



Universitat Oberta
de Catalunya

www.uoc.edu

TRABAJO FINAL DE CARRERA

BASES DE DATOS RELACIONALES

INGENIERÍA TÉCNICA INFORMÁTICA DE SISTEMAS
JUNIO 2014

Autor: Bernardo Pons Bibiloni
Consultor: Manel Rella Ruiz

CONTENIDO



OBJETIVOS

METODOLOGÍA UTILIZADA: CASCADA

PLANIFICACIÓN

DISEÑO LÓGICO: DIAGRAMA ENTIDAD-RELACIÓN

IMPLEMENTACIÓN:

MODELO RELACIONAL

PROCEDIMIENTOS PL/SQL

MÓDULO ABM DE PARLAMENTOS, PARLAMENTARIOS Y GASTOS

MÓDULO ESTADÍSTICA

PRODUCTO OBTENIDO

VALORACIÓN ECONÓMICA

CONCLUSIONES

OBJETIVOS



- ❖ Ejercicio práctico de síntesis de los conocimientos adquiridos en la carrera.
- ❖ Proyecto de diseño, implementación, pruebas y documentación de una base de datos relacional conforme al análisis de requerimientos suministrado.
- ❖ Implementación de la base de datos de un sistema de control del gasto público en los parlamentos europeos.
- ❖ Utilización de *ORACLE SQL Express*, *ORACLE SQL Developer* y programación en *PL/SQL*

METODOLOGÍA UTILIZADA: CASCADA



Análisis de requerimientos

- Básicamente coincide con el contenido del enunciado.
- Eliminación de posibles dudas con el cliente (consultor).



Diseño lógico

- Identificación de entidades y relaciones entre entidades.
- Elaboración del diagrama Entidad - Relación.



Implementación

- Diseño tablas de la base de datos relacional.
- Programación procedimientos *PL/SQL* (módulos ABM, listados y estadísticas).



Pruebas

- Carga de datos en la base de datos mediante scripts en tablas auxiliares.
- Carga de datos mediante ejecución de programas en *PL/SQL* y los módulos ABM.
- Prueba módulos de listados y estadísticas mediante programas en *PL/SQL*.



Documentación

- Elaboración de la memoria.
- Elaboración de la presentación .



Implantación y mantenimiento

- No se llevará a cabo por no formar parte del TFC.

