

DISEÑO E IMPLEMENTACIÓN DE LA BASE DE DATOS DE UN SISTEMA DE CONTROL ENERGÉTICO

Nombre del Estudiante: *Alberto Ramos Saguillo*

ETIS

Nombre del Consultor: Manuel Rella Ruiz

Objetivos

- **Objetivo principal** : Consolidar y ampliar los conocimientos adquiridos a lo largo de la carrera de Ingeniería Técnica Informática en el área de las bases de datos relacionales.
 - Para alcanzar el objetivo propuesto:
 - ✓ Se diseña la Base de Datos (Diagramas conceptuales, lógicos), se crea e implementa todo el código correspondiente para realizar las tareas descritas en el enunciado.
 - ✓ Se implementa un mecanismo de inicialización de la base de datos y se crean juegos de pruebas exhaustivos que permita garantizar y demostrar el buen funcionamiento de las funciones implementadas así como el control de errores y de situaciones de excepción.

Documentos a entregar

- **Plan de Trabajo:** planificación y estimación de las tareas necesarias para llevar a cabo los objetivos previstos.
- **Producto:** scripts de creación de BBDD, procedimientos almacenados.
- **Pruebas :** juego de pruebas exhaustivo para verificar el correcto funcionamiento de la base de datos.
- **Memoria:** documento de síntesis final.
- **Presentación:** resumen claro y conciso del trabajo realizado y de los resultados obtenidos.

Requisitos

- ✓ Analizar los requerimientos del enunciado, manteniendo comunicación directa con el cliente.
- ✓ Los puntos que no quedan suficientemente claros son consultados con compañeros del aula.
- ✓ Detectar funcionalidades a partir de la descripción del cliente
- ✓ Requisitos cortos, claros y concisos.

Requisitos no funcionales

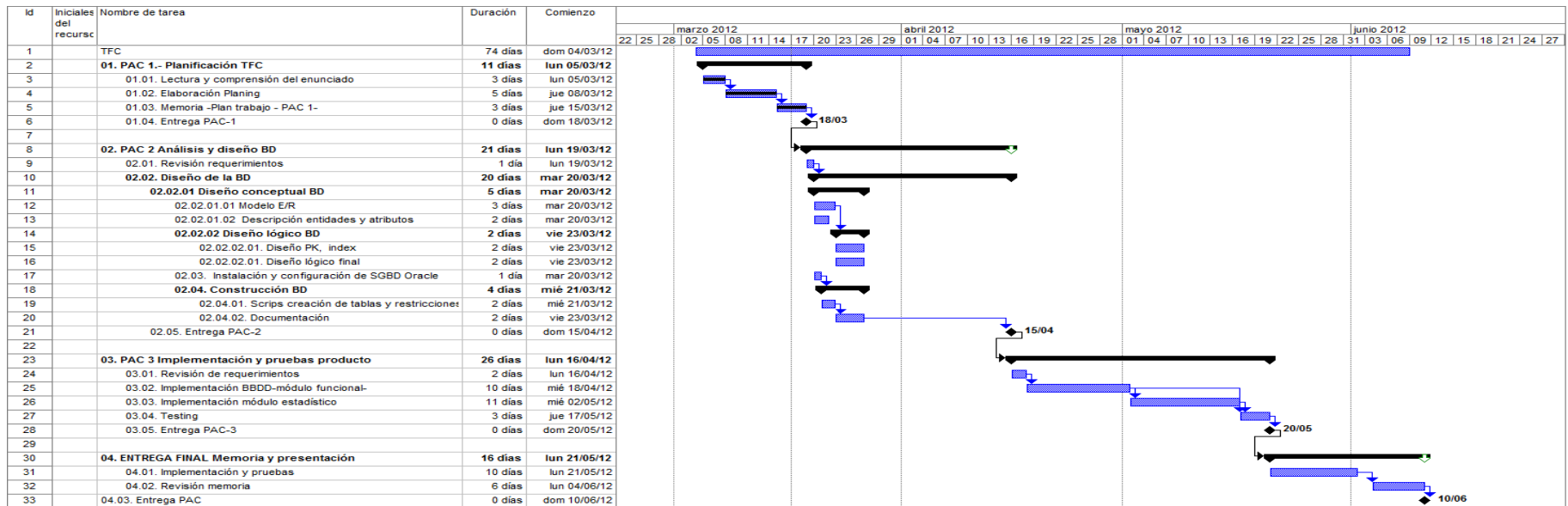
- El diseño de este sistema no contempla el de una interface gráfica para el usuario final del producto.
- El sistema debe tener un comportamiento estable, informando de la correcta ejecución o bien dando el mensaje codificado de error, que podrá consultarse en la tabla correspondiente.
- Los accesos a la base de datos se realizan utilizando el entorno PL/SQL Developer, o bien utilizando los procedimientos creados para el resto de las funcionalidades.

