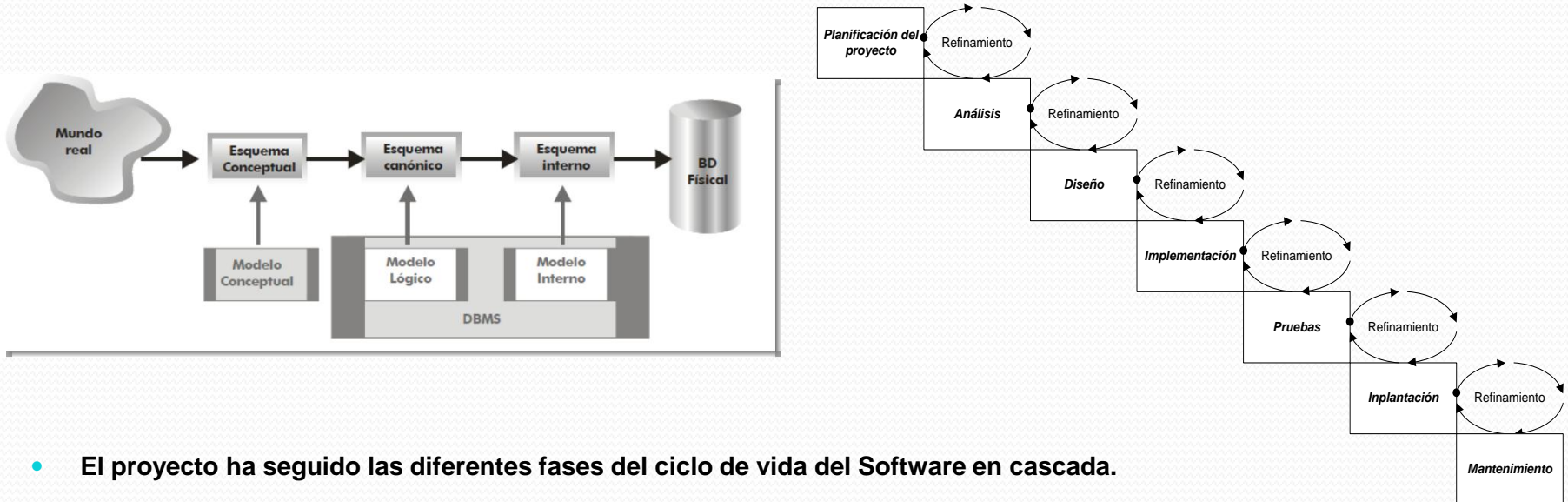
Adjunto en esta presentación sean librado otros documentos :

- ✓ Memoria del Proyecto.
- ✓ Código de creación de las entidades o tablas, secuencias y disparadores o Triggers.
- ✓ Código de los Package que configuran los diferentes módulos.
- ✓ Código para la carga de datos.
- ✓ Código de carga de datos masiva.
- ✓ Juegos de pruebas.
- ✓ Síntesis de la presentación.