PLANIFICACIÓN

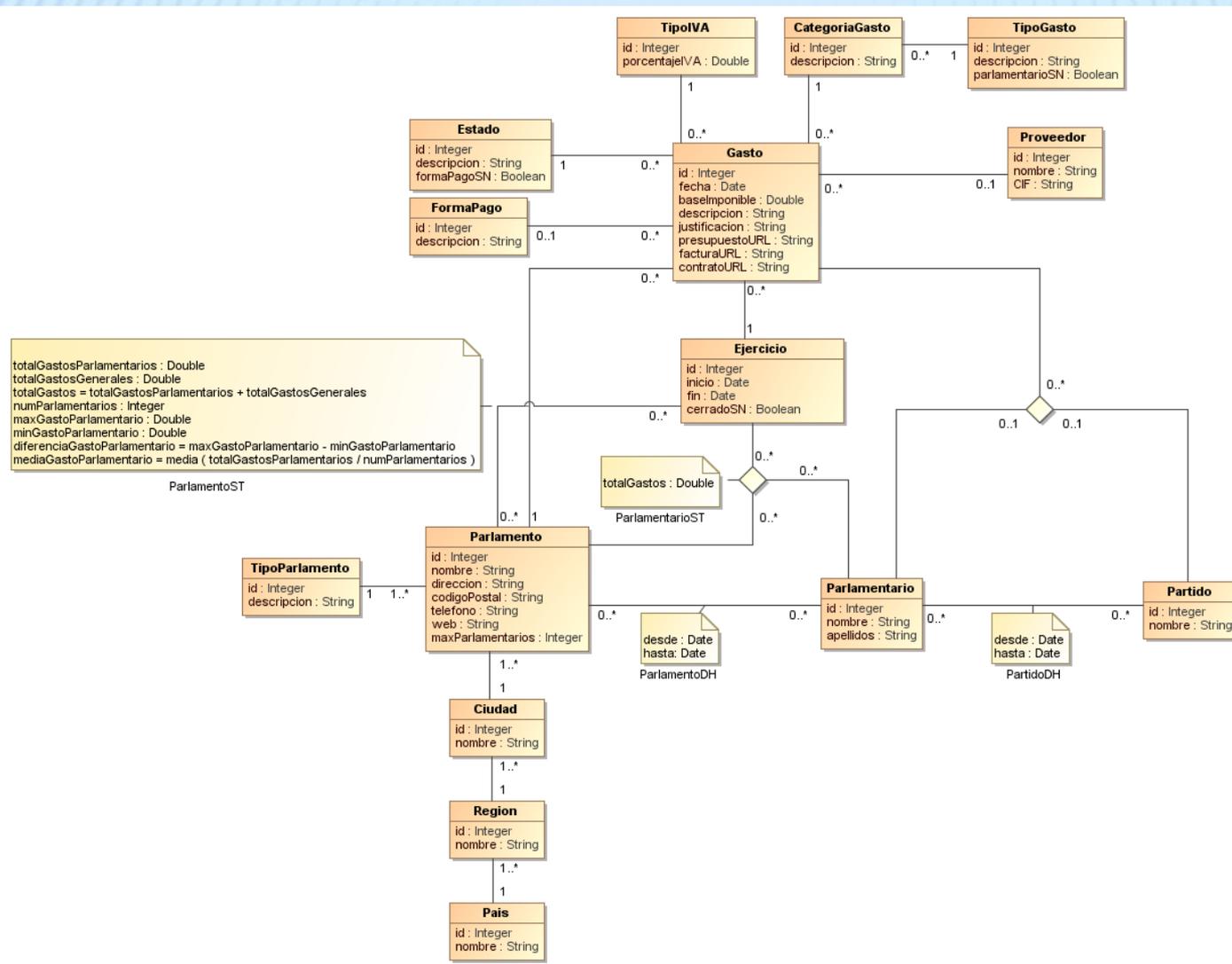


Nombre de tarea	Duración	Comienzo	Fin
Trabajo final de carrera	97 días	mié 09/10/13	mar 14/01/14
Elaboración del plan de trabajo	12 días	mié 09/10/13	lun 21/10/13
Lectura del enunciado del trabajo	1 día	mié 09/10/13	jue 10/10/13
Identificación de tareas a realizar y estimación tiempos	3 días	jue 10/10/13	dom 13/10/13
Descarga e instalación de <i>Microsoft Project</i>	3 días	dom 13/10/13	mié 16/10/13
Elaboración diagrama de <i>Gantt</i>	3 días	mié 16/10/13	sáb 19/10/13
Redacción documento plan de trabajo	2 días	sáb 19/10/13	lun 21/10/13
Entrega PAC1	0 días	lun 21/10/13	lun 21/10/13
Análisis y diseño	22 días	lun 21/10/13	mar 12/11/13
Identificación de entidades y relaciones	6 días	lun 21/10/13	dom 27/10/13
Descarga e instalación de <i>Magic Draw</i>	3 días	dom 27/10/13	mié 30/10/13
Elaboración diagrama Entidad-Relación	7 días	mié 30/10/13	mié 06/11/13
Redacción documento PAC2	6 días	mié 06/11/13	mar 12/11/13
Entrega PAC2	0 días	mar 12/11/13	mar 12/11/13
Implementación y pruebas	30 días	mar 12/11/13	jue 12/12/13
Preparación entorno de trabajo	5 días	mar 12/11/13	dom 17/11/13
Creación máquina virtual + instalación S.O.	1 día	mar 12/11/13	mié 13/11/13
Descarga e instalación de <i>Oracle</i>	4 días	mié 13/11/13	dom 17/11/13
Programación SQL	16 días	dom 17/11/13	mar 03/12/13
Creación base de datos	1 día	dom 17/11/13	lun 18/11/13
Creación de tablas	1 día	lun 18/11/13	mar 19/11/13
Programación <i>PL/SQL</i>	7 días	mar 19/11/13	mar 26/11/13
Programación consultas <i>SQL</i>	7 días	mar 26/11/13	mar 03/12/13
Pruebas	4 días	mar 03/12/13	sáb 07/12/13
Programación y ejecución scripts carga de datos	2 días	mar 03/12/13	jue 05/12/13
Programación y ejecución scripts juegos de pruebas	2 días	jue 05/12/13	sáb 07/12/13
Redacción documento	5 días	sáb 07/12/13	jue 12/12/13
Redacción documento PAC3	5 días	sáb 07/12/13	jue 12/12/13
Entrega PAC3	0 días	jue 12/12/13	jue 12/12/13
Elaboración documentación final a entregar	33 días	jue 12/12/13	mar 14/01/14
Elaboración memoria del proyecto	21 días	jue 12/12/13	jue 02/01/14
Elaboración presentación proyecto	12 días	jue 02/01/14	mar 14/01/14
Entrega documentación final	0 días	mar 14/01/14	mar 14/01/14

“Ningún plan, por bien trazado que esté, sobrevive al primer encuentro con el enemigo”.
Proverbio militar anónimo

- ❖ El proyecto ha sufrido un retraso por diversos motivos que han obligado a entregar el TFC en el semestre siguiente.
- ❖ La curva de aprendizaje de *ORACLE SQL, SQL Developer* y *PL/SQL* ha sido más larga de lo previsto. La peculiar implementación *SQL* de *ORACLE* ha provocado retrasos.
- ❖ *SQL Developer* es una herramienta *CASE* cuya funcionalidad y documentación son insuficientes.

