

# Diseño e implementación de una base de datos para la gestión de regalos en grupo de una red social.



## Dedicatoria y agradecimientos

Le dedico este proyecto a la persona más importante en mi vida, mi mujer.

Gracias por el esfuerzo realizado durante estos últimos 4 años que para hacerlo más difícil, han coincidido con el nacimiento de nuestra primera hija. Has sabido educarla de manera excepcional, teniendo en cuenta que no podía ayudarte mucho por la falta de tiempo.

Sin tu esfuerzo no hubiera tenido el tiempo suficiente para acabar la carrera.

Gracias.

## Resumen

El trabajo final de carrera (TFC) es una asignatura que está pensada para realizar un trabajo de síntesis con los conocimientos adquiridos a lo largo del plan de estudios.

A la hora de realizar un proyecto se debe escoger una especialidad de las muchas temáticas que ofrece la informática. Concretamente este proyecto se centra en las bases de datos relacionales.

A partir de un enunciado, el alumno deberá: analizar los requerimientos, diseñar la base de datos, implementar los procedimientos con la lógica del programa y garantizar el correcto funcionamiento con un plan de pruebas.

Durante la construcción del proyecto se van realizando entregas de documentación tal y como exige la evaluación continuada de la asignatura.

Con estas entregas el alumno también aprende la forma correcta de gestionar un proyecto, documentando las fases más habituales (plan de trabajo, análisis e implementación) para terminar con la memoria del proyecto.

Palabras clave: Bases de datos, Oracle, pl/sql.

## Índice de contenido.

1.- Descripción del proyecto. ....	8
1.1.- Requisitos funcionales del sistema. ....	9
1.2.-Requerimientos de la metodología.....	12
1.3.- Responsabilidad/seguridad en el uso de los procedimientos. ....	12
1.4.- Metodología del desarrollo del proyecto. ....	12
1.5.- Planificación. ....	13
1.5.1.- Relación de actividades y estimación del tiempo. ....	13
1.5.2.- Diagrama de Gantt.....	15
1.5.3.- Productos obtenidos.....	16
1.5.4-Breve descripción del resto de capítulos. ....	17
2.- Análisis y diseño. ....	18
2.1.- Revisión de los requerimientos.....	18
2.1.1.- Procedimientos incoherentes. ....	18
2.1.2.- Mejoras aplicadas. ....	19
2.2.- Diseño conceptual.....	21
2.2.1.- Diagrama de casos de uso.....	21
2.2.2.- Diagrama entidad relación.....	22
2.2.2.1.- Entidades.....	23
2.3.- Diseño lógico.....	29
2.3.- Diseño físico. ....	32
2.3.1.- Disminuir el tiempo de respuesta. ....	32
2.3.2.- Minimizar el espacio de almacenamiento. ....	33
2.3.3.- Integridad de los datos. ....	33
3.- Implementación. ....	34
3.1.- Procedimientos. ....	34
3.1.1.- Homenajeado.....	35
3.1.2.- Amigos.....	40
3.1.3.- Administrador. ....	50
3.1.4.- Proveedor.....	52
3.1.5.- Empresa transportes.....	53
3.1.6.- Procesos Batch.....	54
4.- Plan de contingencias. ....	55
5.- Plan de pruebas.....	56

6.- Módulo estadístico.....	56
6.1.- Análisis y diseño .....	56
6.1.1.- Diagrama entidad relación.....	57
6.1.1.1.- Entidades.....	57
6.1.2.- Diseño lógico .....	58
6.2.- Procedimientos. ....	59
6.3.- Plan de pruebas.....	65
7.- Valoración económica y recursos utilizados. ....	66
7.1.- Recursos humanos. ....	66
7.2.- Recursos técnicos y materiales. ....	66
7.3.- Coste del proyecto. ....	67
8.- Mejoras futuras.....	69
8.1.- Cálculo de los gastos de envío. ....	69
8.2.- Envío de notificaciones de productos afines. ....	69
9.- Conclusiones. ....	70
10.- Glosario. ....	70
11.- Bibliografía. ....	71
12.- Anexos.....	72
12.1- Creación de la base de datos. ....	72
12.2.- Procedimientos con doble interfaz. ....	73
12.3.- Procedimientos de pruebas del sistema.....	77
12.3.1.- Procedimientos de pruebas del módulo estadístico.....	78

## Índice de tablas.

Tabla 1, relación de actividades y tiempo.....	14
Tabla 2, fechas de entrega documentación/producto.....	17
Tabla 3, entidad Proveedores. ....	24
Tabla 4, entidad Productos. ....	24
Tabla 5, entidad categorías. ....	24
Tabla 6, entidad tipos_iva. ....	24
Tabla 7, entidad Países.....	24
Tabla 8, entidad precio_productos. ....	24
Tabla 9, entidad Usuarios.....	25
Tabla 10, entidad Mes.....	25
Tabla 11, entidad Grupo_compras.....	25
Tabla 12, entidad Participantes.....	26

Tabla 13, entidad Lista_deseos.....	26
Tabla 14, entidad Regalos.....	26
Tabla 15, entidad Estados_regalos.....	26
Tabla 16, entidad Automata_estados.....	27
Tabla 17, entidad Tipos_celebraciones.....	27
Tabla 18, entidad Provincias.....	27
Tabla 19, entidad Municipios.....	27
Tabla 20, entidad Vías.....	27
Tabla 21, entidad Ubicaciones.....	28
Tabla 22, entidad Envíos.....	28
Tabla 23, entidad Pagos.....	28
Tabla 24, entidad Logs.....	29
Tabla 25, entidad Parámetros.....	29
Tabla 26, Procedimiento: pkg_homenajeado.ListadoCategorias.....	36
Tabla 27, Procedimiento: pkg_homenajeado.ListadoProductos.....	36
Tabla 28, Procedimiento: pkg_homenajeado.AltaDeseo.....	37
Tabla 29, Procedimiento: pkg_homenajeado.ListadoDeseos.....	38
Tabla 30, Procedimiento: pkg_homenajeado.ModificaDeseo.....	39
Tabla 31, Procedimiento: pkg_homenajeado.BajaDeseo.....	39
Tabla 32, Procedimiento: pkg_homenajeado.ListadoRegalosRecibidos.....	40
Tabla 33, Procedimiento: pkg_amigos. AltaGrupo.....	42
Tabla 34, Procedimiento: pkg_amigos.ListadoGrupos.....	42
Tabla 35, Procedimiento: pkg_amigos.BajaGrupo.....	43
Tabla 36, Procedimiento: pkg_amigos.AltaParticipante.....	43
Tabla 37, Procedimiento: pkg_amigos.BajaParticipante.....	44
Tabla 38, Procedimiento: pkg_amigos.AltaRegalo.....	45
Tabla 39, Procedimiento: pkg_amigos.ListadoRegalosGrupo.....	46
Tabla 40, Procedimiento: pkg_amigos.BajaRegalo.....	46
Tabla 41, Procedimiento: pkg_amigos.AltaPago.....	47
Tabla 42, Procedimiento: pkg_amigos.ListadoPagos.....	48
Tabla 43, Procedimiento: pkg_amigos.ModificaPago.....	49
Tabla 44, Procedimiento: pkg_amigos.BajaPago.....	50
Tabla 45, Procedimiento: pkg_amigos.ListadoParticipantesGrupo.....	50
Tabla 46, Procedimiento: pkg_administrador.ListadoTopTenRegalosCiudades.....	51
Tabla 47, Procedimiento: pkg_administrador.ListadoUsuariosSuperanMedia.....	51
Tabla 48, Procedimiento: pkg_proveedor.ProcesarPedido.....	53
Tabla 49, Procedimiento: pkg_proveedor.InformaEnvio.....	53
Tabla 50, Procedimiento: pkg_repartidor.InformarEntrega.....	54
Tabla 51, Procedimiento: pkg_procesos_batch.BajaGruposFechaLimite.....	55
Tabla 52, plan de contingencias.....	55
Tabla 53, entidad Ejercicios.....	58
Tabla 54, entidad Productos_ejercicios.....	58
<b>Tabla 55, entidad Ejercicio_meses.....</b>	<b>58</b>
Tabla 56, Procedimiento: pkg_administrador. E1_num_productos_año.....	60

Tabla 57, Procedimiento: pkg_administrador.E2_num_regalos.....	60
Tabla 58, Procedimiento: pkg_administrador.E3_importe_total_regalos. ....	61
Tabla 59, Procedimiento: pkg_administrador.E4_porcentaje_completado.....	61
Tabla 60, Procedimiento: pkg_administrador.E5_media_gasto_regalos. ....	62
Tabla 61, Procedimiento: pkg_administrador.E6_usuario_mas_regalos. ....	62
Tabla 62, Procedimiento: pkg_administrador.E7_usuario_mas_gasto. ....	63
Tabla 63, Procedimiento: pkg_administrador.E8_usuario_mas_participativo.....	64
Tabla 64, Procedimiento: pkg_administrador.E9_grupo_compra_mas_gasto. ....	65
Tabla 65, Procedimiento: pkg_administrador.ListadoUsuariosSuperanMedia. ....	65
Tabla 66, relación de tareas/tiempo/recurso. ....	68
Tabla 67, resumen de horas por técnicos. ....	68
Tabla 68, coste total del proyecto.....	68
Tabla 69, Procedimiento: pkg_homenajeado.ListadoCategorias. ....	73
Tabla 70, Procedimiento: pkg_homenajeado.ListadoProductos. ....	74
Tabla 71, Procedimiento: pkg_homenajeado.ListadoDeseos.....	74
Tabla 72, Procedimiento: pkg_amigos. ListadoGrupos.....	75
Tabla 73, Procedimiento: pkg_amigos.ListadoRegalosGrupo.....	75
Tabla 74, Procedimiento: pkg_amigos.ListadoPagos.....	76
Tabla 75, Procedimiento: pkg_amigos.ListadoParticipantesGrupo.....	77
Tabla 76, Procedimiento: pkg_test.Elimina_Datos.....	77
Tabla 77, Procedimiento: pkg_test.Pruebas_Unitarias.....	77
Tabla 78, Procedimiento: pkg_test.Pruebas_Funcionales.....	78
Tabla 79, Procedimineto: pkg_test.Pruebas_Estadisticas. ....	78
Tabla 80, Procedimiento: pkg_test.Generador_datos_estadisticos.....	80

## Índice de imágenes.

Imagen 1, fases del ciclo de vida.....	13
Imagen 2, plan de trabajo. ....	15
Imagen 3, análisis y diseño.....	15
Imagen 4, implementación y pruebas.....	16
Imagen 5, entrega producto. ....	16
Imagen 6, diagrama de casos de uso. ....	21
Imagen 7, diagrama entidad relación. ....	23
Imagen 8, diseño lógico.....	30
Imagen 9, distribución procedimientos. ....	34
Imagen 10, Diagrama entidad relación del módulo estadístico. ....	57
Imagen 11, diseño lógico del módulo estadístico. ....	58
Imagen 12, Configuración correcta del sql developer. ....	72
Imagen 13, ficheros del producto. ....	73

## 1.- Descripción del proyecto.

Una importante red social nos ha solicitado un sistema de base de datos que facilite la gestión de los regalos en grupo.

Con este nuevo módulo, la red social quiere diferenciarse de la competencia ofreciendo nuevas funcionalidades que permitan estar más “unidos” entre sus amistades.

Los beneficios principales del nuevo sistema serán los siguientes:

- **Garantía de satisfacción:** Para evitar la difícil tarea de no saber que regalar en un acontecimiento, será el mismo usuario quien seleccione los artículos que quiera recibir. De esta manera se simplifica la elección del regalo y nos aseguramos que no será devuelto.
- **Visibilidad/participación:** Cualquier amigo podrá crear un grupo de compra describiendo la finalidad (cumpleaños, navidad, despedida de trabajo, etc) y siendo visible para el resto de amistades del homenajeado. De esta forma la participación en la compra implicará que los participantes ni siquiera se deban conocer entre ellos. Con esta medida se espera que la participación del grupo de compra aumente en participantes.
- **Gestión de la recaudación:** El sistema registrará todas las acciones de pago que vayan realizando los participantes. De esta manera se elimina la figura de tesorero que normalmente nadie quiere realizar por la responsabilidad que conlleva.
- **Compra/envío automático:** Una vez se llegue al importe del regalo antes de la fecha máxima de recaudación, el sistema ordenará la compra en la tienda virtual indicando la dirección de envío evitando que ningún integrante del grupo tenga que ocuparse de esta acción.

El sistema a implementar se centrará en el diseño de la base de datos y en los procedimientos que debe ofrecer para almacenar/consultar la información.

No se tendrá que implementar interfaz gráfico ya que esta funcionalidad ya recae sobre el sistema de la red social.

Una vez explicado el objetivo del proyecto, en el siguiente apartado se explican los requisitos funcionales del sistema.

## 1.1.- Requisitos funcionales del sistema.

Los requisitos funcionales que tiene que cumplir la base de datos serán los siguientes:

[R1]: Datos mínimos que debe almacenar del grupo de participantes:

- Identificador de la persona homenajeada (email).
- Regalos seleccionados por el grupo de la lista de deseos.
- Dedicatoria general de todo el grupo.
- Identificadores de los miembros que forman el grupo de compra. De cada uno de los miembros también se guardará:
  - Dedicatoria personal de manera opcional.
  - Cantidad económica que ha contribuido en cada uno de los regalos seleccionados. La cantidad en la que cada miembro participa es libre.
- Fecha límite de la recogida de fondos.
- Identificador de la persona receptora del regalo (puede coincidir con la persona homenajeada o ser otra persona que más tarde le entregue en persona).
- Dirección postal del envío del regalo.
- Fecha límite de la recepción del regalo.
- Booleano que indique si el importe de cada participación será público dentro del grupo o solo se mostrará los miembros que han participado.
- Booleano que indique si el grupo ha estado eliminado. No se permitirán hacer bajas físicas porque necesitamos guardar los datos para posteriores estadísticas.
- Se permitirá crear otros atributos que puedan considerarse de interés.

[R2]: El modelo también guardará los datos asociados a los regalos seleccionados por el grupo:

- Los regalos se identifican por su código de producto y estarán asociados a un proveedor concreto.
- Los regalos también tendrán una categoría asociada. Las categorías estarán codificadas y se podrán ampliar en el futuro.
- Cada regalo tendrá un coste económico asociado y un tipo de IVA concreto.
- Cada regalo tiene un estado asociado en función del siguiente ciclo de vida:
  - Añadido\_a\_un\_grupo: Estado inicial cuando se selecciona un regalo en un grupo.
  - Eliminado: Cuando se decide eliminar un regalo. Si el regalo tuviera aportaciones de los miembros del grupo, se deberá realizar sus correspondientes devoluciones.
  - No\_suficientes\_fondos: Cuando se supere la fecha mínima de recogida de fondos y el importe recaudado no llegue al importe del artículo.
  - Pago\_completado: Cuando se consigue el importe del artículo
  - Enviado: Cuando el proveedor envía el artículo.
  - Recibido: Cuando se recibe el artículo.
- Si el equipo del proyecto lo considera necesario, se podrían ampliar los estados de los productos para dar más detalles en su gestión.

[R3]: Todas las transacciones económicas (pagos y devoluciones) deberán quedar registradas de forma automática para posteriores auditorías. La información mínima será:

- Identificador de la persona.
- Identificador del grupo.
- Identificador del regalo.
- Importe de la aportación/devolución.
- Método de pago elegido con sus datos asociados:
  - Número de cuenta bancaria.
  - Número de tarjeta de crédito.
  - Paypal.
  - Etc.....

[R4]: El sistema también tendrá que contemplar tablas para dejar los catálogos de cada proveedor. Con esto queremos indicar que cada proveedor deberá poder guardar sus productos clasificados en categorías.