Otros Requisitos

- El sistema de Gestión de Base de Datos escogido será Oracle.
- Portabilidad del sistema sobre hardware convencional, no específico.
- Integridad y consistencia de la Base de Datos, buscando las normalizaciones standards, evitando redundancia de datos y mantenimiento complejo.
- Eficiencia en la gestión del espacio de ocupación de memoria y disco.
- Los procedimientos almacenados contarán con el tratamiento de las excepciones que puedan producirse al computar entradas no esperadas.
- La política de permisos seguida, es común a todos los usuarios, tendrán acceso a todas las funcionalidades; por lo que no se generan perfiles específicos.
- No se imponen restricciones al sistema operativo.
- No se contempla la realización de tareas periódicas de backup de los datos almacenados en la Base de Datos.
- Se crea una tabla LOG, para tener registro de todas las transacciones de la Base de Datos .
- En sistema llevará un control depurativo de excepciones que pueden generarse durante la manipulación de los datos o en la introducción de los mismos.

Temporización

- Fecha inicial proyecto: 29/02/2012
- Fecha final proyecto: 10/06/2012



Fecha	Nombre del Hito	Descripción
29/02/2012	Inicio del curso	Fecha de inicio del curso
05/03/2012	Trovada	Encuentro presencial -orientación académica-
18/03/2012	PAC 1	Entrega de la primera parte de la memoria
15/04/2012	PAC 2	Entrega de la segunda parte de la memoria
20/05/2012	PAC 3	Entrega de la tercera parte de la memoria
21/05/2012	Inicio final TFC	Inicio enunciado del final de TFC
10/06/2012	Entrega final TFC	Final entrega producto, pruebas y memoria

Diseño del sistema

Diseño conceptual

- **Esquema conceptual.**

- Desarrollamos el esquema conceptual de la organización de la BD independientemente del Sistema Gestor de Base de Datos (SGBD).
- Obtenemos al diagrama UML, como resultado de esta etapa

Diseño lógico

- **Lógico - Relacional**

- Transformaremos el modelo UML, utilizando como tecnología un SGBD relacional.
- Intenta refinar el diseño anterior, presentándolo de una forma algo más afín, contemplamos
- Claves primarias y externas
- Restricciones (PK, FK, NOT NULL)

Diseño físico

- **Creación de la base de datos**

- Para ello partiremos del diseño lógico y aplicaremos los conocimientos concretos que tenemos sobre el SGBD
- Buscaremos optimizar la eficiencia y completar el modelo con aspectos físicos dependientes del SGBD concreto utilizado

Lógico-relacional

- El diseño lógico-relacional refina el anterior con claves externas.
- Del resultado de las modificaciones, se incorporan nuevas entidades.
- Se simplifica e independiza el mantenimiento en la gestión de codificación.
- Las bajas se formalizan lógicamente. Cambia el estado.
- Se decide la creación de las tablas de consultas, estadísticas y el log de la BD.