DISEÑO LÓGICO: DIAGRAMA ENTIDAD-RELACIÓN



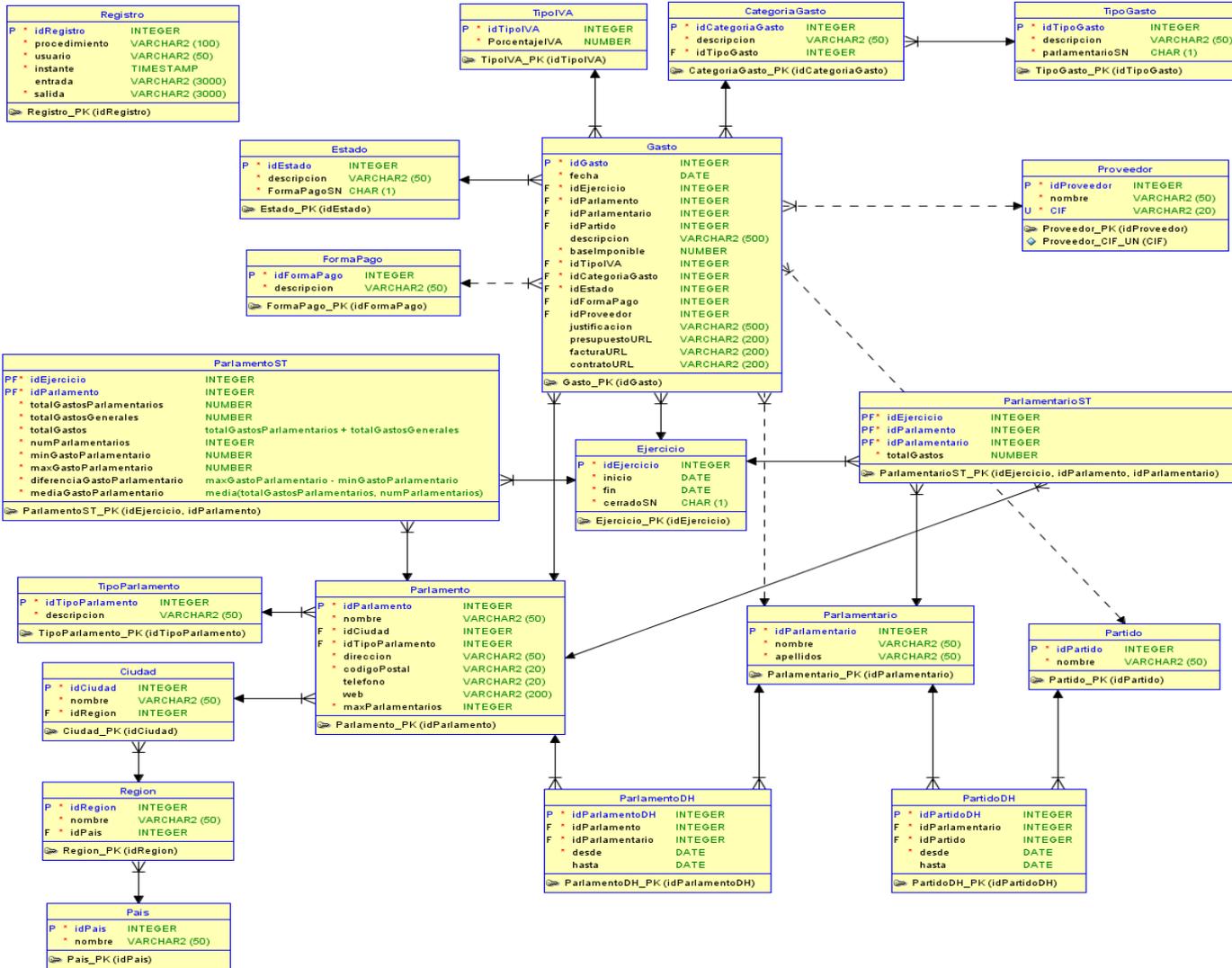
15 Entidades

- ❖ Parlamentario
- ❖ Parlamento
- ❖ Partido
- ❖ TipoParlamento
- ❖ Ejercicio
- ❖ Gasto
- ❖ Estado
- ❖ CategoriaGasto
- ❖ TipoGasto
- ❖ FormaPago
- ❖ TipoIVA
- ❖ Proveedor
- ❖ Ciudad
- ❖ Region
- ❖ Pais