[R5]: La aplicación tendrá que ofrecer los siguientes procedimientos:

- Procedimientos de modificación:
  - A. Alta y baja de un grupo de regalos. Las bajas serán lógicas permitiendo contabilizar los datos en los procesos de consulta posteriores.
  - B. Alta y baja de regalos dentro de un grupo de participantes. En el caso de las bajas si se detecta que se han realizado pagos, se deberá realizar automáticamente las devoluciones correspondientes a los integrantes que hayan participado.
  - C. Alta y baja de participantes de un grupo de participantes. Si se da de baja una persona que ha participado se deberá realizar una devolución del importe.
  - D. Alta y baja del homenajeador en el grupo de compra. Si existen regalos seleccionados cuando queremos realizar la baja, el sistema deberá cambiar el estado de los regalos a “cancelado” y realizar devoluciones a los integrantes que hayan participado.
  - E. Alta baja y modificación de las aportaciones en un regalo. Estas modificaciones incluyen las transacciones económicas en la tabla de las transacciones.
  - F. Alta, baja y modificación de la lista de regalos de un miembro de la red social. En el proceso de alta el usuario podrá indicar si permite que ese regalo pueda ser seleccionado por futuros grupos de compra. De esta manera el usuario podrá decidir si desea repetir regalo o no. En el caso de eliminar un producto de su lista, se deberá contrastar si ese regalo está en un grupo de compra pendiente de recogida de dinero para cambiar el estado a “eliminado” y realizar las devoluciones.
  - G. Para simplificar no hará falta implementar los procedimientos correspondientes a las tablas auxiliares como:
    - a) Categorías.
    - b) Proveedores.
    - c) Catálogos de regalos del proveedor.
    - d) Países.
    - e) Provincias.

- f) Municipios, etc...
- Procedimientos de consulta:
    - Dado un miembro de la red social: el listado de regalos que desea incluyendo el precio y el proveedor asociado, así como la fecha de alta del deseo.
    - Listado de todos los grupos de participantes que todavía tienen un regalo pendiente. Indicará la suma del importe total de todos los regalos, el importe que queda pendiente de conseguir y el porcentaje respecto al importe total conseguido y el restante.
    - Dada una categoría: el listado de todos los regalos asociados, incluyendo el proveedor que lo proporcionen.
    - Dado un usuario de la red social: el listado de todos los regalos que ha recibido.
    - El listado de las personas que se encuentran por encima de la media a nivel de dinero que han destinado a los regalos grupales.
    - Dado un año: El listado de las 10 ciudades donde más regalos se han recibido ese año. Cada una de las 10 ciudades también mostrará el sumatorio total del importe de los regalos.

Se valorará la implementación de otros procedimientos o funcionalidades que puedan ser de utilidad y deberán ser consensuados con el tutor.

[R7]: Módulo estadístico: Los siguientes procedimientos estadísticos tendrán la restricción que el coste de su ejecución tendrá que ser constante. Esto quiere decir que no se pueden obtener utilizando vistas, funciones de agregación (sum, avg, etc). Tampoco se podrá delegar en un proceso nocturno que actualice la información como normalmente suele funcionar los sistemas data warehouse. En resumen, la información deberá estar siempre actualizada y el coste de la consulta siempre será el mismo.

Las consultas del módulo estadístico son las siguientes:

- Dado un producto y un año: el número de veces que se ha regalado a diferentes personas. No hace falta contemplar los regalos que nunca se han regalado.
- Dado un año: el número de regalos que se han adquirido utilizando nuestra aplicación.
- Dado un año: el importe total de los regalos que se han adquirido en nuestra aplicación.
- Dado un año: el porcentaje de regalos que han sido comprados respecto a los que tienen un estado "no\_suficientes\_fondos".
- Dado un año: el porcentaje medio de gasto en regalos de los miembros de la red.
- Dado un año y un mes concreto: el identificador de la persona que más regalos ha recibido.
- Dado un año y un mes concreto: identificador de la persona que más dinero ha contribuido.
- Dado un año y un mes concreto: el identificador de la persona que más ha participado en regalos.
- Dado un año y un mes concreto: el identificador del grupo que más gasto en regalos ha realizado.

## 1.2.-Requerimientos de la metodología.

A la hora de estandarizar los procedimientos, el cliente nos comunica que debemos cumplir con los siguientes requisitos:

- Todo procedimiento tendrá como mínimo un parámetros de salida con el nombre RSP de tipo string, que indicará si la ejecución ha finalizado correctamente (valor "OK") o si ha sucedido un error (valor "ERROR + tipo de error").
- Todos los procedimientos tratarán las excepciones que se puedan producir, trasladando lo ocurrido en el parámetro de salida RSP.
- El sistema guardará todas las peticiones a procedimientos en una tabla de log, guardando el nombre del procedimiento ejecutado, los parámetros de entrada y de salida.
- Los procedimientos tendrán que estar bien documentados para facilitar el uso a los programadores de la capa de presentación. Esto implica:
  - Describir los procedimientos a alto nivel.
  - Describir el tipo y el dominio de valores de cada parámetro de entrada/salida y los diferentes mensajes de error controlados que puede devolver.
  - Realizar los comentarios necesarios en el código para su correcto mantenimiento.

## 1.3.- Responsabilidad/seguridad en el uso de los procedimientos.

Todas las interacciones (altas, bajas, modificaciones, consultas) deberán realizarse a partir de los procedimientos que se definan. No nos hacemos responsables de los posibles problemas que se puedan ocasionar el hecho de manipular los datos accediendo directamente a las tablas.

Por otra parte el sistema desconoce por completo todas las relaciones de amistad que gestiona la red social y no deberá comprobar sus afinidades. Esto significa que se si añade una persona en un grupo de compra es porque la red social ya valida esa transacción previamente y nosotros hemos de realizar esas acciones sin ponerlas en duda.

## 1.4.- Metodología del desarrollo del proyecto.

A la hora de crear un software se debe de elegir una metodología de cómo se va a desarrollar el software. En este proyecto se ha decidido realizar una metodología en cascada porque el tiempo del proyecto es reducido (3 meses) y la complejidad del proyecto no es elevada.

La principal característica del ciclo de vida en cascada es que se debe dar por finalizado una fase antes de empezar la siguiente. Las diferentes fases las podemos observar en la imagen 1:

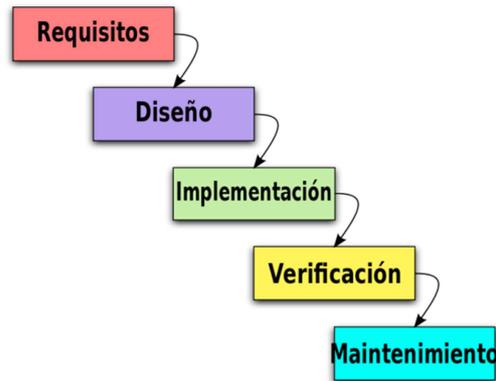


Imagen 1, fases del ciclo de vida.

Como se explica en el punto [1.5.3.- Productos obtenidos.](#), cada fase del ciclo de vida en cascada va acompañada de una entrega en la documentación, exceptuando la fase del mantenimiento que queda fuera del alcance inicial del proyecto.

## 1.5.- Planificación.

En los siguientes apartados se definen el conjunto de actividades y su estimación a lo largo del proyecto.

### 1.5.1.- Relación de actividades y estimación del tiempo.

Dentro de la gestión del propio proyecto, se definen una relación de actividades consecutivas asignándole un tiempo aproximado. A partir del calendario definido se puede detectar desviaciones en la planificación original y tener que aplicar las medidas que se han definido en el apartado: 4.- Plan de contingencias., para conseguir terminar el proyecto en la fecha establecida.

El horario laboral que se ha definido a lo largo de todo el proyecto es de una dedicación diaria de 2h. Se ha decidido ese valor para simplificar los cálculos y no tener en cuenta los días festivos del calendario. Además con esa medida es posible aumentar la dedicación si se detectaran retrasos respecto la planificación inicial.

En la tabla 1 se muestra la relación de actividades necesarias para realizar el proyecto:

Nombre de tarea	Duración	Comienzo	Fin
<b>Inicio proyecto</b>	0 días	mié 26/02/14	mié 26/02/14
<b>plan de trabajo (PEC1)</b>	<b>17,5 días</b>	<b>mié 26/02/14</b>	<b>sáb 15/03/14</b>
<b>Lectura requerimientos TFC</b>	3 hrs	mié 26/02/14	jue 27/02/14
<b>Análisis preliminar del enunciado</b>	5 hrs	jue 27/02/14	sáb 01/03/14

Consultas al cliente	3 hrs	dom 02/03/14	lun 03/03/14
Consulta otros proyectos	5 hrs	lun 03/03/14	mié 05/03/14
Lectura redacción de textos	4 hrs	jue 06/03/14	vie 07/03/14
Instalación software necesario (MS project)	1 hr	sáb 08/03/14	sáb 08/03/14
Generar documentación PEC1	10 hrs	sáb 08/03/14	jue 13/03/14
Revisión documentación	4 hrs	jue 13/03/14	sáb 15/03/14
Entrega PEC1	0 hrs	dom 16/03/14	dom 16/03/14
<b>Análisis y diseño (PEC2)</b>	<b>28 días</b>	<b>dom 16/03/14</b>	<b>dom 13/04/14</b>
Instalación software necesario (magic draw uml)	1 hr	dom 16/03/14	dom 16/03/14
Análisis requerimientos	15 hrs	dom 16/03/14	dom 23/03/14
Preguntas cliente	5 hrs	lun 24/03/14	mié 26/03/14
Generación casos de uso	3 hrs	mié 26/03/14	jue 27/03/14
Diseño conceptual	10 hrs	vie 28/03/14	mar 01/04/14
Diseño lógico	4 hrs	mié 02/04/14	jue 03/04/14
Diseño físico	4 hrs	vie 04/04/14	sáb 05/04/14
Generar documentación PEC2	10 hrs	dom 06/04/14	jue 10/04/14
Revisión documentación	4 hrs	vie 11/04/14	sáb 12/04/14
Entrega PEC2	0 días	dom 13/04/14	dom 13/04/14
<b>Implementación y pruebas (PEC3)</b>	<b>27 días</b>	<b>lun 14/04/14</b>	<b>dom 11/05/14</b>
Instalación software (Oracle 11/ sql Developer)	3 hrs	lun 14/04/14	mar 15/04/14
Creación entorno (usuario, tablespace,tablas, constraints..)	4 hrs	mar 15/04/14	jue 17/04/14
Creación procedimientos ABM	14 hrs	jue 17/04/14	jue 24/04/14
Creación procedimientos de consulta	4 hrs	vie 25/04/14	sáb 26/04/14
Creación procedimientos de estadísticos	5 hrs	dom 27/04/14	mar 29/04/14
Creación procedimiento de logs	5 hrs	mar 29/04/14	jue 01/05/14
Pruebas unitarias	4 hrs	vie 02/05/14	sáb 03/05/14
Generar documentación PEC3	10 hrs	dom 04/05/14	jue 08/05/14
Revisar documentación	2 hrs	vie 09/05/14	vie 09/05/14
Entrega PEC3	0 días	dom 11/05/14	dom 11/05/14
<b>Entrega del proyecto</b>	<b>34 días</b>	<b>lun 12/05/14</b>	<b>dom 15/06/14</b>
Pruebas finales	10 hrs	lun 12/05/14	vie 16/05/14
Scripts carga de datos y pruebas	10 hrs	sáb 17/05/14	mié 21/05/14
Generar memoria	20 hrs	jue 22/05/14	sáb 31/05/14
Generar presentación	20 hrs	dom 01/06/14	mar 10/06/14
Entrega final (producto, memoria y presentación)	0 días	dom 15/06/14	dom 15/06/14
<b>Tribunal virtual</b>	<b>3 días</b>	<b>mié 25/06/14</b>	<b>vie 27/06/14</b>
Defensa del proyecto	3 días	mié 25/06/14	vie 27/06/14

Tabla 1, relación de actividades y tiempo.

### 1.5.2.- Diagrama de Gantt

Una vez definido las tareas y su tiempo asociado, ya podemos representar su distribución a partir del diagrama de Gantt.

Las siguientes 4 imágenes (2, 3, 4 y 5), nos muestra el diagrama de Gantt por cada una de las fases. He decido dividir las imágenes para que se pueda visualizar mejor:

#### 1) Plan de trabajo:

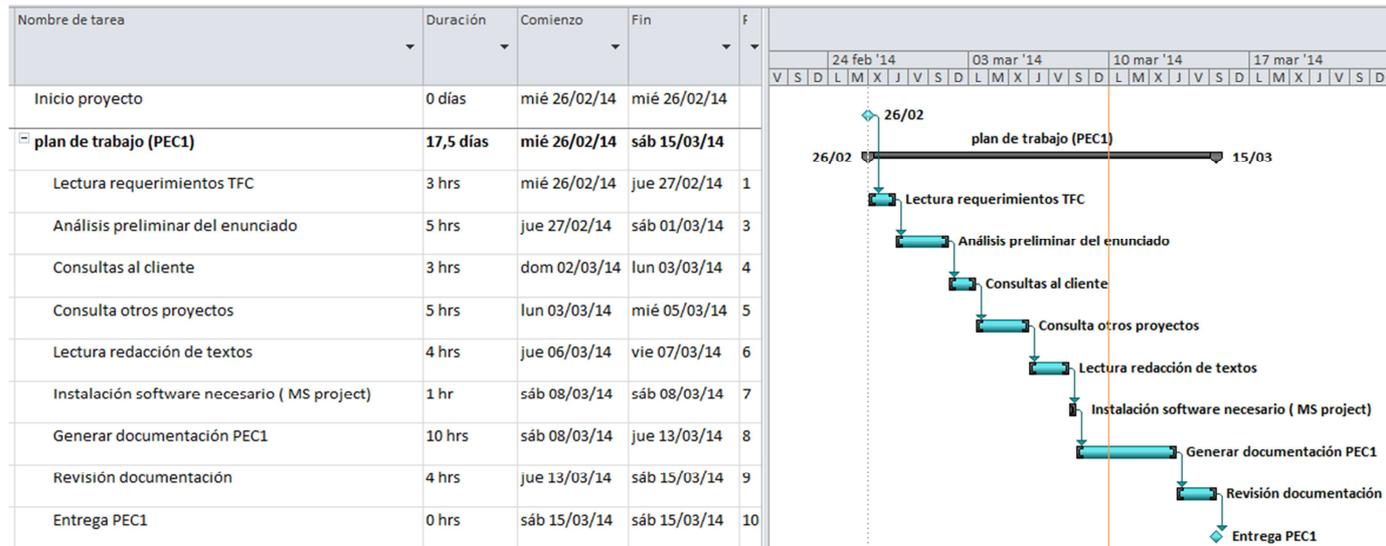


Imagen 2, plan de trabajo.

#### 2) Análisis y diseño:

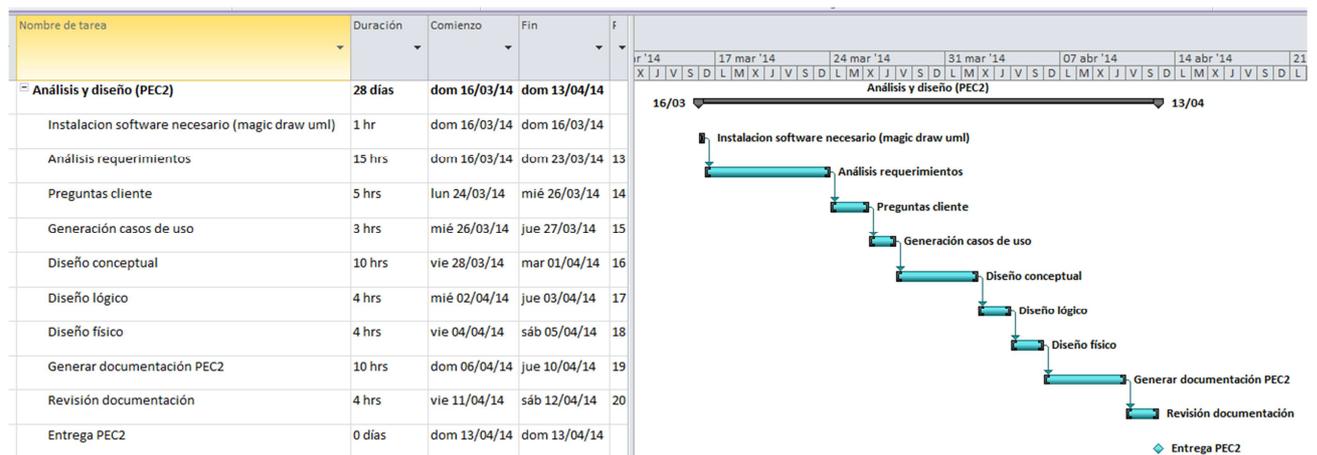


Imagen 3, análisis y diseño.