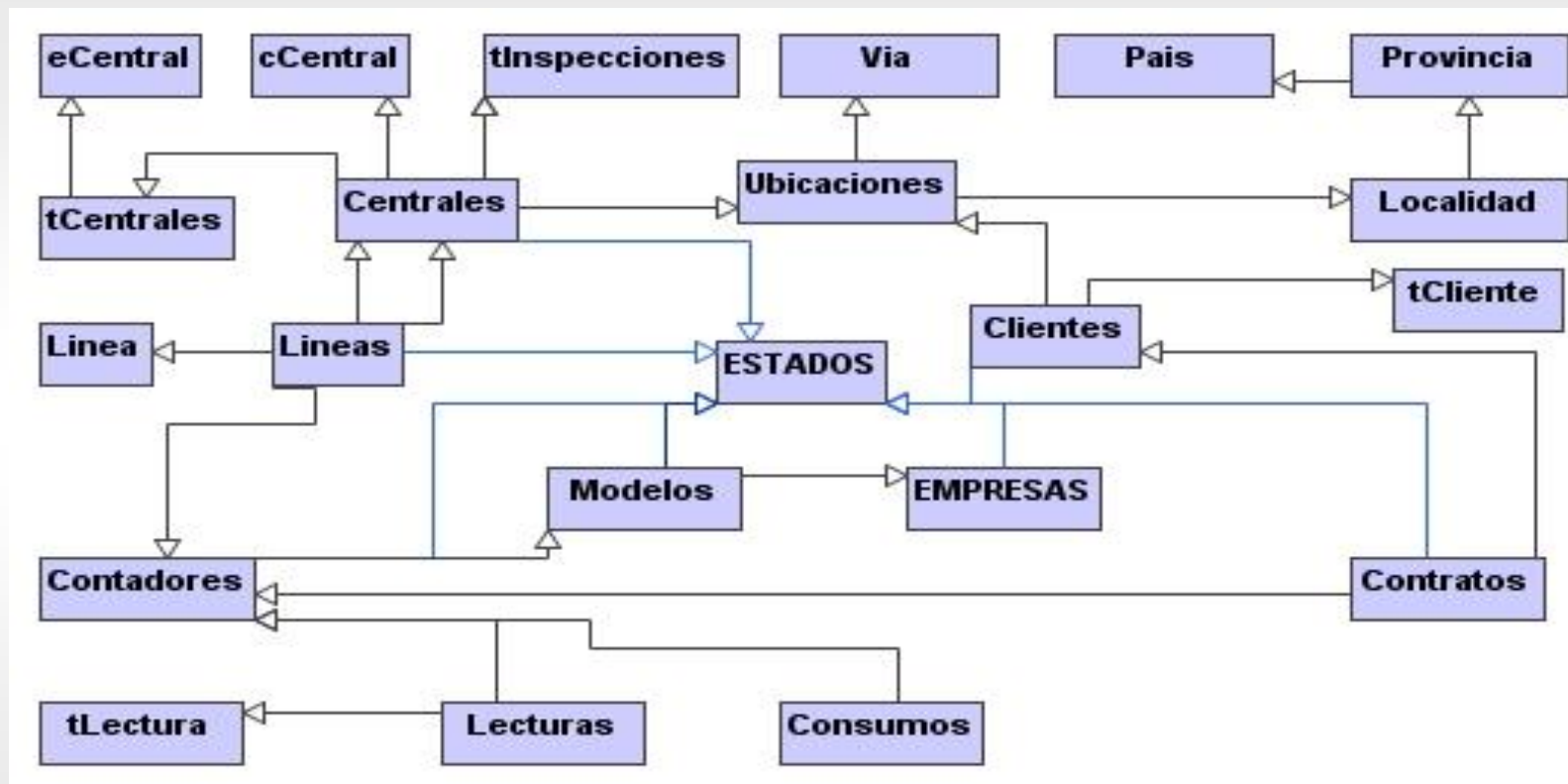
Diseño físico

- ❖ A partir de los pasos anteriores se decidió como se debía crear nuestro SBGD
 - **Objetivos**
 - ✓ Disminuir el tiempo de respuesta
 - ✓ Minimizar espacio en el disco
 - ✓ Seguridad de Datos
 - ✓ Optimizar Recursos