17 Relaciones

- ❖ 15 binarias
- ❖ 2 ternarias

MODELO RELACIONAL

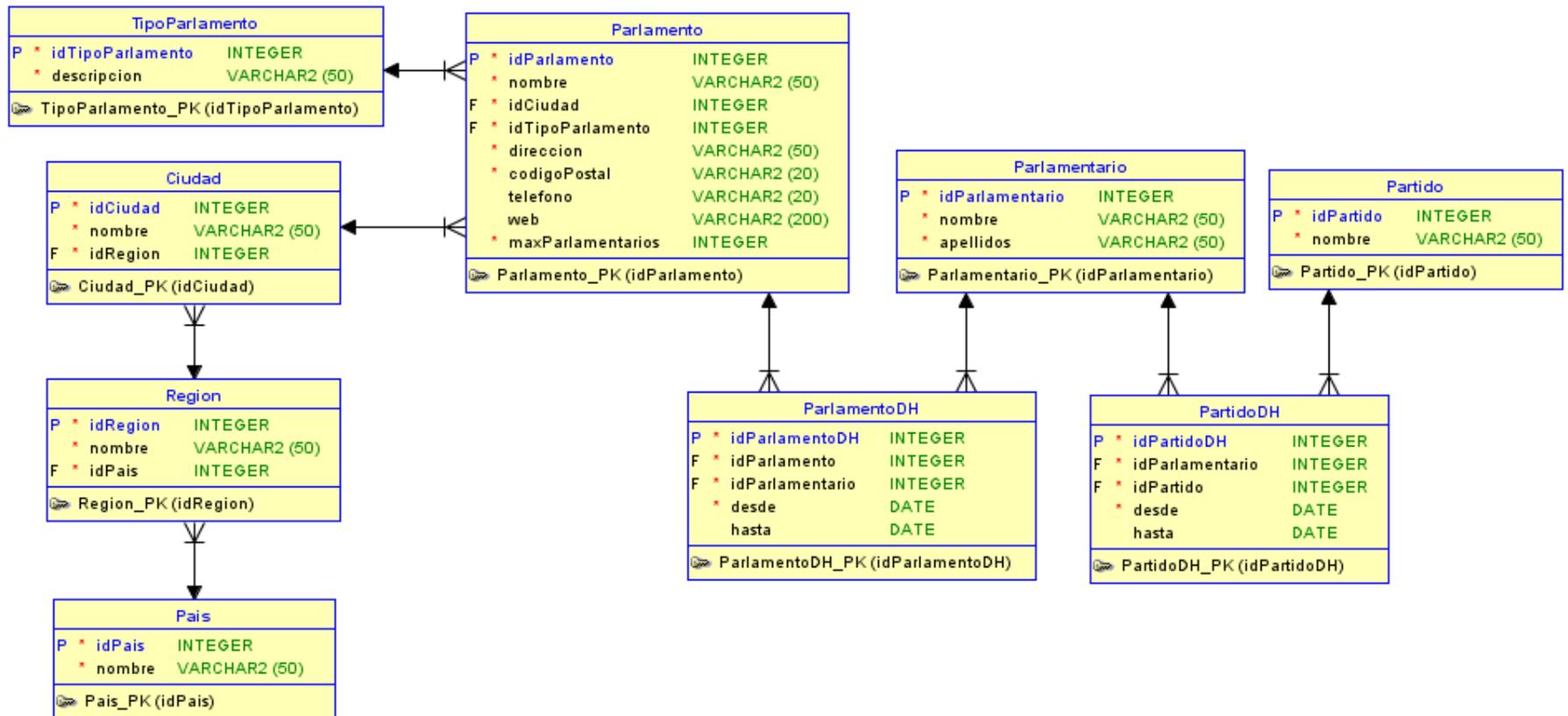


20 Tablas

- ❖ Parlamentario
- ❖ Parlamento
- ❖ Partido
- ❖ TipoParlamento
- ❖ Ejercicio
- ❖ Gasto
- ❖ Estado
- ❖ CategoriaGasto
- ❖ TipoGasto
- ❖ FormaPago
- ❖ TipoIVA
- ❖ Proveedor
- ❖ Ciudad
- ❖ Region
- ❖ Pais
- ❖ PartidoDH
- ❖ ParlamentoDH
- ❖ ParlamentoST
- ❖ ParlamentarioST
- ❖ Registro