FLUJO DE LA INFORMACIÓN

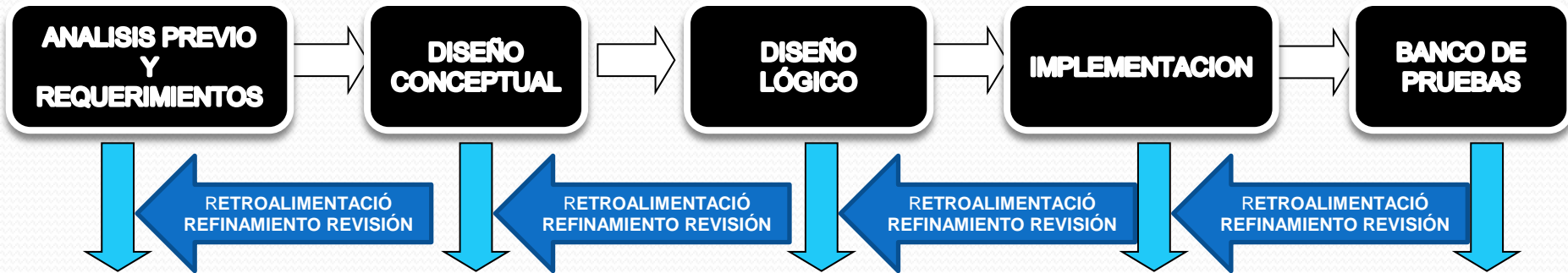


CICLO DE VIDA DE SOFTWARE



- El proyecto ha seguido las diferentes fases del ciclo de vida del Software en cascada.
- En cada fase el principio de refinamiento ha sido seguido con absoluta cautela con el fin de cumplir con las necesidades requeridas por el cliente. Sin dejar de lado la retroalimentación entre ellas para conseguir un producto con máxima calidad y excelencia.
- Durante las fases sea estandarizado y normalizado, con el fin de cumplir con la reutilización, portabilidad, fiabilidad y mantenimiento. Así conseguir una reducción de costes en las futuras modificaciones que el cliente o usuario final necesite.

ETAPAS DEL DISEÑO DE LA ARQUITECTURA DE LA BASE DE DATOS.



En esta etapa o fase se ha realizado un primer estudio de cuales son los objetivos del proyecto del Sistema de Base de Datos Relacional centrada al usuario final.

En el diseño conceptual por refinamiento obtenemos una estructura de la base de datos independiente de la tecnología que se utilizará. Se hace un estudio de las entidades extraídas de las especificaciones iniciales y del mundo real en que intervienen.

El diseño lógico mediante sucesivos refinamientos para ser adaptado fielmente a la plataforma tecnológica que se quiere utilizar en la siguiente etapa según las especificaciones requeridas.

Se procede a la implementación mediante el lenguaje específico de la plataforma tecnológica ORACLE 10g como Sistema de Gestión de Base de Datos. (SGBD).

La etapa final, se generan diferentes juegos de pruebas para poder ver el comportamiento del producto desarrollado. Se hace un escrupuloso seguimiento y análisis de las extracciones hasta alcanzar el producto finalizado.

ETAPA 1.- ANALISIS PREVIO DE REQUERIMIENTOS.