### 3) Implementación y pruebas:

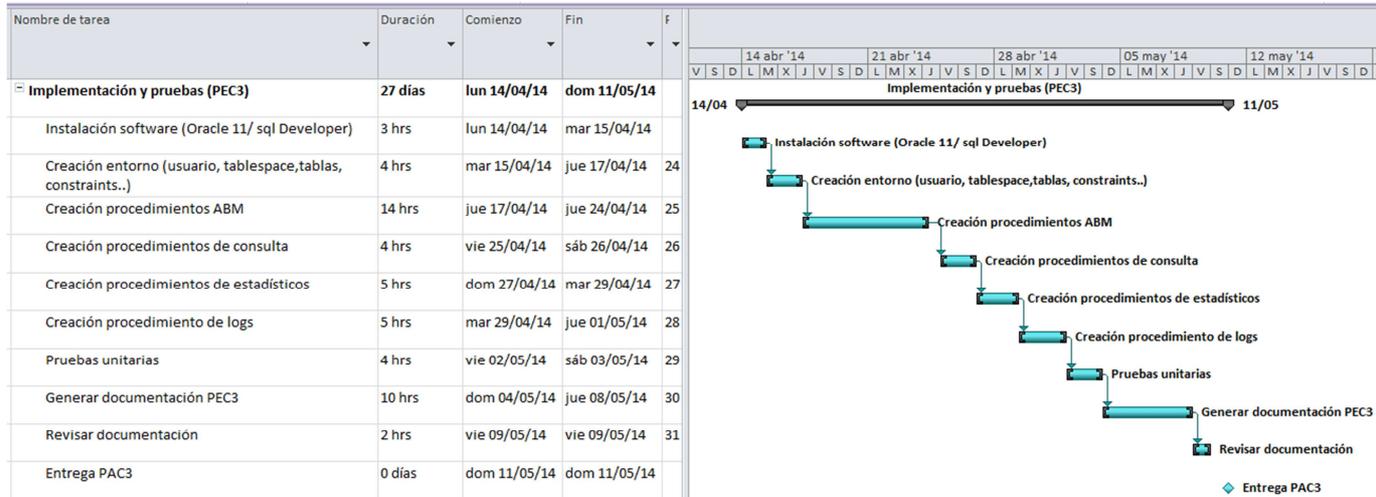


Imagen 4, implementación y pruebas.

### 4) Entrega y defensa:

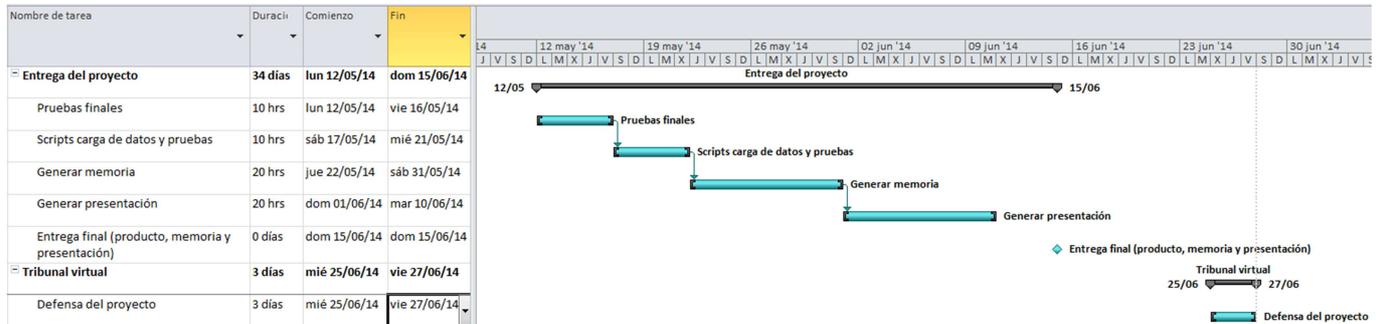


Imagen 5, entrega producto.

#### 1.5.3.- Productos obtenidos.

El proyecto se ha estructurado en 4 fases y cada una de ellas se obtiene la siguiente documentación:

- **PEC1 (Plan de trabajo):** Exposición de los requerimientos/funcionalidades, definición del alcance del proyecto y planificación de las tareas con su coste asociado.
- **PEC2 (Análisis y diseño):** Análisis de los requerimientos y diseño de la base de datos.
- **PEC3 (Implementación):** Creación de los objetos de la base de datos e implementación de los procedimientos.
- **PEC4 (Entrega del producto):** Entrega final de la memoria del proyecto, producto (todos los scripts) y presentación del proyecto.

Cada una de las fases anteriores tiene las siguientes fechas de entrega que podemos ver en la tabla 2:

Entregas	Fecha de entrega
PEC 1: Plan de trabajo	16/03/2014
PEC 2: Análisis y diseño	13/04/2014
PEC 3: Implementación	11/05/2014
PEC 4: Entrega del producto	15/06/2014

Tabla 2, fechas de entrega documentación/producto.

#### 1.5.4-Breve descripción del resto de capítulos.

Una vez se ha explicado el proyecto con su desarrollo, tiempos y productos obtenidos, se realiza una breve descripción de los capítulos que veremos a continuación:

- **Análisis y diseño:** Se analiza los requerimientos del usuario, se proponen mejoras del producto y se realiza el diseño de la base de datos.
- **Implementación:** Fase de construcción del software. Se definen el interfaz para que el sistema de la red social pueda interactuar con el sistema de compras grupales.
- **Plan de contingencias:** Identificación de las causas que pueden comportar un retraso en la correcta ejecución del proyecto con la relación de acciones a realizar para que no sucedan.
- **Plan de pruebas:** Conjunto de pruebas que se ofrecen al cliente para garantizar el correcto funcionamiento del producto.
- **Módulo estadístico:** Es tal la importancia de este módulo que se decide centralizar en un apartado toda la información que hace referencia a este bloque (análisis, procedimientos y pruebas).
- **Valoración económica y recursos necesarios:** Identificación de los recursos/tiempos necesarios para la construcción del software y valoración económica del proyecto.
- **Mejoras futuras:** Propuestas para seguir evolucionando el programa.
- **Conclusiones:** Resumen de las conclusiones del proyecto.
- **Glosario:** Identificación y explicación de las palabras utilizados en la memoria.
- **Bibliografía:** Relación de las fuentes de información consultadas.
- **Anexos:** Información complementaria del proyecto.

## 2.- Análisis y diseño.

Dentro de este capítulo encontraremos toda la información que hace referencia al análisis y diseño del nuevo sistema de información a construir.

A la hora de diseñar una base de datos, esta se divide en los siguientes apartados:

- **Revisión de los requerimientos:** Se detectan funcionalidades incoherentes y se aportan una serie de mejoras que darán más calidad al producto.
- **Diseño conceptual:** Se identifican los diferentes casos de uso que van a interactuar con el sistema y se define el modelo entidad-relación en UML.
- **Diseño lógico:** A partir del diseño conceptual, se concretan los elementos y las relaciones entre las entidades.
- **Diseño físico:** Optimización de las estructuras para que puedan ser utilizadas en el SGBD.

### 2.1.- Revisión de los requerimientos.

Una vez analizado todos los requerimientos del cliente, nos encontramos en disposición de justificar los requerimientos incoherentes y un conjunto de mejoras que facilitará el uso del programa.

#### 2.1.1.- Procedimientos incoherentes.

En los siguientes puntos se identifican los requerimientos que el analista ha considerado innecesarios:

1. **Alta homenajado:** En el momento de crear un grupo de compra ya debemos saber quién será la persona homenajada, con lo cual, no tiene sentido tener un procedimiento independiente para realizar esta acción.
2. **Baja homenajado:** Cuando se crea un grupo de compra toda la información de ese grupo hace referencia a la persona homenajada (regalos, participantes, felicitación, dirección de entrega, motivo del grupo de compra, etc...). El permitir cambiar el usuario homenajado hace que la información que gestiona el grupo deje de tener sentido y más cuando no existen procedimientos para cambiar el resto de atributos del grupo de compra. Si se produce la necesidad de querer cambiar el usuario homenajado vemos más correcto el dar de baja el grupo y crear uno nuevo con los datos correctos.

### 2.1.2.- Mejoras aplicadas.

Las mejoras que pueden ser introducidas dentro de la planificación inicial son las siguientes:

1. **Árbol de categorías:** Para una correcta organización de las categorías que puedan ofrecer los proveedores, se hace necesario una estructura de tipo árbol donde los primeros niveles queden las categorías más genéricas y en los últimos las más específicas.
2. **Identificación de la multiplicidad del deseo:** Uno de los puntos más críticos del nuevo sistema es la gestión de los regalos del homenajeado. Inicialmente no se nos ha indicado que pasa cuando varios grupos de compra con el mismo homenajeado, están tramitando el mismo regalo de manera paralela. La situación final puede ser frustrante y sería un motivo de fracaso del proyecto. Para evitar esta situación proponemos los siguientes cambios:
  - a. Cuando el usuario define la lista de deseos, debe indicar si desea recibir ese regalo de manera única o por el contrario se pueden repetir en varios grupos de compra.
  - b. Si el usuario ha definido un regalo único, solo un grupo podrá tramitarlo. En el caso que ese grupo elimine el regalo o no llegue a tiempo a la recaudación, será seleccionable por el siguiente grupo de compra.
  - c. Si el usuario cambia la propiedad de un regalo no único a único, el sistema debe comprobar cuantos grupos de compras tienen seleccionado ese regalo en un estado que lo pueda eliminar. De ese conjunto solo debe dejar activo el que más dinero tenga recaudado siempre y cuando no exista el mismo producto en un estado a medio tramitar por el proveedor, empresa de reparto o haya sido entregado.
3. **Configuración parametrizable:** Existen ciertos valores a la hora de realizar comprobaciones en los procedimientos que se han definido en una tabla de parámetros. De esta manera su mantenimiento es mucho más fácil, ágil y está centralizado en una tabla. Un ejemplo del uso sería:
  - a. Definir el número de días máximo que puede existir entre la fecha límite de recaudación con la fecha del envío.
  - b. Definir el margen de días que la empresa de transporte puede informar respecto a la fecha del grupo de compra.
4. **Baja automática de los grupos de compra:** Una vez se ha superado la fecha de recogida de fondos, no tiene sentido seguir interaccionando con el grupo de compra por parte de los participantes del grupo. Así que se aprovechará el proceso batch que cambia los regalos pendientes de recaudación, al estado sin fondos suficientes y marcará el grupo de compra de baja. De esta manera se bloquean las acciones de modificación como serían: añadir regalos, alta de participantes, etc...

5. **Procedimientos de consulta:** Se decide complementar las siguientes consultas para dotar al sistema de más información:
- a. **Consulta de categorías:** Listado sin parámetros de entrada que mostrará el identificar de la categoría, su descripción, el número de productos asociados, el número total de productos que son ofrecidos por los diferentes proveedores en todos los países y en el caso de pertenecer a una categoría superior, su identificador y la descripción. Esta información será útil para poder llamar al listado de productos que ofrecen los diferentes proveedores ya que requiere informar de una categoría. Sin la existencia de este listado el usuario tendría que ir probando valores hasta encontrar uno que fuera correcto.
  - b. **Consulta de personas homenajeadas:** Lo primero que piensa un amigo antes de crear un grupo de compra es: ¿Cómo puedo saber si ya existe un grupo de compra de la personas x? Para evitar esta situación se ha pensado en reaprovechar el listado que muestra la información de los grupos pendientes de pago. En el caso de informar el identificador de la persona homenajeadado (email) devolverá la información de todos los grupos que se hayan creado indistintamente del estado en que se encuentre. De esta manera podrá seguir consultando con otras consultas, los regalos de cada grupo, sus participantes, etc...
  - c. **Consulta de regalos de un grupo:** Una opción interesante a la hora de realizar los pagos es saber que regalos están pendiente de ser completados y el importe pendiente que puede realizar.
  - d. **Consulta de pagos realizados:** Ya que el sistema guarda todas las transacciones de los pagos, debe existir un procedimiento de consulta donde pueda ver los pagos que ha realizado un participante en un grupo de compra.
  - e. **Consulta de participantes:** Creemos necesario un listado que muestre la relación de participantes de un grupo concreto, con su estado (alta, baja) y el sumatorio del importe aportado en regalos. Además si se informa del email del participante sacará la misma información pero de todos los grupos donde conste el participante.

Con todas estas mejoras aumentamos las funcionalidades del sistema ofreciendo un producto potente y funcional.

Añadir también que existen otras mejoras que el equipo de analistas ha detectado. Estas serán explicadas en el apartado: [8.- Mejoras futuras.](#)

## 2.2.- Diseño conceptual.

En la primera fase del diseño, debemos identificar los conceptos más representativos que nos ayude a obtener un modelo de base de datos independiente al SGBD utilizado.

El primer paso es identificar los diferentes casos de uso que interactúan con el sistema para así tener claro toda la información que el sistema debe guardar. El segundo paso se creará un diagrama entidad-relación en UML.

### 2.2.1.- Diagrama de casos de uso.

Con el diagrama de casos de uso podemos representar que funcionalidades debe aportar el sistema y que actor/usuario puede interactuar con ella.

En la siguiente imagen se muestra el diagrama de casos de uso de nuestro sistema de información.