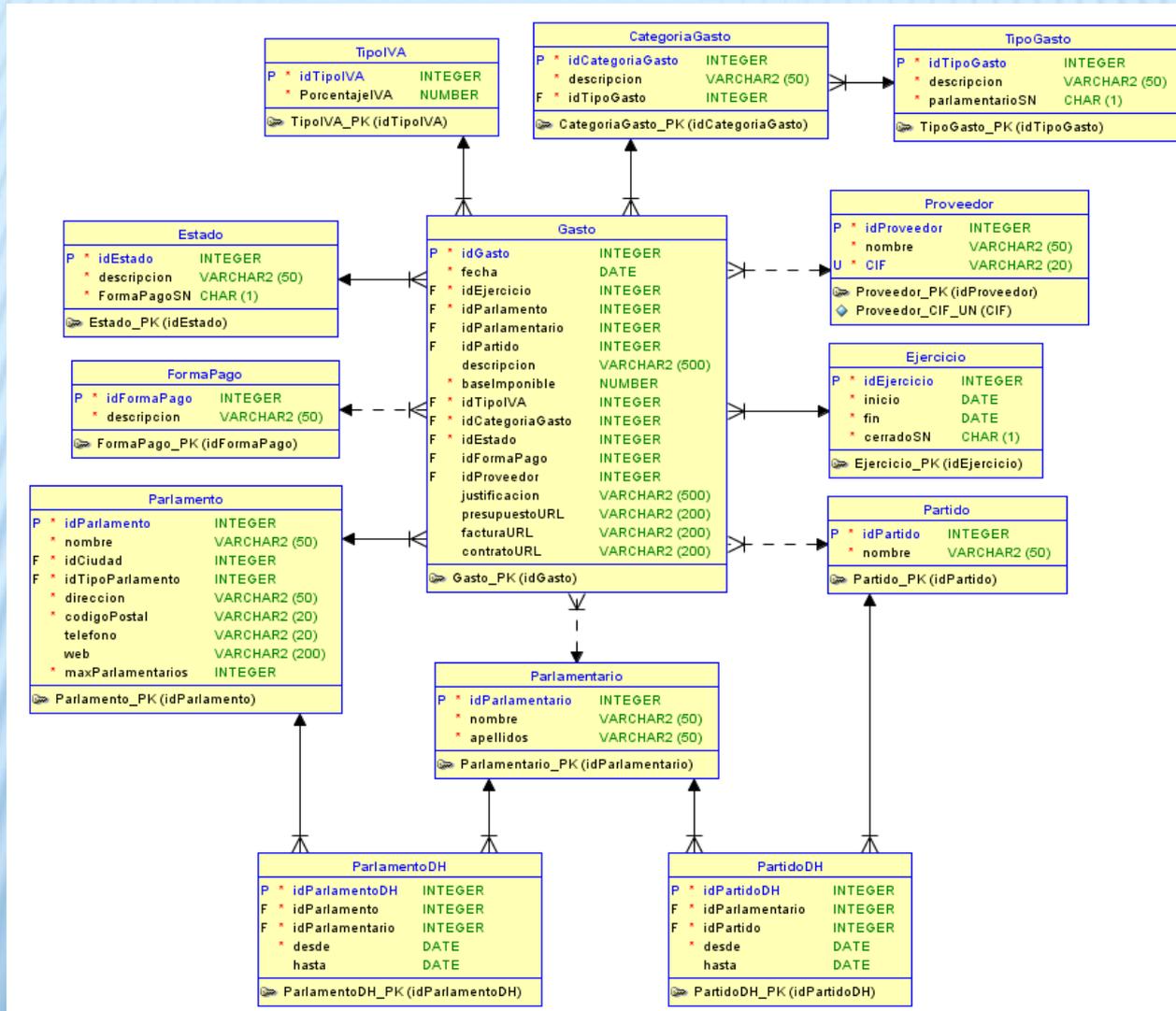


DETALLE MODELO RELACIONAL (1/3)