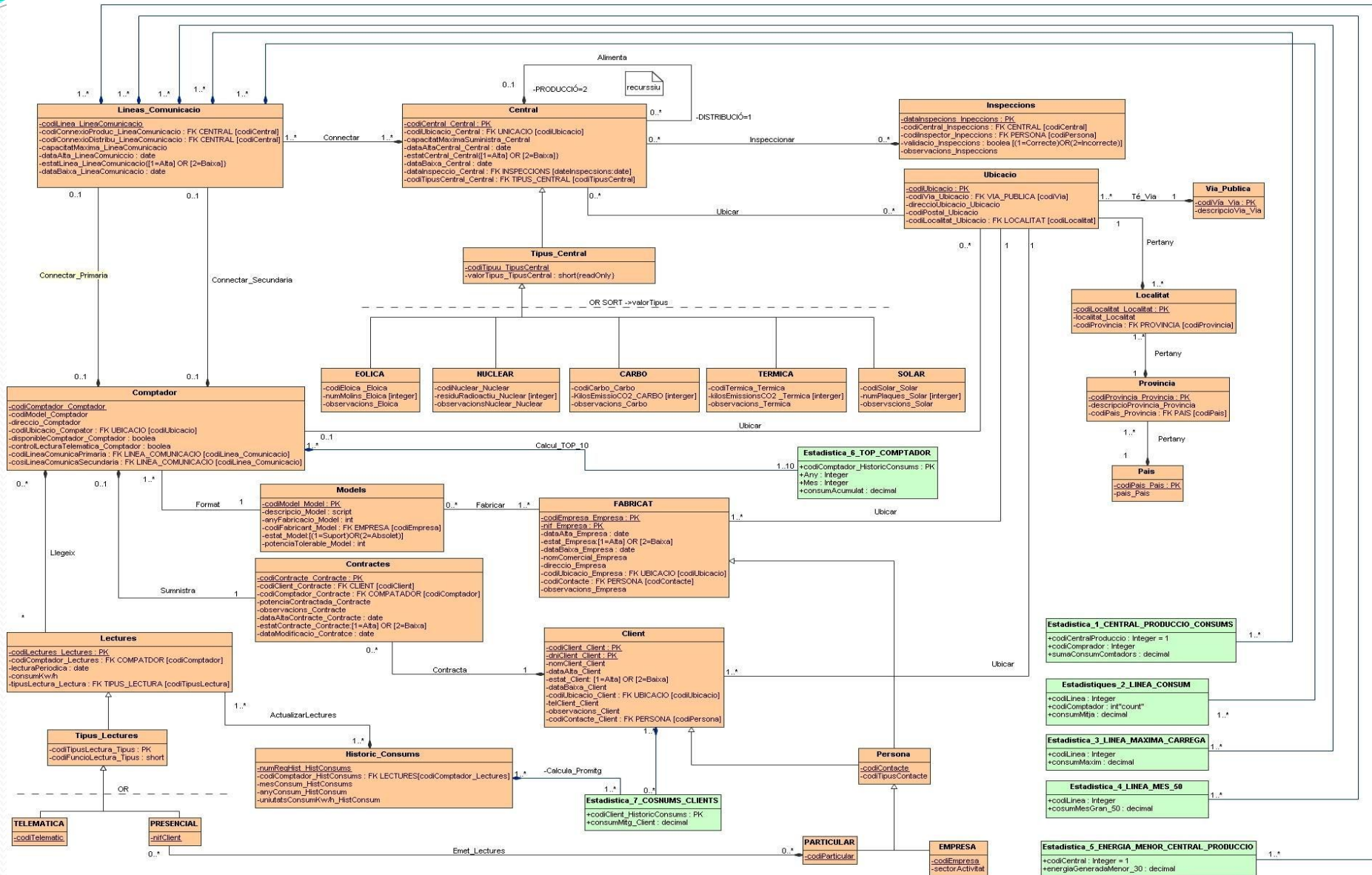
1. **ANALITZAR:** Puntos básicos del que sea de hacer en el proyecto a desarrollar utilizando las herramientas de diseño y los aplicativos necesarios.
2. **DEFINIR:** el detalle de las necesidades de información que se deberán de resolver mediante la implementación en el lenguaje PL/SQL.
3. **DOCUMENTARSE:** referente los conocimientos generales que serán necesarios sobre el dominio de la aplicación y el lenguaje de programación que se tendrá que utilizar, hacer un esbozo de los algoritmos que deberán de generarse en el desarrollo, analizando siempre en contenido que usuario necesita y teniendo presente las funcionalidades que deberá de tener el futuro software .
4. **DESARROLLAR:** una temporalidad de objetivos mediante un esquema a seguir para conseguir hitos con el fin de crear una aplicación que no incida en desviaciones de costes y que el producto final sea una aplicación completa, eficiente, optimizada y robusta.

ETAPA 2.- DISEÑO CONCEPTUAL MODELO ENTIDAD RELACIÓN.