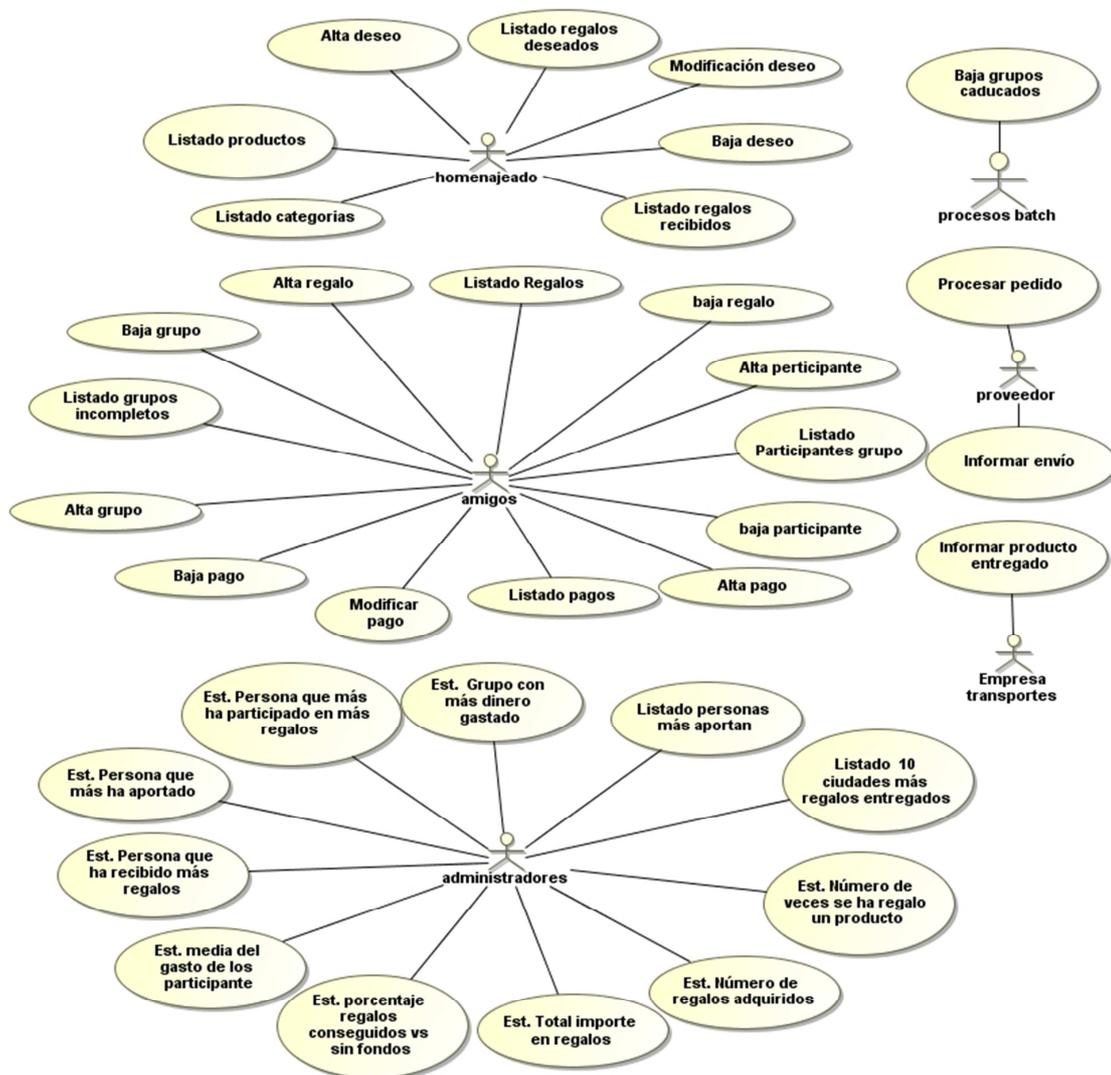


Imagen 6, diagrama de casos de uso.

Como se puede observar en la imagen anterior, se han identificado los siguientes usuarios que podrán realizar las siguientes acciones:

- **Homenajeado:** Usuario de la red social que podrá realizar las siguientes acciones:
  - Alta/baja/modificar los regalos que quiere recibir.
  - Consultar las categorías, productos y regalos recibidos.
  
- **Amigos:** Son las amistades del homenajeado que participan en un grupo de compra. Las acciones que puede además de las del propio homenajeado son:
  - Alta/Baja de un grupo.
  - Alta/Baja de los regalos del grupo.
  - Alta/Baja de participantes en el grupo.
  - Alta/baja/modificación de los pagos.
  - Consultas: grupos, regalos, participantes y pagos.
  
- **Administradores:** El usuario administrador será en encargado de comprobar el correcto funcionamiento del sistema con todas las consultas anteriores y las del **módulo estadístico**.
  
- **Procesos batch:** Proceso nocturno encargado de cambiar el estado a todos los productos seleccionados por los grupos de compra que no han llegado al importe en la fecha máxima.
  
- **Proveedor:** Los proveedores que se den de alta en el sistema utilizarán los siguientes métodos:
  - Acceder al listado de los productos que han de tramitar.
  - Informar del código de seguimiento del envío.
  
- **Repartidor:** La empresa de reparto nos informará a partir del número de seguimiento que el paquete ha sido entregado en su destino.

La relación de usuarios anteriores es una manera de agrupar las funcionalidades del programa en usuarios pero la seguridad de quien ejecuta cada procedimiento recae sobre la aplicación de la red social.

### 2.2.2.- Diagrama entidad relación.

En la siguiente imagen, podemos observar el modelo entidad relación que identifica las entidades, atributos y relaciones.



<b>Descripción</b>	Nombre del proveedor	Amazon
--------------------	----------------------	--------

Tabla 3, entidad Proveedores.

Entidad: Productos		
<b>Descripción</b>	Identifica a los productos que ofrecen los proveedores.	
<b>Relaciones</b>	Participa en la entidad asociativa “precios_productos” y pertenece a una categoría para mejorar su clasificación.	
Atributos	Descripción	Ejemplo valor
<b>Id_producto</b>	Identificador único del producto	1
<b>Descripción</b>	Nombre del producto	Nikon D4s
<b>Baja</b>	Gestiona la baja lógica del registro.	F

Tabla 4, entidad Productos.

Entidad: Categorías		
<b>Descripción</b>	Identifica a categorías que servirán para clasificar mejor los diferentes productos.	
<b>Relaciones</b>	Con la entidad productos y de manera reflexiva para crear subcategorías	
Atributos	Descripción	Ejemplo valor
<b>Id_categoria</b>	Identificador único de la categoría.	1
<b>Descripción</b>	Nombre de la categoría	Alimentación

Tabla 5, entidad categorías.

Entidad: Tipos_iva		
<b>Descripción</b>	Identifica los diferentes tipos de iva que tiene cada país.	
<b>Relaciones</b>	Con la entidad asociativa “precios_productos”.	
Atributos	Descripción	Ejemplo valor
<b>Id_tipo_iva</b>	Identificador del tipo de IVA.	1
<b>Descripción</b>	Descripción del tipo de IVA.	Normal
<b>IVA</b>	Porcentaje del impuesto	21

Tabla 6, entidad tipos\_iva.

Entidad: Países.		
<b>Descripción</b>	Identifica a los países.	
<b>Relaciones</b>	Con la entidad tipos_iva y provincias.	
Atributos	Descripción	Ejemplo valor
<b>Id_pais</b>	Identificador único del país.	1
<b>Descripción</b>	Nombre del país	España

Tabla 7, entidad Países.

Entidad: Precio_productos.		
<b>Descripción</b>	Entidad asociativa que define el precio del producto en función del proveedor y del tipo de impuesto que tenga en el país que lo distribuye.	
<b>Relaciones</b>	Participa en una relación asociativa “lista_deseos”.	
Atributos	Descripción	Ejemplo valor
<b>Precio</b>	Precio del producto (sin aplicar el impuesto asociado)	150,45
<b>Baja</b>	Gestiona la baja lógica del registros	F

Tabla 8, entidad precio\_productos.

Entidad: Usuarios.		
<b>Descripción</b>	Identifica a los usuarios de la red social que intervienen en nuestro sistema.	
<b>Relaciones</b>	<p>Es la entidad más referenciada en nuestro modelo. Las relaciones son las siguientes:</p> <ul style="list-style-type: none"> <li>Entidad asociativa “ejercicio_meses”: 3 relaciones identificando al usuario que más regalos ha recibido, más dinero ha gastado y más ha participado.</li> <li>Entidad “precios_productos” para crear la clase asociativa “lista_deseos”.</li> <li>Entidad “grupo_compras”: 3 relaciones de las cuales una relaciona la persona homenajeadada, la persona receptora del regalo y finalmente se crea una entidad asociativa “participantes” con las personas que participan en el grupo de compra.</li> <li>Entidad “envios” donde hace referencia a la persona que finalmente se hace cargo de la recepción del regalo.</li> </ul>	
Atributos	Descripción	Ejemplo valor
<b>Id_usuario</b>	Identificador del usuario en nuestro sistema	1
<b>email</b>	Identificador del usuario de la red social	<a href="mailto:Juan.mar@gmail.com">Juan.mar@gmail.com</a>

Tabla 9, entidad Usuarios.

Entidad: Mes		
<b>Descripción</b>	Entidad que enumera los meses del año.	
<b>Relaciones</b>	Relación con la entidad “ejercicios” para formar la entidad asociativa “ejercicio_meses”	
Atributos	Descripción	Ejemplo valor
<b>Mes</b>	Numérico que identifica el mes	5

Tabla 10, entidad Mes.

Entidad: Grupo_compras		
<b>Descripción</b>	Entidad que gestiona las propiedades de un grupo de compra.	
<b>Relaciones</b>	Tiene una relación con la entidad “ejercicio_meses” para indicar el grupo que más gasto ha realizado. Participa en las siguientes entidades asociativas “participantes” y “regalos”.	
Atributos	Descripción	Ejemplo valor
<b>Id_grupo</b>	Identificador único del grupo	1
<b>Dedicatoria</b>	Dedicatoria general del grupo	Muchas felicidades, ya son 51 años!
<b>D_limite_fondos</b>	Fecha límite para recoger el dinero de los regalos.	15-05-2014
<b>D_recep_regalo</b>	Fecha en la que se tienen que recibir los regalos.	20-05-2014
<b>Importe_publico</b>	Indica si el importe de cada participante es público o no.	S
<b>Baja</b>	Gestiona la baja lógica de la entidad.	N

Tabla 11, entidad Grupo\_compras.

Entidad: Participantes		
<b>Descripción</b>	Entidad asociativa que relaciona los usuarios que participan en un grupo de compra.	
<b>Relaciones</b>	Tiene relación con la entidad “envíos” ya que debe guardar el participante que realiza la aportación.	
Atributos	Descripción	Ejemplo valor
<b>dedicatoria</b>	Texto personalizado del participante al homenajeado	Felicidades campeón!!!!
<b>Baja</b>	Gestiona la baja lógica de la entidad	N

Tabla 12, entidad Participantes.

Entidad: Lista_deseos		
<b>Descripción</b>	Entidad asociativa que guarda los productos que desea recibir por parte de los grupos de compra.	
<b>Relaciones</b>	Participa con la entidad “grupo_compras” en la entidad asociativa “regalos”.	
Atributos	Descripción	Ejemplo valor
<b>D_alta</b>	Fecha que selecciona un producto	15-04-2014
<b>Regalo_unico</b>	Indica si quiere recibir más de una vez ese regalo por grupos diferentes de compras	N

Tabla 13, entidad Lista\_deseos.

Entidad: Regalos		
<b>Descripción</b>	Representa los regalos seleccionados por el grupo de compra de la lista de deseos del homenajeado.	
<b>Relaciones</b>	La entidad está relacionada con la entidad “envios” y la entidad “pagos”.	
Atributos	Descripción	Ejemplo valor
<b>D_mod_estado</b>	Fecha que se actualiza en cada cambio de estado del producto.	15-04-2014
<b>Precio_final</b>	Importe final del producto (precio + impuestos) y que no variará aunque cambie el precio del producto por parte del proveedor o se cambie los impuestos.	156,38

Tabla 14, entidad Regalos.

Entidad: Estados_regalos		
<b>Descripción</b>	Maestro que identifica todos los estados que puede tener un producto.	
<b>Relaciones</b>	Tiene una relación con la entidad “regalos” y 2 relaciones con la entidad “autómata_estados” para indicar la pareja de estados válidos en una transición.	
Atributos	Descripción	Ejemplo valor
<b>Id_estado</b>	Identificador único del estado	1
<b>Descripción</b>	Descripción del estado	Añadido a un grupo

Tabla 15, entidad Estados\_regalos.

Entidad: Automata_estados		
<b>Descripción</b>	Entidad que contendrá la lógica del negocio respecto los cambios de estados de los productos. A partir de una transacción y un estado inicial, nos indica es estado final del producto.	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>
<b>Transacción</b>	Descripción de la transacción que queremos realizar	Eliminar producto

Tabla 16, entidad Automata\_estados.

Entidad: Tipos_celebraciones		
<b>Descripción</b>	Maestro que codifica los diferentes tipos de celebraciones.	
<b>Relaciones</b>	Se relaciona con la entidad "grupo_compras".	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>
<b>Id_celebración</b>	Identificador único del tipo de celebración	1
<b>Descripción</b>	Descripción del tipo de celebración	Cumpleaños

Tabla 17, entidad Tipos\_celebraciones.

Entidad: Provincias		
<b>Descripción</b>	Maestro de provincias.	
<b>Relaciones</b>	Con la entidad "municipios".	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>
<b>Id_provincia</b>	Identificador único de la provincia	1
<b>Descripción</b>	Descripción de la provinca	Barcelona

Tabla 18, entidad Provincias.

Entidad: Municipios		
<b>Descripción</b>	Maestro de municipios.	
<b>Relaciones</b>	Con la entidad "ubicaciones".	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>
<b>Id_municipio</b>	Identificador único del municipio	1
<b>Descripción</b>	Descripción del municipio	Terrassa

Tabla 19, entidad Municipios.

Entidad: Vías		
<b>Descripción</b>	Maestro de tipos de calle.	
<b>Relaciones</b>	Con la entidad "ubicaciones".	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>
<b>Id_via</b>	Identificador único del tipo de vía	1
<b>Descripción</b>	Descripción de la vía	Calle

Tabla 20, entidad Vías.

Entidad: Ubicaciones		
<b>Descripción</b>	Entidad que guarda las direcciones postales.	
<b>Relaciones</b>	Con la entidad "grupo_compras" para definir la dirección de entrega (esta puede cambiar a lo largo del tiempo y la entidad "envíos" donde finalmente se envía los productos de un pedido.	
<b>Atributos</b>	<b>Descripción</b>	<b>Ejemplo valor</b>

<b>Id_ubicacion</b>	Identificador único de la dirección	1
<b>Direccion</b>	Texto principal de la dirección	Menendez y Pelayo
<b>Numero</b>	Número de la ubicación	10
<b>Piso</b>	El piso de la ubicación	1
<b>Puerta</b>	La puerta de la ubicación	2
<b>CP</b>	Código postal	08223

Tabla 21, entidad Ubicaciones.

Entidad: Envíos		
<b>Descripción</b>	Entidad que almacena los datos referentes a un envío.	
<b>Relaciones</b>	Con la entidad “regalos” para definir los productos que son enviados.	
Atributos	Descripción	Ejemplo valor
<b>Id_envio</b>	Identificador único del envío.	1
<b>Ref_externa</b>	Los envíos son gestionados por empresas de transportes y este sería su identificador único de su sistema	4131001486
<b>D_entrega</b>	Fecha en la que se realiza la entrega.	29-05-2014

Tabla 22, entidad Envíos.

Entidad: Pagos		
<b>Descripción</b>	Entidad que almacena las transacciones económicas de los participantes y los productos que quiere participar.	
<b>Relaciones</b>	Tiene una relación reflexiva para relacionar las devoluciones con el cobro positivo al que hace referencia.	
<b>Observaciones</b>	La entidad Pagos se especializa en subclases en función del medio de pago utilizado pudiendo ser: <ul style="list-style-type: none"> <li>• Cuenta bancaria.</li> <li>• Numero cuenta.</li> <li>• Paypal.</li> </ul>	
Atributos	Descripción	Ejemplo valor
<b>Id_pago</b>	Identificador único del pago	1
<b>Importe</b>	Cantidad que aporta. Puede ser positivo o negativo	50,50
<b>D_pago</b>	Fecha en la que se realiza el pago	29-05-2014
<b>Cuenta</b>	Atributo de la especialización “paypal”	<a href="mailto:Antonio@gmail.com">Antonio@gmail.com</a>
<b>Numero_tarjeta</b>	Atributo de la especialización “números_tarjetas”	1221-4611-7132-9541
<b>Numero_cuenta</b>	Atributo de la especialización “cuenta bacnaria”	ES8521030166321234567890

Tabla 23, entidad Pagos.

Entidad: Logs		
<b>Descripción</b>	Entidad que registra la actividad en los procedimientos pl/sql	
Atributos	Descripción	Ejemplo valor
<b>Id_log</b>	Identificador único del registro de log	1
<b>D_hora</b>	Fecha y hora en la que registra el log	24-04-2014 16:45:10
<b>Procedimiento</b>	Procedimiento ejecutado	Usuario.alta_deseo

<b>Inf_entrada</b>	Atributos utilizados en los parámetros de entrada	Id_usuario=1,id_producto_producto=2
<b>Inf_salida</b>	Atributos devueltos por el procedimiento	Id_deseo=1
<b>Inf_rsp</b>	Guarda la respuesta (confirmación/error) del procedimiento	OK

Tabla 24, entidad Logs.