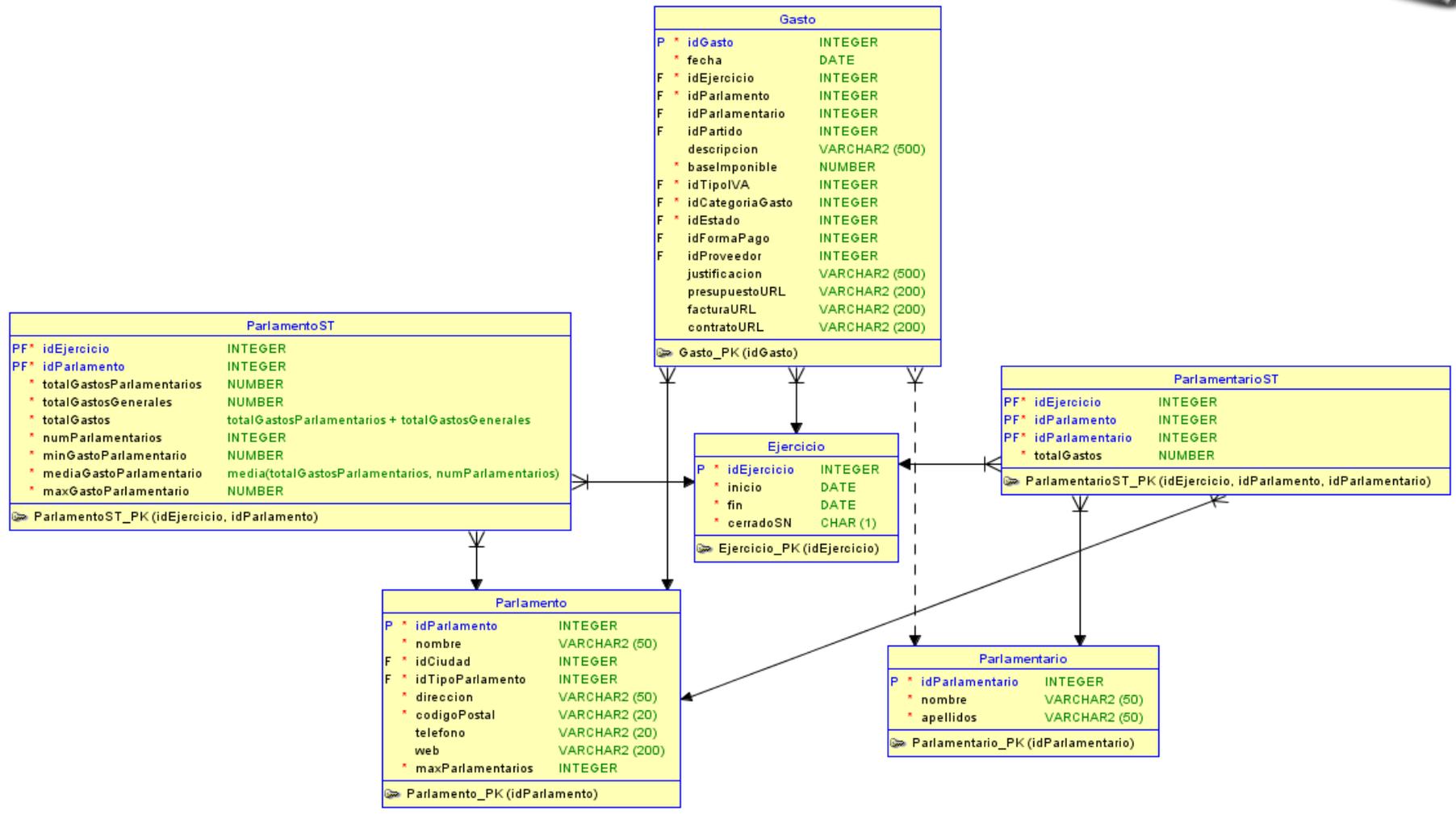


DETALLE MODELO RELACIONAL (2/3)





DETALLE MODELO RELACIONAL (3/3)



PROCEDIMIENTOS PL/SQL



Paquetes

- La utilización de paquetes introduce la modularidad en el diseño que a su vez favorece la organización y reduce la complejidad. La creación de paquetes también favorece la reutilización de código y permite cierto grado de encapsulamiento, es decir, ocultación de los detalles de la implementación a los potenciales usuarios del paquete reduciendo la probabilidad de la aparición de errores al utilizarlo.
- **pkg_comun:** Paquete de procedimientos comunes de utilidad.
- **pkg_parlamento:** Paquete de procedimientos de altas, bajas y modificaciones en la tabla Parlamento.
- **pkg_parlamentario:** Paquete de procedimientos de altas, bajas y modificaciones en la tabla Parlamentario y tablas relacionadas ParlamentoDH y PartidoDH.
- **pkg_gasto:** Paquete de procedimientos de altas, bajas y modificaciones en la tabla Gasto.
- **pkg_listado:** Paquete de procedimientos que realizan los listados solicitados.
- **pkg_estadistica:** Paquete de procedimientos que calculan las estadísticas solicitadas.