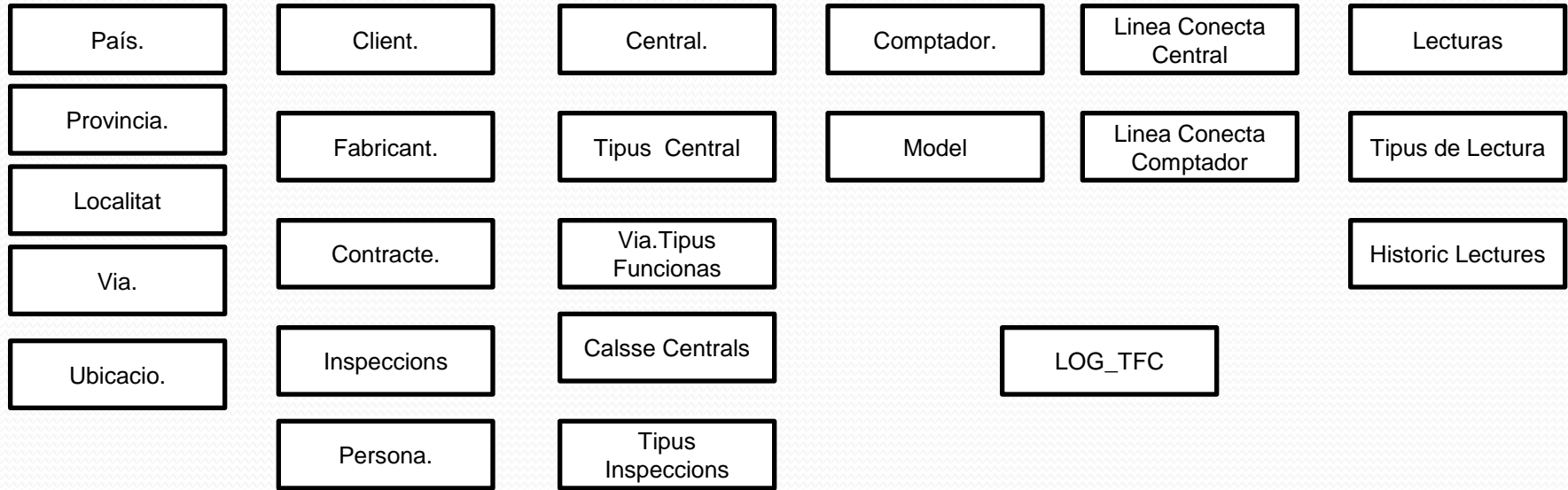


1. **IDENTIFICACIÓN Y DESCRIPCIÓN DE LAS ESTRUCTURAS DE DATOS:** en esta fase o etapa se lleva a termino el desarrollo del esquema conceptual que tendrá la aplicación.
2. **ESTRUCUTRA:** deberá de resolver todo lo que el cliente en las especificaciones iniciales precisa y solicita que haga la aplicación
3. **FINLIDAD:** aportar conocimiento, poder extraer estadísticas y consultas requeridas por el cliente. Dado que el objetivo final es la obtención de estadísticas coherentes , ha sido necesario realizar una estructura acorde con los planteamientos exigidos por el usuario final de la aplicación.

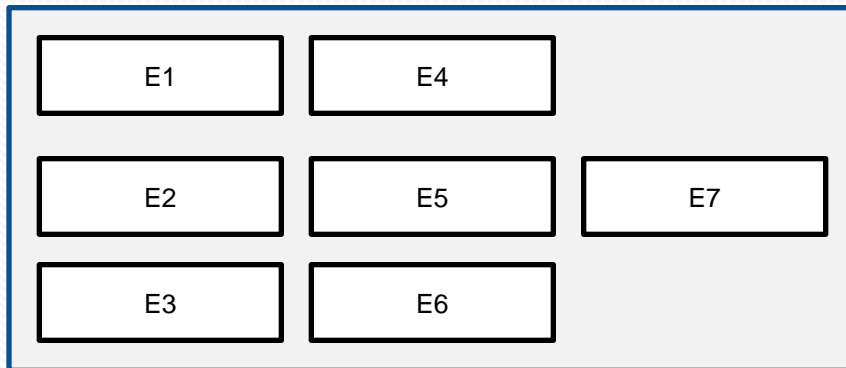
DIAGRAMA MODELO E/R



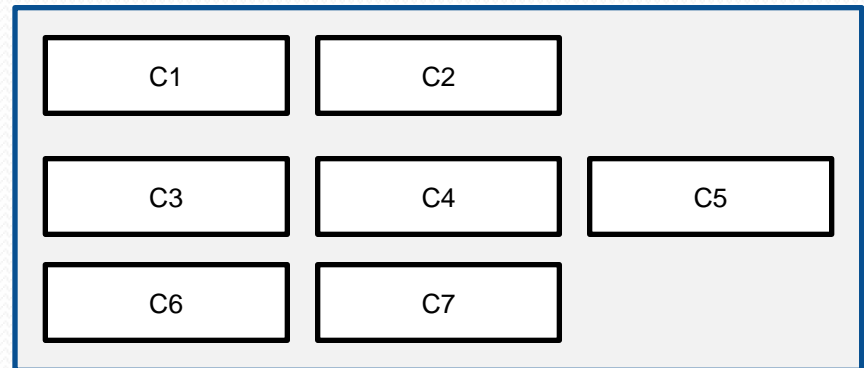
CONJUNTO DE ENTIDADES



Modulo Estadísticas



Modulo Consultes



ETAPA 3.- DISEÑO CONCEPTUAL MODELO ENTIDAD RELACIÓN.



Partiendo de diseño conceptual se efectúa la transformación en una estructura de datos del modelo relacional.

Hay dos tipologías en su transformación a llevar a cabo: la de entidades i la de interrelaciones.