 - **Tabla de Log**
 - ✓ Tabla destinada a guardar histórico de llamadas a procedimientos y su resultado

Creación de tablas

Creamos las tablas según el diseño anterior



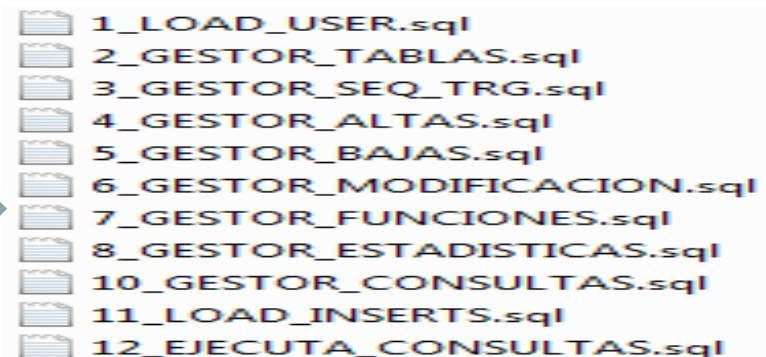
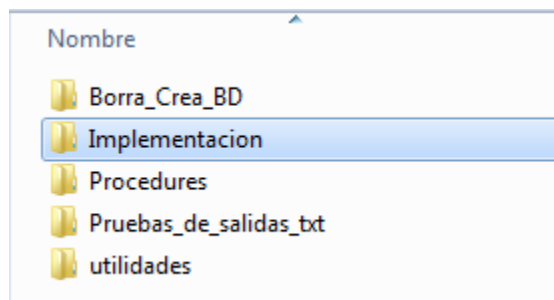
Estadísticas

- Módulo Estadístico
 - Diseño similar a un datawarehouse
 - Después de las lecturas de contadores, actualizamos consumos y las tablas de estadísticas correspondientes. Obtendremos consultas inmediatas e históricos para análisis.



Procedimientos

- ✓ Se utilizan procedimientos almacenados organizados en gestores según la funcionalidad.
- ✓ Se organizan en carpetas con las funcionalidades específicas.
- ✓ Se generan dos scripts para pruebas completas de la BD. El primero comprueba todos los procedimientos y el segundo, con carga suficiente de datos, para generar resultados, estadísticas y consultas.



Resultado Procedimientos

Características:

Parámetros de salida:

- ✓ 'OK' : si todo bien + descripción
- ✓ 'ERROR: ' + descripción de error -> Excepción