Polimorfismo

- Se ha utilizado el polimorfismo en la implementación de algunos procedimientos
- Mediante el uso del polimorfismo, diversos procedimientos que comparten un mismo nombre, pueden tener implementaciones distintas de la misma tarea, en función de que reciban un número distinto de parámetros o parámetros de distinto tipo.
- Por ejemplo en el paquete pkg_comun existen tres implementaciones del procedimiento existe_parlamentarioST:
 - **existe_parlamentarioST (idEjercicio, idParlamento, idParlamentario)**
 - **existe_parlamentarioST (idEjercicio, idParlamentario)**
 - **existe_parlamentarioST (idEjercicio)**
- De no existir la posibilidad de utilizar polimorfismo se tendría que haber utilizado nombres distintos para procedimientos que realizan la misma tarea.

PROCEDIMIENTOS PL/SQL (ESTRUCTURA)



Características

```
PROCEDURE <nombre_procedimiento> (  
    <lista_parametros_entrada> IN <tipo_parametro_entrada>,  
    <lista_parametros_salida> OUT <tipo_parametro_salida>)  
AS  
BEGIN  
    p_rsp := 'OK';  
    BEGIN  
        .  
        .  
        .  
        <codigo_del_procedimiento>  
        .  
        .  
        .  
    EXCEPTION  
    WHEN <identificador_excepcion_tratada_1>  
        p_rsp := 'ERROR: <texto_error_excepcion_1> [' || <valor_parametro> || '] <continuacion_texto_error>';  
    WHEN <identificador_excepcion_tratada_2>  
        p_rsp := 'ERROR: <texto_error_excepcion_2> [' || <valor_parametro> || '] <continuacion_texto_error>';  
        .  
        .  
        .  
    WHEN <identificador_excepcion_tratada_n>  
        p_rsp := 'ERROR: <texto_error_excepcion_n> [' || <valor_parametro> || '] <texto_error>';  
    WHEN OTHERS THEN  
        p_rsp := 'ERROR: ' || SQLERRM;  
    END;  
    entrada := ' <literal_parametro_entrada> = [' || <valor_parametro_entrada> || ']';  
    salida := ' <literal_parametro_salida> = [' || <valor_parametro_salida> || '] + rsp = ' || <valor_parametro_rsp>;  
    pkg_comun.entrada_registro ( '<nombre_paquete>.<nombre_procedimiento>', USER, SYSTIMESTAMP, entrada, salida ) ;  
EXCEPTION  
WHEN OTHERS THEN  
    DBMS_OUTPUT.PUT_LINE ( 'ERROR INSERTANDO ENTRADA EN REGISTRO: ' || SQLERRM ) ;  
END <nombre_procedimiento>;
```

- ❖ Se utiliza una estructura interna común en los procedimientos compuesta por dos bloques anidados para el tratamiento de excepciones.
- ❖ El bloque externo inicializa al principio el indicador de error y al final construye los *strings* entrada y salida antes de registrar el resultado de la ejecución del procedimiento.
- ❖ El bloque interno realiza las tareas propias del procedimiento y gestiona las posibles excepciones modificando el indicador de error si se captura alguna excepción.
- ❖ Obtenemos así procedimientos más compactos y sin repetición de código.

MÓDULO ABM

MEJORAS REALIZADAS A LOS REQUERIMIENTOS



Generalización

- La solución implementada mediante las tablas ParlamentoDH (que relaciona las tablas Parlamento y Parlamentario) y PartidoDH (que relaciona las tablas Partido y Parlamentario) es una generalización del modelo de datos solicitado.
- Permite controlar los diversos períodos que un parlamentario está en un parlamento y en un partido.
- Un parlamentario puede pasar períodos en distintos parlamentos. Puede estar incluso en el mismo parlamento en varios períodos distintos. Pero en una fecha determinada sólo puede estar en un Parlamento como máximo.
- Un parlamentario puede pasar períodos en distintos partidos. Puede estar incluso en el mismo partido en varios períodos distintos. Pero en una fecha determinada sólo puede estar en un Partido como máximo.

Tabla ParlamentoDH

- **Períodos de un Parlamentario en un Parlamento.**
- **Clave primaria:**
 - Secuencia idParlamentoDH.
- **idParlamento:** Identificador del parlamento.
- **idParlamentario:** Identificador del parlamentario.
- **desde:** Fecha inicial del período. No puede ser nula y debe ser posterior a cualquier fecha de períodos anteriores de este parlamentario en ese u otros parlamentos.
- **hasta:** Fecha final del período. Puede ser nula para indicar que ese período no ha finalizado todavía. Si no es nula debe ser posterior a la fecha inicial del período correspondiente.