1. **TRANSFORMACIÓN DE ENTIDADES:** cada entidad del modelo ER se ha transformado en una relación del modelo relacional. Los atributos de la relación y la clave primaria de la entidad es la clave primaria de la relación.
2. **TRANSFORMACION DE INTERRELACIONES:**
 - Conectividad 1:N** donde se ha añadido la cardinalidad a la relación del lado N a una clave forana que referencia a la otra relación.
 - Conectividad M:N** conectividad la cual sea transformado en una nueva relación. Su clave primaria está configurada por los atributos de la clave primaria de las dos entidades que intervienen en su relación. Los atributos de la interrelación son los atributos de la nueva relación.
 - Especialidad – Generalización** La relación de la entidad superclase tiene como clave primaria la de la superclase y contiene todos los atributos comunes a ambas. Las relaciones de las subclases tienen como clave primaria la de la entidad superclase y contiene los atributos específicos de la subclase.



COMPONENTES LÓGICOS DE DATOS:

1. **TABLAS:** Sea realizado una implementación con lenguaje SQL de cada una de las tablas, con sus correspondientes claves primarias y foranas junto con las restricciones exigidas y detectadas.
2. **SECUENCIAS I TRIGGERS:** Sean creado las secuencias para cada una de las tablas que concursan en el sistema con el fin de generar de forma secuencial los identificadores unívocos o claves primarias durante el proceso de inserción de datos en las tablas, así mismo los triggers o disparadores que ejecutan dichos secuenciadores.
3. **FUNCIONES:** su finalidad que retornan un texto y son validados sus datos o números que contienen.
4. **PROCEDIMIENTOS:** han sido agrupados en estructuras ordenadas según su funcionalidad y usabilidad de los usuarios. Estas estructuras nombras PACKAGE contienen procedimientos en cada una de ellas los cuales ofrecen la funcionalidad en las extracciones, inserciones y manipulaciones de datos, así mismo en dichas estructuras sea efectuado un cuidadoso control de errores, lanzando los mismo en excepciones enunciativas visualmente con el fin de poder gestionar de forma eficiente los tratamientos de los datos integrados en las tablas, dicha depuración de control sea ha realizado tanto en el procedo de entrada de datos como en la salida de los mismos. El control valida o comprueba tanto los parámetros de entrada como los de salida mediante las propias sentencias que incluye estas estructuras.
 - **Parámetros de salida:** como parámetros de salida , todos lo procedimientos retornan el resultado de las ejecuciones efectuadas.
 - ❖ **Ok si la ejecución es satisfactoria.**
 - ❖ **ERROR: + descripción** si se produce un comportamiento anómalo o incorrecto.



5. REGISTRO DE EJECUCIONES:

La aplicación lleva un control de las ejecuciones realizadas por los procedimientos ejecutados . Todos lo procedimientos registran en tiempo real su ejecución en la tabla LOG_TFC, actúa como un cuaderno de bitácora de la aplicación.

6. AGRUPACION MODULAR DE LAS RELACIONES DE PACKAGES:

Sea diversificado según su utilidad y funcionalidad en módulos en los cuales coexisten los PACKAGE con las correspondientes funcionalidades procedimentales necesarias, en total sean creado 78 ficheros sql de código para estos PACKAGES

El criterio seguido en esta ordenación ha sido en la valoración del nivel crítico que cada uno de los elementos intervienen en el proceso de explotación de datos y el ámbito de negocio que precisan los mismos.

DISSEÑO EN FASE DE IMPLEMENTACIÓN

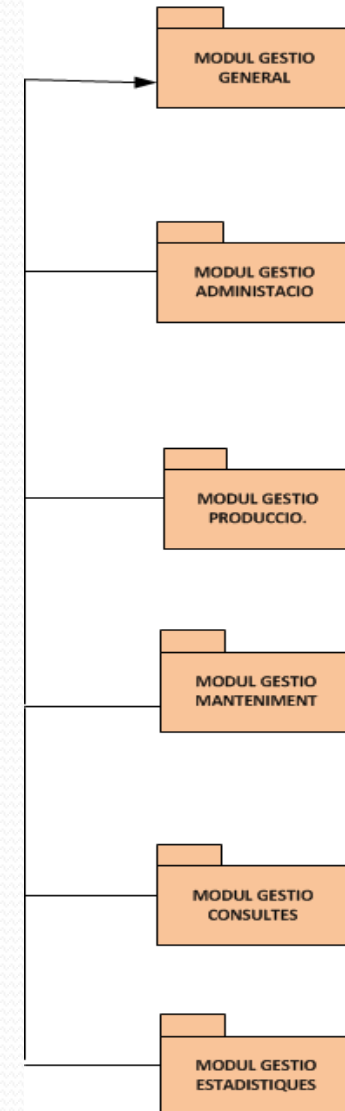
En la fase de implementación sea definido el proyecto en seis puntos fundamentales organizados de forma modular:

- MODUL GESTIO GENERAL.
- MODUL GESTIO ADMINISTRACIO.
- MODUL GESTIO PRODUCCIÓ.
- MODUL GESTIO MANTENIMENT.
- MODUL CONSULTES.
- MODUL ESTADISTIQUES.

Cada modulo contiene los Packages implementados en PL/sql que gestionan las funcionalidades de la entidades.

Se justifica esta estructura modular para dar una mejor cobertura y facilidad en la fase de implementación del producto final conseguir así una mayor eficiencia de utilización de los diferentes scripts e incidiendo en reducción de costes cuando deba de confeccionarse la interfície final del proyecto, independientemente de la plataforma de programación que se decida en la segunda fase en que el equipo de programadores deberá de desarrollar el producto final.

ESTRUCUTRA MODULAR DE LA APLICACIÓ



MODULOS CLASIFICADOS SEGÚN FUNCIONALIDADES

Contenido de los ficheros que efectúan todas la funcionalidades exigidas clasificados en módulos según las funcionalidades que el usuario o usuarios deberán de llevar a termino a nivel de negocio.

MODUL GESTIO GENERAL.

1. GESTIO_GENERAL_LOG_TFC.sql
2. PACKAGE_GENERAL_LOG_TFC.sql

MODUL GESTIO ADMINISTRACIO.

1. GESTIO_CLIENT.sql
2. GESTIO_CONTRACTES.sql
3. GESTIO_FABRICANT.sql
4. PACKAGE_GESTIO_CLIENT.sql
5. PACKAGE_GESTIO_CONTRACTES.sql
6. PACKAGE_GESTIO_FABRICANT.sql

MODUL GESTIO MANTENIMENT.

1. GESTIO_ESTAT.sql
2. GESTIO_LOCALITAT.sql
3. GESTIO_PAIS.sql
4. GESTIO_PERSONA.sql
5. GESTIO_PROVINCIA.sql
6. GESTIO_UBICACIO.sql
7. GESTIO_VIA.sql
8. PACKAGE_GESTIO_ESTAT.sql
9. PACKAGE_GESTIO_LOCALITAT.sql
10. PACKAGE_GESTIO_PAIS.sql
11. PACKAGE_GESTIO_PERSONA.sql
12. PACKAGE_GESTIO_PROVINCIA.sql
13. PACKAGE_GESTIO_UBICACIO.sql
14. PACKAGE_GESTIO_VIA.sql

MODUL GESTIO PRODUCCIÓ.

1. GESTIO_CENTRAL.sql
2. GESTIO_CLASSE_CENTRAL.sql
3. GESTIO_COMPTADOR.sql
4. GESTIO_CONECTA_CENTRALS.sql
5. GESTIO_CONECTAR_COMPTADOR.sql
6. GESTIO_HIST_LLECTURES.sql
7. GESTIO_INSPECCIONS.sql
8. GESTIO_LLECTURES.sql
9. GESTIO_LINEA_TIPUS.sql
10. GESTIO_MODEL.sql
11. GESTIO_TIPUS_CENTRALS.sql
12. GESTIO_TIPUS_FUNCIONS.sql
13. GESTIO_TIPUS_INSPECCIONS.sql
14. GESTIO_TIPUS_LLECTURA.sql
15. PACKAGE_GESTIO_CENTRALS.sql
16. PACKAGE_GESTIO_CLASSE_CENTRAL.sql
17. PACKAGE_GESTIO_COMPTADOR.sql
18. PACKAGE_GESTIO_CONECTA_CENTRALS.sql
19. PACKAGE_GESTIO_CONECTA_COMPTADORS.sql
20. PACKAGE_GESTIO_HIST_LLECTURES.sql
21. PACKAGE_GESTIO_INSPECCIONS.sql
22. PACKAGE_GESTIO_LLECTURES.sql
23. PACKAGE_GESTIO_LINEA_TIPUS.sql
24. PACKAGE_GESTIO_MODEL.sql
25. PACKAGE_GESTIO_TIPO_CENTRAL.sql
26. PACKAGE_GESTIO_TIPUS_FUNCIONS.sql
27. PACKAGE_GESTIO_TIPUS_INSPECCIONS.sql
28. PACKAGE_GESTIO_TIPUS_LLECTURA.sql