Entidad: Parámetros		
<b>Descripción</b>	Almacena diferentes parámetros de la aplicación para que puedan ser cambiados sin la necesidad de tocar el código de la aplicación.	
Atributos	Descripción	Ejemplo valor
<b>Descripción</b>	Nombre del parámetros	INTERVALO_MAX_ENTREGA
<b>Valor</b>	Valor del parámetros	3

Tabla 25, entidad Parámetros.

Las entidades que hacen referencia al módulo estadístico están ubicadas en el siguiente apartado:

#### [6.1.1.1.- Entidades.](#)

### 2.3.- Diseño lógico.

A partir del diseño conceptual anterior, generamos el diseño lógico relacional que utilizaremos en nuestro gestor de base de datos Oracle 11g.

Para poder pasar de un diagrama a otro, necesitamos seguir las siguientes premisas:

- Las entidades originan relaciones.
- Las relaciones binarias (1:1 y 1:N) originan claves foráneas.
- Las interrelaciones M:N generan nuevas relaciones formando una clave única a partir de las claves principales. En nuestro modelo, hemos añadido una clave técnica como clave primaria para no ir arrastrando más claves foráneas de las necesarias en las siguientes interrelaciones. Esta es una decisión del modelo físico pero se ha implementado en esta fase porque el programa que genera el modelo lógico (sql data modeler) es capaz de generar los scripts de la estructura de la base de datos.
- Existen varias técnicas a la hora de transformar las especializaciones y teniendo en cuenta nuestro caso (el atributo que lo diferencia es de formato texto), se ha decidido crear un maestro de tipos de pago y un atributo de texto genérico en la misma tabla de pagos.
- No todas las entidades detectadas en el diagrama anterior, generan una relación en el diagrama lógico. Un ejemplo sería la enumeración “mes” que directamente es un atributo en la relación “ejercicios\_meses”

La siguiente imagen muestra el diseño lógico:

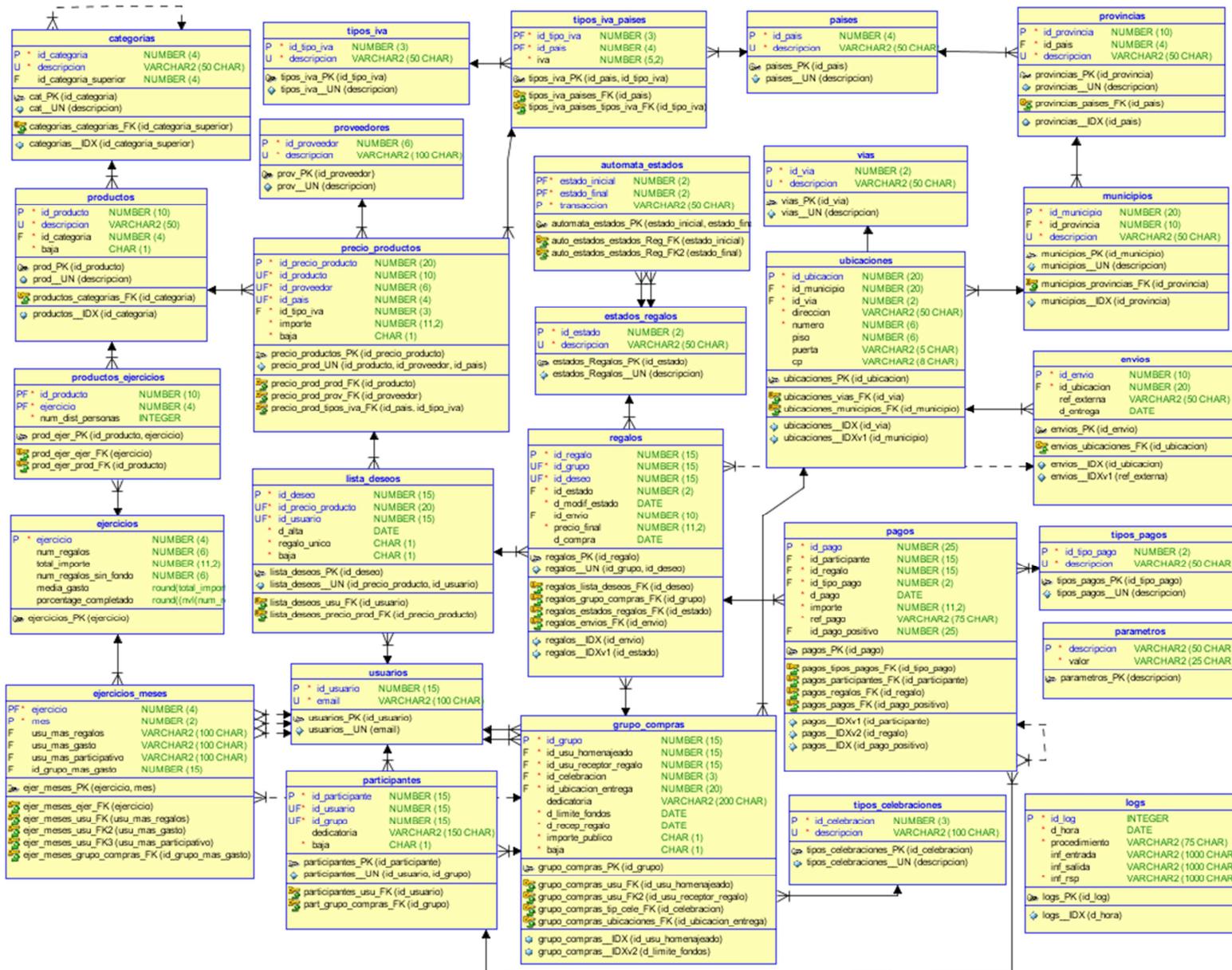


Imagen 8, diseño lógico.

La nomenclatura del modelo lógico es la siguiente:

- **Relaciones:** El nombre de la relación se resalta en negro.
- **Atributos:** Los atributos de las relaciones se agrupan entre paréntesis y separados por comas.
- **Clave primaria:** La clave principal se destaca entre los atributos con un subrayado.
- **Clave candidata:** La clave candidata se destaca sobre los atributos con un subrayado discontinuo.
- **Claves foráneas:** Los atributos que son claves foráneas se identifican con una flecha el “atributo.relación” del cual depende.
- **Atributos nulos:** Los atributos que puedan contener valores nulos se identifican con su nombre en cursiva>.

Las relaciones del modelo lógico son las siguientes:

1. **Países** (id\_pais, descripción).
2. **Provincias** (id\_provincia, id\_pais, descripción).
  - Id\_pais → id\_pais.países.
3. **Municipios** (id\_municipio, id\_provincia, descripción).
  - Id\_provincia → id\_provincia.provincias.
4. **Vías** (id\_vía, descripción).
5. **Ubicaciones** (id\_ubicacion, id\_municipio, id\_vía, dirección, numero, piso, puerta, cp)
  - id\_vía → id\_vía.vías.
  - id\_municipio → id\_municipio.municipios.
6. **Categorías** (id\_categoria, descripción, id\_categoria\_superior)
  - Id\_categoria\_superior → id\_categoria\_superior.categorias.
7. **Productos** (id\_producto, descripción, id\_categoria, baja)
  - Id\_categoria → id\_categoria.categorias
8. **Proveedor** (id\_proveedor, descripción)
9. **Tipos\_iva** (id\_tipos\_iva, id\_pais, descripción, iva)
  - Id\_pais → id\_pais.países
10. **Precio\_productos** (id\_precio\_producto, id\_producto, id\_proveedor, id\_tipos\_iva, importe, baja)
  - Id\_producto → id\_producto.productos.
  - Id\_proveedor → id\_proveedor.proveedores.
  - Id\_tipos\_iva → id\_tipos\_iva.tipos\_iva.
11. **Lista\_deseos** (id\_deseo, id\_precio\_producto, id\_usuario, d\_alta, regalo\_unico, baja)
  - Id\_precio\_producto → id\_precio\_producto.precio\_productos.
  - Id\_usuario → id\_usuario.usuarios
12. **Usuarios** (id\_usuario, email)
13. **Estados\_regalos** (id\_estado, descripción)
14. **Autómata\_estados** (estado\_inicial, estado\_final, transaccion)
  - Estado\_inicial → id\_estado.estados\_regalos.
  - Estado\_final → id\_estado.estados\_regalos.
15. **Regalos** (id\_regalo, id\_grupo, id\_deseo, id\_estado, d\_modif\_estado, id\_envio)
  - Id\_grupo → id\_grupo.grupo\_compras
  - Id\_deseo → id\_deseo.lista\_deseos

- Id\_estado → id\_estado.estados\_regalos
  - Id\_envio → id\_envio.envios
16. **Participantes** (id\_participante, id\_usuario, id\_grupo, *dedicatoria*, *baja*)
- Id\_usuario → id\_usuario.usuarios.
  - Id\_grupo → id\_grupo.grupo\_compras
17. **Grupo\_compras** (id\_grupo, id\_usu\_homenajead, id\_usu\_receptor\_regalo, id\_celebracion, id\_ubicacion\_entrega, *dedicatoria*, *d\_limite\_fondos*, *d\_recep\_regalo*, *importe\_publico*, *baja*)
- Id\_usu\_homenajead → id\_usuario.usuarios.
  - Id\_usu\_receptor\_regalo → id\_usuario.usuarios.
  - Id\_celebracion → id\_celebracion.tipos\_celebraciones
  - Id\_ubicacion\_entrega → id\_ubicacion.ubicaciones
18. **Tipos\_celebraciones**(id\_celebracion, descripcion)
19. **Pagos**(id\_pago, id\_participante, id\_regalo, id\_tipo\_pago, d\_pago, importe, ref\_pago, id\_pago\_positivo)
- Id\_participante → id\_participante.participantes
  - Id\_regalo → id\_regalo.regalos
  - Id\_tipo\_pago → id\_tipo\_pago.tipos\_pago.
  - Id\_pago\_positivo → id\_pago.pagos.
20. **Tipos\_pago**(id\_tipo\_pago, descripcion)
21. **Logs**(id\_log, d\_hora, procedimiento, *inf\_entrada*, *inf\_salida*, *inf\_rsp*)
22. **Envios**(id\_envio, id\_ubicacion, *ref\_externa*, *d\_entrega*)
- Id\_ubicacion → id\_ubicacion.ubicaciones
23. **Parametros** (descripcion, valor)

Las relaciones que hacen referencia al módulo estadístico se encuentran en el apartado: [6.1.2.- Diseño lógico.](#)

## 2.3.- Diseño físico.

El diseño físico es la última etapa en el diseño de la base de datos. El objetivo general es satisfacer los requerimientos del sistema optimizando la relación costes/beneficios.

Esto se concreta en los siguientes objetivos:

- Disminuir los tiempos de respuesta.
- Minimizar el espacio de almacenamiento.
- Proporcionar la máxima seguridad en la integridad de los datos.

### 2.3.1.- Disminuir el tiempo de respuesta.

Las principales características que inciden en el tiempo de respuesta de un sistema son las siguientes:

- **Creación de claves primarias numéricas:** Es más eficiente la búsqueda de un campo primario definido como numérico que uno de texto, sobre todo, cuando el texto tiene una longitud mucho mayor. Un ejemplo de esta decisión sería como identificamos el “usuario” de la red social. El sistema nos informa del atributo único email pero no es recomendable utilizarlo como clave principal ni por rendimiento ni espacio que se malgasta en el resto de las relaciones.
- **Creación de claves técnicas en entidades asociativas:** Como ya se ha comentado en el diseño lógico, en las relaciones asociativas compuestas por dos o más entidades se ha generado una clave técnica para que resulte más eficiente las consultas con sus tablas relacionadas. Con esta decisión también se aprovecha el espacio ya que las nuevas relaciones con otras entidades solo se relacionan por una clave.
- **Creación de índices en las consultas más utilizadas:** Por defecto Oracle crea los índices de las claves primarias/únicas pero siempre cabe la posibilidad de crear nuevos a partir de los atributos más consultados. La definición de los nuevos índices se suelen generar una vez el desarrollo se ha terminado y así poder analizar las consultas más repetitivas y con un rendimiento insuficiente.
- **Creación de un *tablespace* exclusivo para los índices:** Un *tablespace* es un fichero lógico que utiliza el SGBD para almacenar toda la información que se asocie a él. Una manera de aumentar el rendimiento es crear 2 *tablespaces*, pudiendo guardar en uno toda la información del sistema y en el otro los índices utilizados. Con esta simple separación, el SGBD realizará más eficientemente las consultas.

### 2.3.2.- Minimizar el espacio de almacenamiento.

Indistintamente de cómo se haya diseñado la base de datos, siempre podemos reducir su espacio físico utilizando técnicas de compresión.

Las desventajas de utilizar esta técnica es que el rendimiento del sistema es inferior al tener que comprimir/descomprimir los datos en las acciones de consulta/escritura.

Mientras no existan problemas de espacio en el servidor, recomendamos no utilizar la compresión.

### 2.3.3.- Integridad de los datos.

Una de las características principales de un SGBD es proporcionar los mecanismos necesarios que garanticen la integridad de los datos.

Concretamente las *constraints* nos garantizan que los datos sean únicos a nivel de clave primaria/candidata y que las relaciones entre entidades sean coherentes. Todas las *constraints*

entre tablas se han definido como *restricted*. De esta manera nos aseguramos que no se puedan eliminar registros que intervengan en otras relaciones.

Todas las *constraints* ya han sido identificadas en el modelo lógico (claves primarias/candidatas/foráneas/atributos no nulos) y sería muy repetitivo volverlas a enumerar.

### 3.- Implementación.

Este apartado hace referencia a la fase de implementación donde se detalla el interfaz de los procedimientos.

La instalación del producto se explica en el punto [12.1- Creación de la base de datos](#). La lectura es **obligatoria** para evitar problemas debido a los caracteres especiales y las posibles diferencias entre el juego de caracteres del sistema operativo y la base de datos.

#### 3.1.- Procedimientos.

Todos los requerimientos del cliente han sido implementados a través de procedimientos almacenados en paquetes.

En aquellas gestiones de tablas como es el caso de los productos que no existe un interfaz definido, se ha implementado la lógica utilizando triggers para que el sistema sea coherente. La baja de un producto desencadena la baja de los deseos de los usuarios del sistema y esta a su vez, desencadena la eliminación de los productos en los grupos de compra que se encuentren a medio tramitar.

El criterio para la creación/agrupación de los paquetes ha sido la siguiente:

1. Cada usuario del sistema tiene un paquete con los procedimientos que puede ejecutar. De esta manera queda identificado muy rápidamente las interacciones de cada usuario. Los paquetes que hemos identificado de este tipo son los que se muestran en la imagen de la derecha marcados en rojo.
2. Existen situaciones en las que debemos reaprovechar funcionalidades pero estos no deben influir en la gestión de la transacción. Por este motivo se han creado los paquetes que no están en rojo en la imagen de la derecha.



Imagen 9, distribución procedimientos.

El dividir parte de la lógica en otros procedimientos ha conllevado las siguientes acciones:

- **Gestión de logs:** Todos los procedimientos guardan la información de entrada/salida (exceptuando estructuras multiregistros o listados por pantalla) y el resultado de su ejecución. Para poder asegurar que un error de un procedimiento no cancele los logs del resto, se ha creado una estructura de logs multiregistro que se informa en todos los procedimientos a través de un parámetro IN/OUT. De esta manera el proceso que inicia la acción puede cancelar la transacción en caso de error y guardar todos los logs por donde ha pasado la ejecución del proceso.
- **Gestión de errores:** Cuando un procedimiento genera un error este lo informa a quien lo ha invocado y así sucesivamente hasta llegar al procedimiento que gestiona la transacción. Es por este motivo que a la hora de definir los errores controlados de un procedimiento podemos ver que en algunos casos se haga referencia a parámetros que el procedimiento inicial no son visibles. Estas excepciones no existen en el procedimiento inicial pero se han documentado para mostrar todos los errores.