Tabla PartidoDH

- **Períodos de un Parlamentario en un Partido**
- **Clave primaria:**
 - Secuencia idPartidoDH
- **idPartido:** Identificador del partido.
- **idParlamentario:** Identificador del parlamentario.
- **desde:** Fecha inicial del período. No puede ser nula y debe ser posterior a cualquier fecha de períodos anteriores de este parlamentario en ese u otros partidos.
- **hasta:** Fecha final del período. Puede ser nula para indicar que ese período no ha finalizado todavía. Si no es nula debe ser posterior a la fecha inicial del período correspondiente.

MÓDULO ESTADÍSTICA

MEJORAS REALIZADAS A LOS REQUERIMIENTOS



Generalización

- La solución implementada mediante las tablas ParlamentoST y ParlamentarioST es una generalización de las estadísticas solicitadas.
- Esta modificación implica que **las consultas estadísticas no son siempre estrictamente en tiempo constante 1** pero se realizan siempre sobre un subconjunto reducido de filas y por tanto no se penaliza el tiempo de cálculo de las estadísticas de forma significativa.
- La mayor versatilidad de esta implementación justifica sobradamente la modificación realizada a los requerimientos.
- Entre otras mejoras permite la obtención de estadísticas sobre un número arbitrario de ejercicios y no sólo sobre los últimos cuatro.
- **Esta modificación fue consensuada con el consultor.**

Tabla ParlamentoST

- **Clave primaria:**
 - idEjercicio
 - idParlamento
- **Estadísticas almacenadas:**
 - totalGastosParlamentarios
 - totalGastosGenerales
 - numParlamentarios
 - minGastoParlamentario
 - maxGastoParlamentario
- **Estadísticas calculadas:**
 - totalGastos
 - mediaGastoParlamentario
- **Permite obtener múltiples estadísticas fijando:**
 - (idEjercicio, idParlamento)
 - (idEjercicio)
 - (idParlamento)

Tabla ParlamentarioST

- **Clave primaria:**
 - idEjercicio
 - idParlamento
 - idParlamentario
- **Estadísticas almacenadas:**
 - totalGastos
- **Permite obtener múltiples estadísticas fijando:**
 - (idEjercicio, idParlamento, idParlamentario).
 - (idEjercicio, idParlamento)
 - (idEjercicio, idParlamentario)
 - (idParlamento, idParlamentario)
 - (idEjercicio)
 - (idParlamento)
 - (idParlamentario)

PRODUCTO OBTENIDO

COMPONENTES



PRODUCTO

- Base de datos *ORACLE*.
- Scripts de creación tablas.
- Scripts de creación de paquetes *PL/SQL*.
- Scripts de carga de datos en tablas auxiliares.
- Scripts de pruebas del módulo ABM.
- Scripts de pruebas del módulo listados.
- Scripts de pruebas del módulo estadísticas.

DOCUMENTACIÓN

- Memoria del proyecto.
 - Detalla todo el trabajo realizado.
- Anexo
 - Resultados de los juegos de pruebas.
- Presentación
 - Resumen en diapositivas del trabajo realizado.

VALORACIÓN ECONÓMICA

ESTIMACIÓN DEL PRESUPUESTO DEL PROYECTO



Tareas	Categoría profesional	Horas	Precio	Coste
Análisis, coordinación, documentación	Jefe de proyecto	52	60,00 €	3.120,00 €
Análisis y diseño lógico, creación base de datos	Analista	24	60,00 €	1.440,00 €
Programación PL/SQL, documentación interna	Programador	55	45,00 €	2.475,00 €
Pruebas	Testing	30	30,00 €	900,00 €
Documentación	Documentador	32	30,00 €	960,00 €
	Totales	193		8.895,00 €

CONCLUSIONES

VALORACIÓN



Como aspectos positivos cabe destacar:

- Se ha desarrollado una base de datos que mejora los requerimientos solicitados en el enunciado.
- Se han puesto en práctica los conocimientos adquiridos a lo largo de la carrera en asignaturas como programación estructurada, programación orientada a objetos, estructuras de información y, en especial, en las asignaturas de bases de datos.
- Se han adquirido nuevos conocimientos y utilizado herramientas que no había usado hasta ahora, acumulando experiencia y conocimientos que son aplicables en el mundo laboral

Como aspectos negativos cabe mencionar:

- El trabajo no pudo realizarse conforme a la planificación prevista inicialmente y ha tenido que finalizarse en el semestre siguiente.
- Problemas personales provocaron un retraso en el inicio que se propagó a lo largo del las etapas del proyecto.
- La curva de aprendizaje de la peculiar implementación de *SQL ORACLE*, así como la falta de funcionalidades e insuficiente documentación de la herramienta *CASE SQL Developer* también supusieron una serie de contratiempos que afectaron a la planificación de forma importante.



Universitat Oberta
de Catalunya

www.uoc.edu

GRACIAS POR SU ATENCIÓN