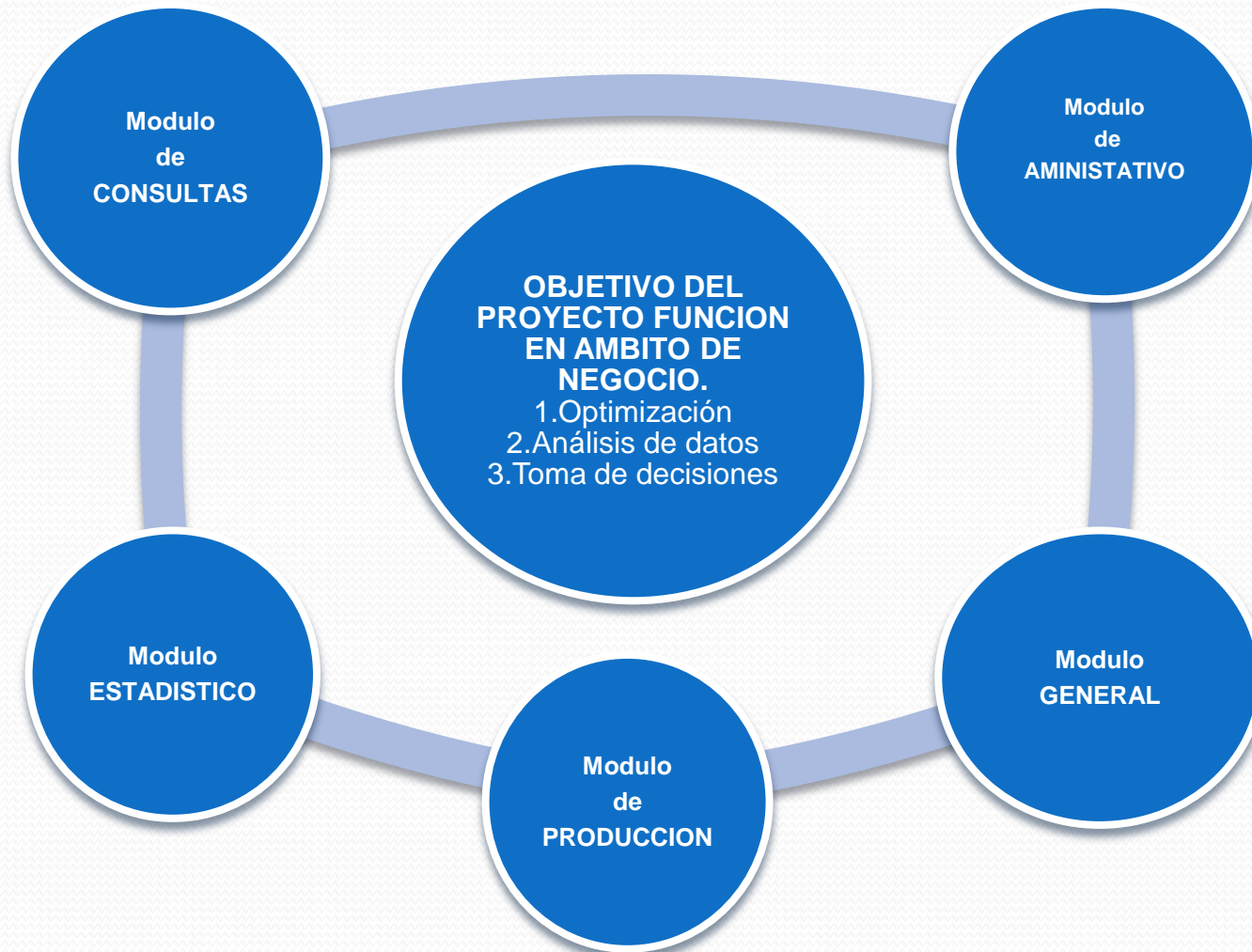
MODUL CONSULTES.