Como veremos en las especificaciones de los procedimientos, hay parámetros de entrada que se alimentan de otros procedimientos de consulta. Para poder recoger correctamente los valores, los procedimientos de consulta tienen 2 interfaces diferentes utilizando la sobrecarga de métodos.

El primer interfaz, muestra por pantalla los listados y el segundo devuelve el conjunto de registros en una estructura de datos definida para que el cliente pueda recorrerla.

Para no repetir los procedimientos que tengan doble interfaz, se detallarán los que sean de tipo listado en el anexo: [12.2.- Procedimientos con doble interfaz.](#)

En los siguientes puntos se detallan las características más importantes (interfaz, información de entrada/salida, errores, etc) de los procedimientos agrupados por usuarios.

### 3.1.1.- Homenajeado.

El usuario homenajeado es el primer usuario que interactúa con el sistema (exceptuando a los proveedores que deben introducir el catálogo de productos).

En las siguientes tablas se detalla los procedimientos:

Procedimiento	pkg_homenajeado.ListadoCategorias		
<b>Funcionalidad</b>	Devuelve una estructura de datos con todas las categorías dadas de alta.		
<b>Parámetros de entrada</b>			
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_categorias( id_categoria, descripción, num_productos, num_prod_prov,	Table( number, varchar2, number, number,	Identificador único de la categoría Descripción de la categoría Número de productos asociados Número de productos a la venta

	id_categoria_superior, varchar2, desc_cat_superior) varchar2)	Identificador de la categoría superior Descripción de la categoría superior
	p_rsp Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>
	exc_no_hay_categorias Others	ERROR: No existen categorías ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_CATEGORIAS TFC.PKG_HOMENAJEADO.T_CATEGORIAS; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.LISTADOCATEGORIAS(P_CATEGORIAS, P_RSP); END;</pre>	

Tabla 26, Procedimiento: pkg\_homenajeado.ListadoCategorias.

Procedimiento	pkg_homenajeado.ListadoProductos		
<b>Funcionalidad</b>	Devuelve una estructura de datos con los productos ofrecidos por los proveedores en función de la categoría informada a partir del procedimiento ListadoCategorias.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_categoria	Number	Identificador que se utilizará para filtrar los productos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_productos( p_id_precio_producto, id_producto, desc_producto, desc_proveedor, desc_pais, pvp, baja)	Tabla( number, number, varchar2, varchar2, varchar2, decimal, varchar2)	Producto ofrecido por un proveedor/país. Identificador único de un producto. Descripción del producto. Descripción del proveedor. País donde lo vende. Precio del producto (impuestos incluidos) Indica si el producto está dado de baja.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_categoria	ERROR: Falta informar el parámetro: p_id_categoria	
	exc_no_productos	ERROR: No existen productos ofrecidos por los proveedores de la categoría: p_id_categoria	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_CATEGORIA NUMBER :=1; P_PRODUCTOS TFC.PKG_HOMENAJEADO.T_LIST_PRODUCTO; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.LISTADOPRODUCTOS(P_ID_CATEGORIA, P_PRODUCTOS); END;</pre>		

Tabla 27, Procedimiento: pkg\_homenajeado.ListadoProductos.

Procedimiento <b>pkg_homenajeado.Altadeseo</b>			
<b>Funcionalidad</b>	El usuario da de alta un producto en su lista de deseos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usuario	Varchar2	Email que identifica al usuario que realiza la acción. Obligatorio.
	p_id_precio_producto	Number	Identificador único de un producto ofrecido por un vendedor. El valor se obtiene del procedimiento "ListadoProductos". Obligatorio.
	p_regalo_unico	Varchar2	Indica si desea recibir una o varias veces el mismo regalo. Valores de entrada (T/F). Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_deseo	Number	Identificador único que relaciona el producto y usuario en la lista de deseos.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_email		ERROR: Falta informar el email
	exc_no_regalo_unico		ERROR: Falta informar el parámetro: p_regalo_unico con los valores {T/F}
	exc_no_inf_producto		ERROR: Falta informar el parámetro: p_id_precio_producto
	exc_no_existe_producto		ERROR: El id_precio_producto informado:p_id_precio_producto no existe
	exc_ya_existe_producto		ERROR: Ya existe el producto en la lista de deseos.
	exc_producto_baja		ERROR: El producto está dado de baja en el catálogo del proveedor
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USUARIO VARCHAR2(100):='juan@gmail.com'; P_ID_PRECIO_PRODUCTO NUMBER:=1; P_REGALO_UNICO CHAR(1):='F'; P_ID_DESEO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.ALTADESEO(P_EMAIL_USUARIO, P_ID_PRECIO_PRODUCTO, P_REGALO_UNICO, P_ID_DESEO, P_RSP); END;</pre>		

Tabla 28, Procedimiento: **pkg\_homenajeado.Altadeseo**.

Procedimiento <b>pkg_homenajeado.ListadoDeseos</b>			
<b>Funcionalidad</b>	Devuelve una estructura con los deseos que el usuario ha dado de alta.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usuario	Varchar2	Identifica al usuario que queremos saber la lista de deseos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_deseos( id_deseo, desc_producto,	Tabla( Number, Varchar2,	Id que relaciona un producto/usuario. Nombre del producto.

	d_alta, desc_proveedor, desc_pais, pvp, regalo_unico, num_otros_grupos)	Fecha, Varchar2, Varchar2, Decimal, Varchar2, smallint)	Fecha de alta del deseo. Nombre del proveedor. Nombre del país donde se vende el producto. Precio del producto (impuestos incluidos). Indica si desea recibir más de una vez el regalo. Indica el número de grupos donde consta el deseo.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email		ERROR: Falta informar el email
	exc_usu_no_regalos		ERROR: No existen lista de deseos del usuario:'    p_email_usuario
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USUARIO VARCHAR2(100) := 'juan@gmail.com'; P_DESEOS TFC.PKG_HOMENAJEADO.T_LIST_DESEOS; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.LISTADODESEOS(P_EMAIL_USUARIO,P_DESEOS,P_RSP); END;</pre>		

Tabla 29, Procedimiento: pkg\_homenajeado.ListadoDeseos.

pkg_homenajeado.ModificaDeseo			
<b>Procedimiento</b>			
<b>Funcionalidad</b>	La única característica que podemos modificar de un deseo es la propiedad de regalo único. Si se cambia la opción a regalo único, se realizará una búsqueda en los grupos de compra para cancelar el exceso de regalos, dejando solo una unidad que tenga más dinero recaudado (siempre y cuando no existan regalos tramitados, en caso contrario, cancela todos los regalos). El atributo p_id_deseo se obtiene de ListadoDeseos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usuario	Varchar2	Identifica al usuario. Obligatorio.
	p_id_deseo	Number	Identifica el deseo que queremos modificar. Obligatorio.
	p_regalo_unico	Varchar2	El atributo que podemos modificar, cambiando su valor de T/F. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email		ERROR: Falta informar el email
	exc_no_id_deseo		ERROR: Falta informar el parámetro: p_id_deseo
	exc_no_regalo_unico		ERROR: El usuario y id_regalo informado no se corresponden:    p_email_usuario     -    p_id_deseo
	exc_no_existe_registro		ERROR: No existe el id_deseo informado:    p_id_deseo

	exc_no_existe_deseo	ERROR: No existe el id_deseo informado:    p_id_deseo
	exc_deseo_baja	ERROR: El deseo está de baja
	Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USUARIO VARCHAR2(100) := 'juan@gmail.com'; P_ID_DESEO NUMBER:= 1; P_REGALO_UNICO CHAR(1) := 'T'; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.MODIFICADESEO(P_EMAIL_USUARIO, P_ID_DESEO, P_REGALO_UNICO, P_RSP) END; </pre>	

Tabla 30, Procedimiento: pkg\_homenajeado.ModificaDeseo.

Procedimiento	pkg_homenajeado.BajaDeseo		
<b>Funcionalidad</b>	Damos de baja un el deseo de nuestra lista. Al realizar la baja implica que en los grupos de compra que se encuentre este deseo, se les cambiará el estado a eliminados si se el regalo no ha sido tramitado por el proveedor. El atributo p_id_deseo se obtiene de ListadoDeseos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usuario	Varchar2	Identifica al usuario. Obligatorio.
	p_id_deseo	Number	Identifica el deseo que queremos dar de baja. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email	ERROR: Falta informar el email	
	exc_no_id_deseo	ERROR: Falta informar el parámetro: p_id_deseo	
	exc_no_existe_registro	ERROR: El usuario y id_regalo informado no se corresponden:    p_email_usuario     -    p_id_deseo	
	exc_deseo_baja	ERROR: El deseo ya está de baja	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USUARIO VARCHAR2(100) := 'juan@gmail.com'; P_ID_DESEO NUMBER:= 1; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.BAJADESEO(P_EMAIL_USUARIO, P_ID_DESEO, P_RSP); END; </pre>		

Tabla 31, Procedimiento: pkg\_homenajeado.BajaDeseo.

Procedimiento	pkg_homenajeado.ListadoRegalosRecibidos		
<b>Funcionalidad</b>	Listado por pantalla de todos los envíos recibidos con el siguiente desglose: <ul style="list-style-type: none"> <li>Datos del envío: Fecha entrega, usuario receptor, REF. Empresa Transporte, Dirección, Municipio, Provincia/País.</li> <li>Datos del producto: Producto, Proveedor, País comprado, precio</li> <li>Datos del participante: Dedicatoria, Participante, Importe (opcional en función si el precio es público o no.)</li> </ul>		
<b>Parámetros de</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>

<b>entrada</b>	p_email_usuario	Varchar2	Identifica al usuario. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email	ERROR: Falta informar el email	
	exc_no_regalos_entregados	ERROR: No existe regalos entregados para el usuario:     p_email_usuario.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USUARIO VARCHAR2(100) := 'juan@gmail.com'; P_RSP VARCHAR2(1000); BEGIN PKG_HOMENAJEADO.LISTADOREGALOSRECIBIDOS(P_EMAIL_USUARIO, P_RSP); END;</pre>		

Tabla 32, Procedimiento: pkg\_homenajeado.ListadoRegalosRecibidos.

### 3.1.2.- Amigos.

Las acciones de los amigos es gestionar los grupos de compra y participar en los pagos de los regalos.

En las siguientes tablas se explican el interfaz de los procedimientos:

<b>Procedimiento</b>	pkg_amigos.AltaGrupo		
<b>Funcionalidad</b>	Da de alta un grupo de compra.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_homenajeado	Varchar2	Usuario al que queremos regalarle los regalos. Obligatorio.
	p_usu_receptor_regalo	Varchar2	Usuario receptor del regalo. Obligatorio.
	p_desc_tipo_celebracion	Varchar2	Descripción del tipo de celebración (Cumpleaños, Jubilación,..).Obligatorio.
	p_desc_municipio	Varchar2	Nombre del municipio del envío. Obligatorio.
	p_desc_via	Varchar2	Tipo de dirección (calle, avenida, plaza...). Obligatorio.
	p_direccion	number	Dirección de envío. Obligatorio.
	p_numero	Number	Número de la dirección. Obligatorio.
	p_piso	Number	Piso de la dirección
	p_puerta	Varchar2	Puerta de la dirección
	p_cp	Varchar2	Código postal
	p_dedicatoria	Varchar2	Dedicatoria general del grupo. Obligatorio.
	p_fecha_limite_fondos	date	Fecha límite para recaudar el dinero de los regalos. Obligatorio.
	p_fecha_recep_regalo	Date	Fecha en la que los regalos se tienen que ser recibidos. Obligatorio.
p_importe_publico	Varchar2	Indica si el importe de los participantes es público. (T/F). Obligatorio.	
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>

<b>salida</b>	p_id_grupo	Number	Identificar único del grupo creado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email	ERROR: Falta informar el email	
	exc_incorrecto_tip_celebracion	ERROR: No existe la celebración: p_desc_tipo_celebracion	
	exc_no_f_fondos	ERROR: Falta informar el parámetro p_fecha_limite_fondos	
	exc_f_fondos_pasado	ERROR: La fecha límite recogida de fondos no puede ser inferior al día de hoy.	
	exc_no_importe_publico	ERROR: El valor informado en p_importe_publico: p_importe_publico es incorrecto, tiene que ser T/F	
	exc_no_f_recep	ERROR: Falta informar el parámetro: p_fecha_recep_regalo	
	exc_fondos_mayor_recepcion	ERROR: La fecha recogida de fondos (p_fecha_limite_fondos) es mayor que la fecha de recepción (p_fecha_recep_regalo)	
	exc_f_recep_superior_max	ERROR: El intervalo máximo entre la fecha de recogida de fondos y la de recepción no puede ser superior a: v_dias días'	
	exc_no_dedicatoria	ERROR: Falta informar el parámetro p_dedicatoria	
	exc_piso_negativo	ERROR: El valor del piso tiene que ser un valor positivo mayor que 0	
	exc_incorrecto_numero	ERROR: No se ha informado correctamente el número de la calle	
	exc_no_direccion	ERROR: Falta informar el parámetro: p_dirección.	
	exc_incorrecto_via	ERROR: No existe la vía: p_desc_via	
exc_incorrecto_municipio	ERROR: No existe el municipio:p_desc_municipio		
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_HOMENAJEADO VARCHAR2(100) := 'juan@gmail.com'; P_USU_RECEPTOR_REGALO VARCHAR2(100) := 'juan@gmail.com'; P_DESC_TIPO_CELEBRACION VARCHAR2(100) := 'Cumpleaños'; P_DESC_MUNICIPIO VARCHAR2(50) := 'Barcelona'; P_DESC_VIA VARCHAR2(50) := 'Calle'; P_DIRECCION VARCHAR2(50) := 'La rasa'; P_NUMERO NUMBER:=1; P_PISO NUMBER:= 1; P_PUERTA VARCHAR2(5) :=2; P_CP VARCHAR2(5) := 08225; P_DEDICATORIA VARCHAR2(200) :='Felicidades!!!! Ya son 54 años!!!'; P_FECHA_LIMITE_FONDOS DATE:= to_date('21052014','ddmmrrrr'); P_FECHA_RECEP_REGALO DATE:= to_date('29052014','ddmmrrrr'); P_IMPORTE_PUBLICO CHAR(1) :='T'; P_ID_GRUPO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.ALTAGRUPO(P_EMAIL_HOMENAJEADO, P_USU_RECEPTOR_REGALO, P_DESC_TIPO_CELEBRACION, P_DESC_MUNICIPIO, P_DESC_VIA, P_DIRECCION, P_NUMERO, P_PISO, P_PUERTA,</pre>		

P\_CP, P\_DEDICATORIA, P\_FECHA\_LIMITE\_FONDOS, P\_FECHA\_RECEP\_REGALO, P\_IMPORTE\_PUBLICO, P\_ID\_GRUPO, P\_RSP);  
END;

Tabla 33, Procedimiento: pkg\_amigos. AltaGrupo.

Procedimiento	pkg_amigos.ListadoGrupos		
<b>Funcionalidad</b>	<p>Genera una estructura de salida con los grupos que contienen regalos pendientes de alcanzar la recaudación.</p> <p>En el caso de querer saber el grupo de una persona, se puede informar del email del usuario y devolverá los grupos donde conste como homenajeados.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usu_homenajeados	Varchar2	Email utilizado para filtrar por los grupos en los cuales consta. Si no se informa devolverá los grupos con regalos pendientes de completar la recaudación
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_t_grupos( id_grupo, usu_homenajeados, desc_celebracion, d_limite_fondos, sum_importe_regalos, importe_restante, porcentaje_completado baja	Tabla( Number Varchar Varchar2 Fecha Decimal Decimal Decimal Varchar2)	Identificador único del grupo. Email de la persona homenajeados. Descripción del tipo de celebración. Fecha límite de fondos. Sumatorio de los importes de los regalos. Sumatorio del importe que falta. Porcentaje completado de pagos. Indica si el grupo está dado de baja (Si,No)
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_hay_grupos		ERROR: No existen grupos que cumplan con las condiciones
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EMAIL_USU_HOMENAJEADO VARCHAR2(100) := 'juan@gmail.com'; P_T_GRUPOS TFC.PKG_AMIGOS.T_LIST_GRUPOS; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.CONSLTAGRUPOS(P_EMAIL_USU_HOMENAJEADO, P_T_GRUPOS,P_RSP); END;</pre>		

Tabla 34, Procedimiento: pkg\_amigos.ListadoGrupos.