Se comprueban los parámetros de entrada

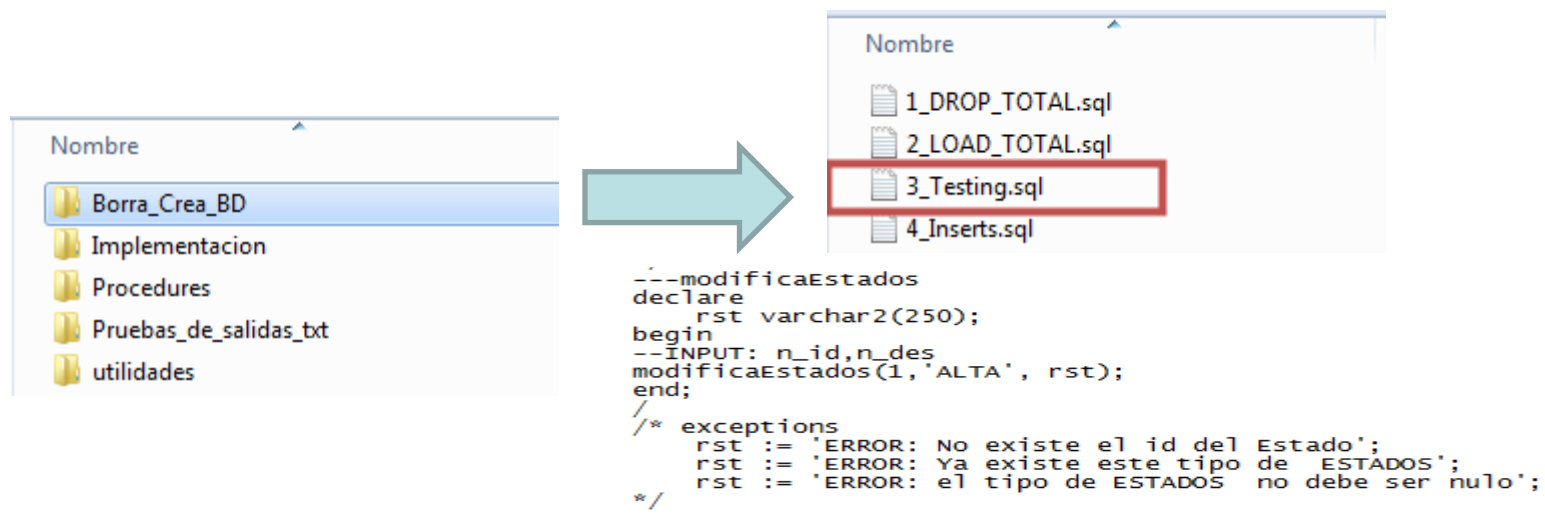
- ✓ Excepción si algún parámetro necesario es nulo
- ✓ Excepción si no se cumplen las condiciones necesarias

Se realizan las operaciones del procedimiento

- ✓ Si todo va bien: mensaje por pantalla.
- ✓ Se inserta una entrada en la tabla de Logs
- ✓ Contenido: fecha, nombre procedimiento, parámetros de entrada separados por comas, salida ('OK')