1. GESTIO_C1.sql
2. GESTIO_C2.sql
3. GESTIO_C3.sql
4. GESTIO_C4.sql
5. GESTIO_C5.sql
6. GESTIO_C6.sql
7. GESTIO_C7.sql
8. PACKAGE_GESTIO_C1.sql
9. PACKAGE_GESTIO_C2.sql
10. PACKAGE_GESTIO_C3.sql
11. PACKAGE_GESTIO_C4.sql
12. PACKAGE_GESTIO_C5.sql
13. PACKAGE_GESTIO_C6.sql
14. PACKAGE_GESTIO_C7.sql

MODUL ESTADISTIQUES.

1. GESTIO_E1.sql
2. GESTIO_E2.sql
3. GESTIO_E3.sql
4. GESTIO_E4.sql
5. GESTIO_E5.sql
6. GESTIO_E6.sql
7. GESTIO_E7.sql
8. PACKAGE_GESTIO_E1.sql
9. PACKAGE_GESTIO_E2.sql
10. PACKAGE_GESTIO_E3.sql
11. PACKAGE_GESTIO_E4.sql
12. PACKAGE_GESTIO_E5.sql
13. PACKAGE_GESTIO_E6.sql
14. PACKAGE_GESTIO_E7.sql

SINERGIAS DEL FLUJO DE INFORMACIÓN





Ejemplo de carga y consulta:

```

-- EXECUTA INSERTAR VIA
DECLARE
sortida VARCHAR(500):="";
BEGIN
    GESTION_VIA.PRC_ALTA_VIA('Avda._1',sortida);
    GESTION_VIA.PRC_ALTA_VIA('C./',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Pge.',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Pol. Ind.',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Ctra.',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Passeig',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Camí',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Travessera',sortida);
    GESTION_VIA.PRC_ALTA_VIA('Rbla.',sortida);
END
/
-- CONSULTA DE LES VIES PUBLIQUES
DECLARE
sortida VARCHAR(500):="";
BEGIN
    GESTION_VIA.PRC_CONSULTA_VIA(sortida);
END;
/
    
```

INSERCIÓN DE DATOS Y REGISTRO DE PRUEBAS:

Se han insertado multitud de datos de forma masiva y recurrente las diferentes tablas mediante los procedimientos diseñados a su finalidad. En esta etapa sea intentado en todo momento acercarse al máximo a la realidad en que el usuario deberá de encontrarse durante su utilización. La simulación del banco de pruebas sea llevado de forma ordenada con el fin de evitar comportamientos imprevisibles durante las ejecuciones, en la memoria del proyecto se explicita de forma secuenciada tanto la creación de los componentes como su utilización.

En el conjunto de ficheros entregados lleva adjunto un fichero con el nombre de CARREGA.sql el cual contiene las ejecuciones y los datos utilizados en la fase de diagnostico y de pruebas de esta aplicación.

Aspecto Dinámico Empresarial del Proyecto

Instalación del Sistema
 Adaptación consolidada.

Aspecto física del Sistema de BDD.

Optimización del mismo.

Formación de usuarios período de adaptación.

Compromiso en prestar Soporte al Sistema.

Mantenimiento constante y mejoras futuras.

CONCLUSIONES FINALES

1. El producto final cumple los requisitos iniciales y las funcionalidades requeridas que se esperaban.
2. Se ha normalizado la Base de Datos al máximo con el fin de conseguir la optimización de los procesos en las extracciones, ganando en productividad y reducción de costes.
3. El diseño estructurado y modular ha sido estandarizado y pensado para de reducir costes que incidan en las siguientes fases del futuro producto final, fases que en este TFC no entran a formar parte de su desarrollo.
4. Sean incluido algunas automatizaciones de procesos durante las ejecuciones, las cuales reducen el nivel de errores posibles y una mayor fiabilidad, la robustez del producto.
5. Este trabajo ha servido para demostrar mi capacidad de hacer frente a un proyecto de cierta complejidad, haciendo uso del rigor y metodología de los conocimientos, técnicas y habilidades que he adquirido a lo largo de mis estudios de Ingeniería Informática de Gestión en la UOC.

Toda la dedicación en horas y el esfuerzo que ha supuesto en la elaboración de este proyecto, he podido tener la oportunidad de disfrutar en plenamente en desarrollar un tema tan apasionante y de gran sensibilidad pública..