Procedimiento	pkg_amigos.BajaGrupo.		
<b>Funcionalidad</b>	<p>Realiza una baja del grupo de compra extendiendo sus efectos a:</p> <ul style="list-style-type: none"> <li>• Participantes.</li> <li>• Regalos no completados (se aplica devolución)</li> </ul>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que queremos dar de baja. Obligatorio. El valor se obtiene del ListadoGrupos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado

		correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo
	exc_no_existe_grupo	ERROR: El id_grupo informado no existe.
	Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_GRUPO NUMBER:= 1;   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.BAJAGRUPPO(P_ID_GRUPO,P_RSP); END;</pre>	

Tabla 35, Procedimiento: pkg\_amigos.BajaGrupo.

<b>Procedimiento</b>	<b>pkg_amigos.AltParticipante.</b>		
<b>Funcionalidad</b>	Da de alta a un usuario en un grupo de compras.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que queremos dar de alta. Obligatorio. El valor se obtiene del ListadoGrupos. Obligatorio.
	p_email_participante	Varchar2	Email de la persona que se quiere unir al grupo de compra. Obligatorio.
	p_dedicatoria	Varchar2	Dedicatoria personalizada.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email	ERROR: Falta informar el email	
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo	
	exc_no_existe_grupo	ERROR: El id_grupo informado no existe.	
	exc_usu_ya_existe	ERROR: El participante (p_email_participante) ya existe en el grupo (p_id_grupo)	
	exc_participante_homenajead	ERROR: No se puede añadir como participante del grupo de compra al usuario homenajead.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_GRUPO NUMBER:= 1;   P_EMAIL_PARTICIPANTE VARCHAR2(100) := 'juan@gmail.com';   P_DEDICATORIA VARCHAR2(150) := 'Felicidades Paco!!!!!!';   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.ALTAPARTICIPANTE(P_ID_GRUPO, P_EMAIL_PARTICIPANTE, P_DEDICATORIA, P_RSP); END;</pre>		

Tabla 36, Procedimiento: pkg\_amigos.AltParticipante.

Procedimiento <b>pkg_amigos.BajaParticipante.</b>			
<b>Funcionalidad</b>	Marca de baja a un participante que tiene las siguientes consecuencias: <ul style="list-style-type: none"> <li>• Aplica la devolución en los regalos que haya participado y no estén tramitados por el proveedor.</li> </ul>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que nos queremos dar de baja. Obligatorio. El valor se obtiene del ListadoGrupos. Obligatorio.
	p_email_participante	Varchar2	Email de la persona que se quiere dar de baja. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_email	ERROR: Falta informar el email	
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo	
	exc_no_existe_grupo	ERROR: El id_grupo informado no existe.	
	exc_usu_no_existe	ERROR: El participante (p_email_participante) no está dado de alta en el grupo (p_id_grupo)	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_GRUPO NUMBER:= 1;   P_EMAIL_PARTICIPANTE VARCHAR2(100) := 'juan@gmail.com';   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.BAJAPARTICIPANTE(P_ID_GRUPO, P_EMAIL_PARTICIPANTE,P_RSP); END;</pre>		

Tabla 37, Procedimiento: **pkg\_amigos.BajaParticipante.**

Procedimiento <b>pkg_amigos. AltaRegalo</b>			
<b>Funcionalidad</b>	Da de alta un regalo en un grupo de compra. El regalo tiene que existir en la lista de deseos del usuario homenajeado y si el regalo está definido como único, comprobará que no se haya tramitado en otro grupo de compra.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que queremos dar de alta el regalo. Obligatorio. El valor se obtiene del ListadoGrupos. Obligatorio.
	p_id_deseo	Number	Identifica un producto que desea el homenajeado. El valor se obtiene del ListadoDeseso. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	P_id_regalo	Number	Identificador único que asocia un id_deseo y un id_grupo.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_deseo	ERROR: El falta informar el parámetro:	

	p_id_dese
exc_no_id_grupo	ERROR: El falta informar el parámetro: p_id_grupo
exc_grupo_baja	ERROR: El id_grupo: p_id_grupo está dado de baja
exc_no_corresponden_datos	ERROR: No se corresponden los datos p_id_deseo: p_id_deseo \ p_id_grupo:p_id_grupo
exc_regalo_unico_ya_iniciado	ERROR: El regalo no se puede dar de alta porque es único y ya existe en otros grupos de compra.
exc_ya_existe	ERROR: El regalo ya está incluido en el grupo.
Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER:= 1; P_ID_DESEO NUMBER:= 1; P_ID_REGALO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.ALTARegALO(P_ID_GRUPO, P_ID_DESEO, P_ID_REGALO, P_RSP); END;</pre>

Tabla 38, Procedimiento: pkg\_amigos.Altaregalo.

<b>Procedimiento</b>	<b>pkg_amigos.ListadoRegalosGrupo.</b>		
<b>Funcionalidad</b>	Devuelve una estructura con la información más representativa de los regalos de un grupo de compra. Si el grupo está de baja también devuelve los regalos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que queremos consultar los regalos. El valor se obtiene del ListadoGrupos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_t_regalos( id_regalo, desc_producto, desc_estado, importe, importe_restante)	Tabla( Number, Varchar2, Varchar2, Decimal, Decimal)	Identificador único del regalo. Descripción del producto. Descripción del estado del regalo. Importe del producto. Cantidad pendiente de pago, utilizado para saber hasta que importe se puede participar en una aportación.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_grupo	ERROR: El falta informar el parámetro: p_id_grupo	
	exc_no_hay_regalos	ERROR: No existen regalos a partir del id_grupo:p_id_grupo	
	Others	ERROR: +sqlcode:+sqlerrm	

<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER:= 1; P_T_REGALOS TFC.PKG_AMIGOS.T_LIST_REGALOS_GRUPO; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.LISTADOREGALOSGRUPO(P_ID_GRUPO, P_T_REGALOS, P_RSP); END;</pre>
------------------------------	--

Tabla 39, Procedimiento: pkg\_amigos.ListadoRegalosGrupo.

Procedimiento	pkg_amigos.BajaRegalo		
<b>Funcionalidad</b>	<p>Da de baja un regalo en un grupo de compra. El estado del regalo tiene que estar en "Añadido a un grupo" o "Pago completado".</p> <p>Dar de baja un regalo implica realizar la devolución de los importes a los participantes del grupo.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_regalo	Number	Identificador del regalo que queremos dar de baja. El valor se obtiene del procedimiento ListadoRegalosGrupo. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_id_regalo		ERROR: Falta informar el parámetro: p_id_regalo
	exc_no_datos		ERROR: No existe el id_regalo:p_id_regalo
	NO_DATA_FOUND		'ERROR: El regalo(p_id_regalo) no se encuentra en un estado correcto para realizar la transición:p_transaccion.
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_REGALO NUMBER:=1; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.BAJAREGALO(P_ID_REGALO,P_RSP); END;</pre>		

Tabla 40, Procedimiento: pkg\_amigos.BajaRegalo.

Procedimiento	pkg_amigos.AltaPago		
<b>Funcionalidad</b>	<p>Un participante de un grupo de compra, realiza el pago de un regalo que pertenece a ese grupo. Si el regalo llega a su importe, cambiará el estado a "Pago completado" generando movimientos en las estadísticas.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_regalo	Number	Identificador del regalo que pagar. El valor se obtiene del procedimiento ListadoRegalosGrupo. Obligatorio.
	P_email_participante	Varchar2	Identificador de la persona que quiere realizar un pago. Debe pertenecer previamente al grupo de compra. Obligatorio.
	p_desc_tipo_pago	Varchar2	Descripción de la forma de pago utilizada (Paypal, Cuenta bancaria, Tarjeta de

			crédito). Obligatorio.
	P_referencia_pago	Varchar2	Valor que asociado al tipo de pago elegido. (ccc, número de la tarjeta, email de la cuenta de paypal). Obligatorio.
	p_importe	Decimal	Importe del pago. Obligatorio
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_pago	Number	Identificador único del pago realizado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_id_regalo		ERROR: Falta informar el parámetro: p_id_regalo
	exc_no_existe_regalo		ERROR: El id_regalo informado no existe: p_id_regalo
	exc_regalos_estado_incorrecto		ERROR: El regalo (p_id_regalo) no se encuentra en un estado que permita cobros.
	exc_participante_no_grupo		ERROR: El participante ('p_email_participante') no está dado de alta en el grupo de compra del regalo: p_id_regalo
	exc_no_inf_tipo_pago		ERROR: Falta informar el parámetro: p_desc_tipo_pago
	exc_tipo_pago_incorrecto		ERROR: El tipo de pago informado no existe en el sistema.
	exc_no_referencia_pago		ERROR: Falta informar el parámetro: p_referencia_pago
	exc_importe_incorrecto		ERROR: El importe informado (p_importe) tiene que ser superior a 0
	exc_exceso_cobro		ERROR: El importe informado (p_importe) exceder el importe pendiente del regalo que es de: v_precio_final-v_imp_pagado
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_REGALO NUMBER:= 1;   P_EMAIL_PARTICIPANTE NUMBER:='juan@gmail.com';   P_DESC_TIPO_PAGO VARCHAR2(50) :='Paypal';   P_REFERENCIA_PAGO VARCHAR2(75) :=juan@gmail.com;   P_IMPORTE NUMBER:= 10.50;   P_ID_PAGO NUMBER;   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.ALTA_PAGO(P_ID_REGALO, P_EMAIL_PARTICIPANTE, P_DESC_TIPO_PAGO, P_REFERENCIA_PAGO,   P_IMPORTE, P_ID_PAGO, P_RSP); END;</pre>		

Tabla 41, Procedimiento: pkg\_amigos.Altapago.

<b>Procedimiento</b>	pkg_amigos.ListadoPagos.		
<b>Funcionalidad</b>	Genera una estructura de datos de todos los pagos que ha realizado un participante en un grupo de compra.		
<b>Parámetros de</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>

<b>entrada</b>	p_id_grupo	Number	Filtra los pagos a partir del id_grupo informado. Obligatorio
	p_email_participante	Varchar2	Filtra los datos a partir del participante indicado. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_t_pagos( id_pago, desc_producto, tipo_pago, ref_pago, d_pago, importe)	Tabla( Number, Varchar2, Varchar2, Varchar2, Date, Decimal)	identificador único de un pago Producto que ha recibido la aportación. Forma de pago (Paypal,...) Referencia del pago Fecha del pago Importe de la aportación.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo.	
	exc_no_email_part	ERROR: Falta informar el parámetro: p_email_participante	
	exc_no_coinciden_datos_entrada	ERROR: No coinciden los parámetros p_email_participante y p_id_grupo	
	exc_no_hay_pagos	ERROR: No hay pagos del participante:p_email_participante con el id_grupo:p_id_grupo	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER:= 1; P_EMAIL_PARTICIPANTE VARCHAR2(100) :='juan@gmail.com'; P_T_PAGOS TFC.PKG_AMIGOS.T_LIST_PAGOS; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.LISTADOPAGOS(P_ID_GRUPO, P_EMAIL_PARTICIPANTE, P_T_PAGOS, P_RSP); END;</pre>		

Tabla 42, Procedimiento: pkg\_amigos.ListadoPagos.

<b>Procedimiento</b>	<b>pkg_amigos.ModificaPago.</b>		
<b>Funcionalidad</b>	La acción modificar un pago solo permite descender el importe pagado. Internamente lo que realiza es una devolución del pago original y otro pago con la cantidad indicada. Se comprobará que el cobro indicado sea válido (no se haya devuelto anteriormente) y que el regalo que hace referencia la devolución permita esa acción.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_pago_modif	Number	Identificador único del pago que se desea modificar. El valor se obtiene del procedimiento ListadoPagos. Obligatorio.
	p_importe_final	Decimal	Importe final que queremos aportar en el regalo. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_pago	Number	Identificador del pago realizado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado

		correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>
	exc_no_id_pago	ERROR: Falta informar el parámetro: p_id_pago.
	exc_id_pago_incorrecto	ERROR: No existe el id_pago: p_id_pago_modif
	exc_importe_final_incorrecto	ERROR: El importe indicado no es correcto, tiene que ser >0 y <v_importe.
	exc_no_id_pago_correcto	ERROR: El id_pago informado no permite devoluciones.
	exc_regalo_estado_incorrecto	ERROR: El estado del regalo no permite una devolución
	Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_PAGO_MODIF NUMBER:= 1; P_IMPORTE_FINAL NUMBER:= 5.45; P_ID_PAGO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.MODIFICAPAGO(P_ID_PAGO_MODIF, P_IMPORTE_FINAL, P_ID_PAGO, P_RSP); END;</pre>	

Tabla 43, Procedimiento: pkg\_amigos.ModificaPago.

<b>Procedimiento</b>	<b>pkg_amigos.BajaPago.</b>		
<b>Funcionalidad</b>	Realiza un nuevo cobro con todos los atributos del cobro original exceptuando: 1.- El signo del importe (negativo). 2.- La fecha que debe ser la actual. 3.- El atributo "id_pago_positivo" queda guardado el cobro positivo al que hace referencia. De esta manera no se puede devolver 2 veces un mismo cobro.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_pago	Number	Identificador único del pago que se desea devolver. El valor se obtiene del procedimiento ListadoPagos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_pago	ERROR: Falta informar el parámetro: p_id_pago.	
	exc_no_existe_pago	ERROR: El id_pago informado no existe	
	exc_no_id_pago_correcto	ERROR: El id_pago informado no permite devoluciones.	
	exc_regalo_estado_incorrecto	ERROR: El estado del regalo no permite una devolución.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_PAGO NUMBER:= 1; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.MODIFICAPAGO(P_ID_PAGO_MODIF, P_RSP); END;</pre>		

Tabla 44, Procedimiento: pkg\_amigos.BajaPago.

Procedimiento	pkg_amigos.ListadoParticipantesGrupo.		
<b>Funcionalidad</b>	Devuelve una estructura de los participantes de un grupo de compra Se puede filtrar por un grupo concreto o por el usuario y nos saldrá todos los grupos donde consta.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Filtra los datos a partir del id_grupo informado.
	p_participante_email	Varchar2	Email del participante que queremos consultar.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	P_t_part(id_grupo, participante_email, importe, baja)	Table(Number, varchar2, number, Varchar2)	Identificador del grupo de compra, Email del participante, Total importe aportado, Indica si está dado de baja.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_inf_entrada	ERROR: Falta informar uno de los 2 parámetros de entrada (p_id_grupo/p_participante_email)	
	exc_existe_grupo	ERROR: No existe el grupo de compra informado:p_id_grupo	
	exc_no_existe_participante	ERROR: No existen ningún participante con el email:p_participante_email	
	exc_no_datos	ERROR: No existen datos a partir de la información proporcionada.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER; P_PARTICIPANTE_EMAIL VARCHAR2(100); P_T_PART PKG_AMIGOS.T_LIST_PARTICIPANTES_GRUPO; P_RSP VARCHAR2(1000); BEGIN P_ID_GRUPO := 1; PKG_AMIGOS.LISTADOPARTICIPANTESGRUPO( P_ID_GRUPO , P_PARTICIPANTE_EMAIL , P_T_PART ,P_RSP); END;</pre>		

Tabla 45, Procedimiento: pkg\_amigos.ListadoParticipantesGrupo.

### 3.1.3.- Administrador.

El usuario administrador es el encargado de extraer la información más representativa del sistema (listados y las consultas del modelo estadístico).