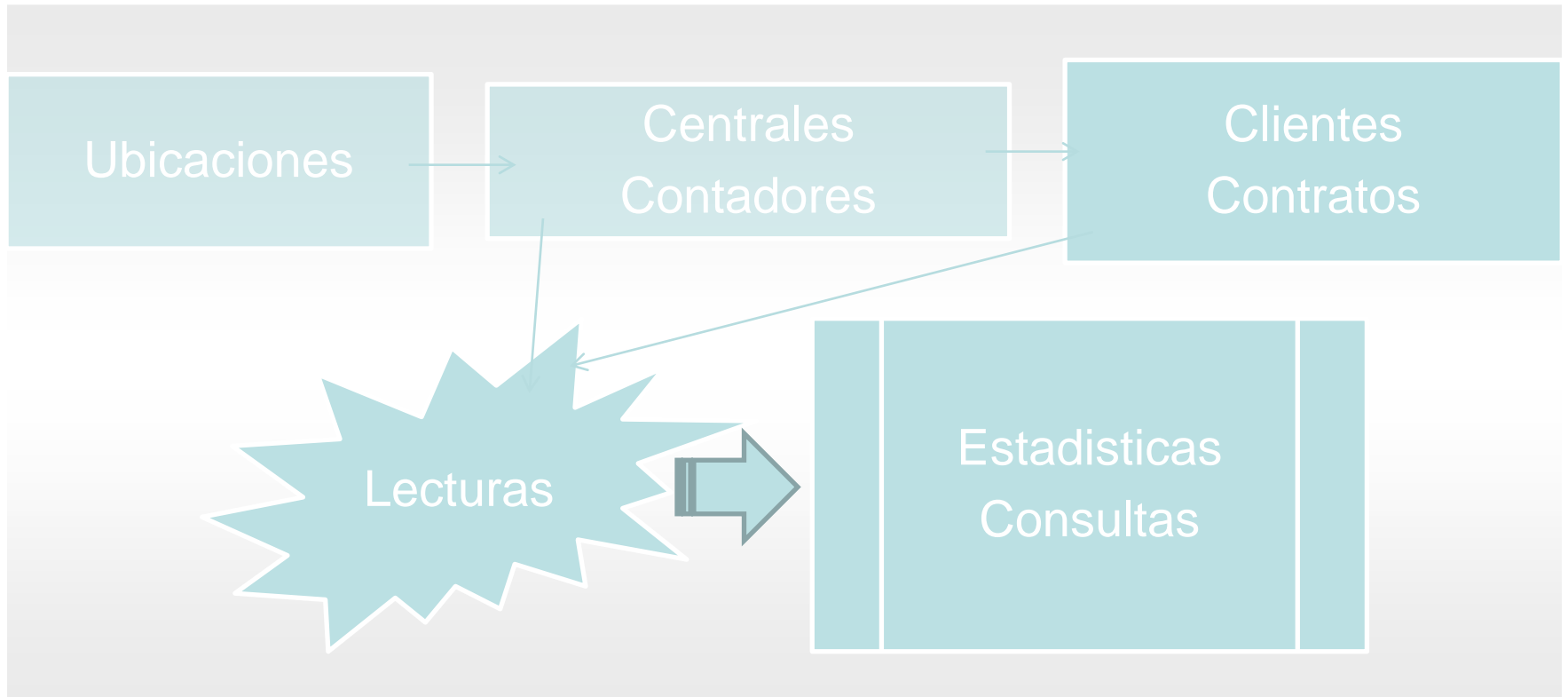
Excepciones

- ✓ Se inserta una entrada en la tabla de Logs.
- ✓ Todos los procedimientos, tienen control de excepciones, sino se recoge específicamente, se graba la que genera el propio SBGB.
- ✓ Salida cadena explicativa del error producido, en función de la excepción capturada.
- ✓ En el script de testing, se muestran en detalle de cada procedimiento



```
---modificaEstados
declare
  rst varchar2(250);
begin
  --INPUT: n_id,n_des
  modificaEstados(1,'ALTA', rst);
end;
/* exceptions
  rst := 'ERROR: No existe el id del Estado';
  rst := 'ERROR: Ya existe este tipo de ESTADOS';
  rst := 'ERROR: el tipo de ESTADOS no debe ser nulo';
*/
-----
```

Gestión



Pruebas

ORDEN:

1. La BD tiene restricciones de integridad referencial
2. Para que unos procedimientos funcionen, otros deber ser llamados antes.
 - Ejemplo: No se pueden dar de alta contratos sino hay un cliente y contador dado de alta.
 - No se pueden dar de alta ubicaciones, sino estan dadas de alta vias, paises, provincias y localidades

- ✓ Hay que realizar la carga de datos antes de comenzar las pruebas
- ✓ Tratan de ser exhaustivas, se busca provocar todos los errores o posibles situaciones.
- ✓ Contemplan las situaciones más importantes e interesantes.
- ✓ Se realizan ejecutando un script
- ✓ La ejecución da el resultado esperado según los requisitos.

Conclusiones

- ✓ Se han alcanzado los objetivos planteados en el enunciado.
- ✓ He adquirido destreza y conocimientos en el desarrollo del proyecto.
- ✓ Se ha logrado eficiencia, robustez y excelencia.
- ✓ Se han automatizado procesos:
 - ✓ Al dar de alta un contrato preparamos los contadores, y los consumos.
 - ✓ Al dar de alta la lectura de contador, actualizamos el histórico de lecturas, los consumos y las tablas de estadísticas.
- ✓ Se utiliza una implementación encadenada, con mejoras añadidas.
- ✓ Se ha conseguido una aplicación modular, con estructura reutilizable para otras aplicaciones. Ej. La gestión de las ubicaciones.
- ❖ En resumen un duro trabajo que ha servido para probar la capacidad de generar un proyecto de cierta complejidad, haciendo uso de los conocimientos adquiridos a lo largo de los estudios carrera en la UOC.