En las siguientes tablas se explican el interfaz de los procedimientos:

<b>Procedimiento</b> pkg_administrador.ListadoTopTenRegalosCiudades.			
<b>Funcionalidad</b>	Listado por pantalla de las 10 ciudades que más regalos han recibido en un año concreto. De cada ciudad, se muestra el número de regalos recibidos y la suma total del importe.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran las estadísticas.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	exc_no_regalos		ERROR: No se han entregado regalos para el ejercicio: p_ejercicio
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_RSP VARCHAR2(1000); BEGIN   PKG_ADMINISTRADOR.ListadoTopTenRegalosCiudades(P_RSP); END;</pre>		

Tabla 46, Procedimiento: pkg\_administrador.ListadoTopTenRegalosCiudades.

<b>Procedimiento</b> pkg_administrador.ListadoUsuariosSuperanMedia.			
<b>Funcionalidad</b>	<p>Listado por pantalla de los participantes que superan la media de gasto. Para calcular la media solo se tienen en cuenta los pagos que hagan referencia a los regalos que estén completados.</p> <p>No existe un intervalo de tiempo definido, con lo cual, tendrá en cuenta la media de todos los ejercicios.</p> <p>El listado indica la media global en la cabecera y como detalle, el email del usuario y su media</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_datos		ERROR: No se han realizado pagos en el sistema, con lo cual, no hay participantes que superen la media de gasto.
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_RSP VARCHAR2(1000); BEGIN   PKG_ADMINISTRADOR.LISTADOUSUARIOSSUPERANMEDIA(P_RSP); END;</pre>		

Tabla 47, Procedimiento: pkg\_administrador.ListadoUsuariosSuperanMedia.

Los procedimientos que hacen referencia al módulo estadístico se explican en el punto: [6.2.- Procedimientos.](#)

### 3.1.4.- Proveedor.

El proveedor interviene en el sistema para procesar los regalos completados e informar de la referencia del envío de la empresa de transportes como se puede ver en la imagen 6.

En las siguientes tablas se explican el interfaz de los procedimientos:

Procedimiento pkg_proveedor.ProcesarPedido.			
<b>Funcionalidad</b>	<p>Devuelve una estructura con todos los envíos que tiene que procesar un proveedor de los grupos que tienen los regalos en estado pagado y ha llegado el día límite de recaudación.</p> <p>El primer nivel de la estructura hace referencia a los datos que identifican el destinatario: (dirección, fecha de entrega, email). El segundo nivel de la estructura hace referencia a los múltiples productos del envío.</p> <p>Al ejecutarse el listado, realiza el cambio de estados de los productos para que no se puedan devolver o eliminar, con lo cual, este procedimiento no se debe de interpretar con un listado de consulta.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_desc_proveedor	Varchar2	Nombre del proveedor. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_pedidos( id_envio, d_recepcion, usuario_receptor, adr1, adr2, adr3, productos( id_precio_producto, desc_producto, desc_pais, precio))	Tabla( Number, Date, Varchar2, Varchar2, Varchar2, Varchar2, Tabla( Number, Varchar2, Varchar2, Decimal))	Identificador único del envío. Fecha a cumplir en el envío. Email de la persona receptora. Dirección (Dirección, número, piso, puerta) Dirección (Código postal, municipio) Dirección (Provincia, país) Identificador del producto/proveedor/país. Descripción del producto. Descripción del país donde se vende. Precio.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_desc_proveedor	ERROR: Falta informar el parámetro: p_desc_proveedor.	
	exc_no_proveedor	ERROR: No existe el proveedor: p_desc_proveedor	
	exc_no_datos	ERROR: No existe pedidos para procesar para el proveedor: p_desc_proveedor	

	Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_DESC_PROVEEDOR VARCHAR2(100) := 'Amazon'; P_PEDIDOS TFC.PKG_PROVEEDOR.T_LIST_PEDIDOS; P_RSP VARCHAR2(1000); BEGIN PKG_PROVEEDOR.PROCESAPEDIDO(P_DESC_PROVEEDOR, P_PEDIDOS, P_RSP); END; </pre>	

Tabla 48, Procedimiento: pkg\_proveedor.ProcesarPedido.

Procedimiento	pkg_proveedor.InformaEnvio.		
<b>Funcionalidad</b>	<p>El proveedor tiene la responsabilidad de informarnos del número de seguimiento de la empresa de transportes cuando ha terminado de procesar el pedido que previamente ha obtenido con el procedimiento "ProcesarPedido".</p> <p>Cuando nos informa del número de seguimiento que tiene un envío, todos los regalos que estén asociados al envío cambiarán su estado a "Enviado".</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_envio	number	Identificador único de un envío. Obligatorio.
	p_referencia_envio	Varchar2	Número de seguimiento de la empresa de transportes. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_envio	ERROR: Falta informar el parámetro p_id_envio.	
	exc_no_referencia_envio	ERROR: Falta informar el parámetro p_referencia_envio.	
	exc_no_datos	ERROR: No existen registros a partir del id_envio:p_id_envio	
	exc_no_estado_correcto	ERROR: El envío informado (p_id_envio) no se encuentra en un estado correcto.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_ENVIO NUMBER:=1; P_REFERENCIA_ENVIO VARCHAR2(50) :='123456'; P_RSP VARCHAR2(1000); BEGIN PKG_PROVEEDOR.INFORMAENVIO(P_ID_ENVIO, P_REFERENCIA_ENVIO, P_RSP); END; </pre>		

Tabla 49, Procedimiento: pkg\_proveedor.InformaEnvio.

### 3.1.5.- Empresa transportes.

La empresa de transportes es la encargada de comunicarnos la fecha de entrega de un pedido como podemos observar en la imagen 7.

En la siguiente tabla se explica el interfaz del procedimiento:

Procedimiento <b>pkg_repartidor.InformarEntrega</b>			
<b>Funcionalidad</b>	<p>Informa la fecha de entrega de un envío a partir de la referencia de envío de la empresa de transportes.</p> <p>Esta acción comporta un cambio de estado a "Recibido" de los regalos que forman parte del envío.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_referencia_envio	number	Número de seguimiento de la empresa de transportes. Obligatorio.
	p_fecha_entrega	Date	Fecha en la que se entrega el pedido. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_ref_externa		ERROR: Falta informar el parámetro: p_referencia_envio.
	exc_no_fecha_entrega		ERROR: Falta informar el parámetro: p_fecha_entrega.
	exc_fecha_fuera_plazo		ERROR: La fecha de entrega informada (p_fecha_entrega) se considera incorrecta al superar la desviación máxima respecto la fecha propuesta de entrega (v_d_recep_regalo).
	exc_no_estado_correcto		ERROR: Los regalos que hacen referencia al envío (p_referencia_envio) no se encuentra en un estado correcto.
	exc_no_datos		ERROR: No existe envíos con la referencia de envío: p_referencia_envio
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_REFERENCIA_ENVIO VARCHAR2(50) := '123456';   P_FECHA_ENTREGA DATE:= to_date('05052014','DDMMRRR');   P_RSP VARCHAR2(1000); BEGIN   PKG_REPARTIDOR.INFORMARENREGA(P_REFERENCIA_ENVIO,P_FECHA_ENTREGA, P_RSP); END;</pre>		

Tabla 50, Procedimiento: **pkg\_repartidor.InformarEntrega**.

### 3.1.6.- Procesos Batch.

Existe un proceso diario que da de baja los grupos que han superado la fecha de recaudación.

En la siguiente tabla se explica el interfaz del procedimiento:

Procedimiento <b>pkg_procesos_batch.BajaGruposFechaLimite.</b>	
<b>Funcionalidad</b>	<p>Proceso batch que se ejecuta todos los días, seleccionado los grupos de compra que han superado la fecha límite de recaudación.</p> <p>Los grupos seleccionados se darán de baja para que los participantes no puedan seguir operando porque ya no tiene sentido.</p>

	El resto de actores (proveedor y repartidor) podrán seguir evolucionando el estado de los regalos. Los regalos que han quedado pendiente de recaudación, cambiarán su estado a "Sin fondos suficientes" para poderlos diferenciar de una baja por decisión de los amigos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	p_num_grupos	Number	Indica el número de grupos que ha dado de baja automáticamente.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo	
	exc_no_existe_grupo	ERROR: El id_grupo informado no existe.	
	exc_grupo_baja	ERROR: El id_grupo informado ya consta de baja.	
	exc_no_p_transaccion	ERROR: Falta informar correctamente el parámetro p_transaccion: p_transaccion	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_NUM_GRUPOS NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_PROCESOS_BATCH.BAJAGRUPOSFECHALIMITE(P_NUM_GRUPOS, P_RSP); END; </pre>		

Tabla 51, Procedimiento: pkg\_procesos\_batch.BajaGruposFechaLimite.

#### 4.- Plan de contingencias.

En la siguiente tabla se identifican los posibles riesgos que pueden suceder a lo largo del proyecto y las medidas necesarias que podemos realizar para minimizarlos.

Riesgo	Probabilidad	Impacto	Contra medida preventiva-correctiva
<b>Enfermedad leve</b>	media	Leve	Vida saludable (dieta, deporte y descanso).
<b>Enfermedad grave</b>	Baja	Crítico	
<b>Pérdida de información</b>	Media	Crítico	Copias de seguridad diarias en diferentes dispositivos (pendrives, dropbox).
<b>Problemas con el equipo</b>	Baja	Medio	Continuación del proyecto en equipos alternativos (portátil).
<b>Problemas conexión Internet</b>	Baja	Leve	Compartir la conexión de internet de otros dispositivos (móvil).
<b>Retrasos acumulados por los riesgos anteriores/planificación incorrecta de los tiempos.</b>	Medio	Crítico	Compensar el déficit aumentando la dedicación diaria y fines de semana. En casos de necesidad extrema, paralización de otros proyectos menos importantes para destinar los recursos al 100%.

Tabla 52, plan de contingencias.

## 5.- Plan de pruebas.

Para asegurar el correcto funcionamiento de la aplicación, se han definido las siguientes pruebas:

- **Pruebas unitarias:** Se centran en comprobar cómo responden los procedimientos cuando no se informan correctamente o se intentan duplicar las acciones. También hay ejemplos de resultados correctos de todos los procedimientos y listados de consultas.
- **Pruebas funcionales:** Justificación de 2 puntos clave en la aplicación: La gestión del deseo único por parte del homenajeado y el correcto funcionamiento de las devoluciones (se comprueban todas las maneras diferentes de provocar una devolución).

Las pruebas del módulo estadísticos se explican en el punto: [6.3.- Plan de pruebas.](#)

Existen 2 maneras de ejecutar las pruebas:

1. Ejecutando el fichero “pruebas\_procedimientos.bat” del fichero de instalación. Este fichero ejecutará todas las consultas con sus respectivos ficheros de log.
2. Ejecutar los procedimientos definidos en el paquete “pkg\_test”. La definición de su interfaz y funcionamiento se explica detalladamente en el anexo: [12.3.- Procedimientos de pruebas del sistema.](#)

## 6.- Módulo estadístico.

En este apartado se explica todo lo relacionado con el módulo estadístico en cada una de sus fases (análisis y diseño, procedimientos y pruebas).

### 6.1.- Análisis y diseño.

Una vez analizado los requerimientos de las estadísticas obtenemos un denominador común. Hay que guardar un conjunto de información en función de **cuando** ocurren las siguientes acciones:

- Un producto cambia su estado a “pago completado”
- Un producto cambia su estado a “entregado”.
- Un producto cambia su estado a “sin fondos suficientes”.

Comentar también que existe la posibilidad de recalcular las estadísticas cuando un regalo en estado “pago completado” realiza una devolución o es eliminado. Ante esta situación hemos creído conveniente que las estadísticas fueran fieles a la realidad y recalcularan los datos sin la contabilización de ese regalo que al final no ha sido comprado.

La gestión de este recalcu lo está definido por un registro en la tabla de parámetros (RECALCULAR\_ESTADISTICAS=T).

En los siguientes apartados se profundiza en su análisis.

### 6.1.1.- Diagrama entidad relación.

Como podemos ver en la siguiente imagen, las entidades enmarcadas de color rojo son las encargadas de guardar todas las estadísticas.

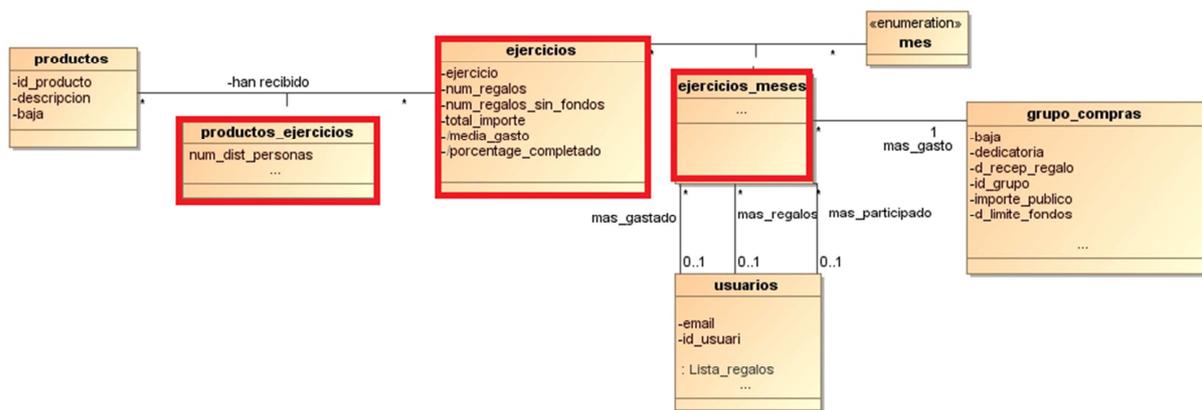


Imagen 10, Diagrama entidad relación del módulo estadístico.

Como se observa en la imagen, las tablas tienen relaciones con otras entidades. Esto es así porque hemos diseñado la base de datos con borrados lógicos y los datos siempre serán consultables.

En el siguiente apartado veremos el detalle de sus atributos y relaciones.

#### 6.1.1.1.- Entidades.

Entidad: Ejercicios.		
<b>Descripción</b>	Entidad que almacena datos estadísticos indexados por años referente al número de regalos, total importe, media de gasto y el porcentaje de regalos completados respecto a los que se han superado el tiempo máximo de recaudación.	
<b>Relaciones</b>	Participa en las entidades asociativa “productos_ejercicios” y “ejercicio_meses”.	
Atributos	Descripción	Ejemplo valor
<b>Ejercicio</b>	Identificador único del ejercicio.	2014
<b>Num_regalos</b>	Contador de regalos comprados en un ejercicio.	135
<b>Total_importe</b>	Sumatorio del importe total en regalos en un ejercicio.	10.365,36
<b>Numero_regalos_sin_fondos</b>	Contador de regalos que no han alcanzado el importe en el plazo indicado por el grupo de compra.	30
<b>Media_gasto</b>	Media del importe de las participaciones. Es un campo	35,80

	calculado a partir de la fórmula: total_importe/num_regalos	
<b>Porcentaje_completado</b>	Muestra el % de los regalos que han llegado al importe respecto a los que no. Es un campo calculado a partir de la fórmula: num_regalos*100/num_regalos+num_regalos_sin_fondos	75

Tabla 53, entidad Ejercicios.

Entidad: Productos_ejercicios		
<b>Descripción</b>	Entidad asociativa que guarda el número de regalos comprados en un año por personas diferentes.	
Atributos	Descripción	Ejemplo valor
<b>Num_dist_personas</b>	Contador que indica el número de veces que un producto se ha vendido a diferentes personas en un año	25

Tabla 54, entidad Productos\_ejercicios.

Entidad: Ejercicios_meses	
<b>Descripción</b>	Entidad asociativa que guarda estadística en función del año y el mes.
<b>Relaciones</b>	2 relaciones con la entidad “usuarios” y una con la entidad “grupos_compras”.

Tabla 55, entidad Ejercicio\_meses.

### 6.1.2.- Diseño lógico.

En la siguiente imagen se observa los atributos de las 3 relaciones del módulo estadístico.

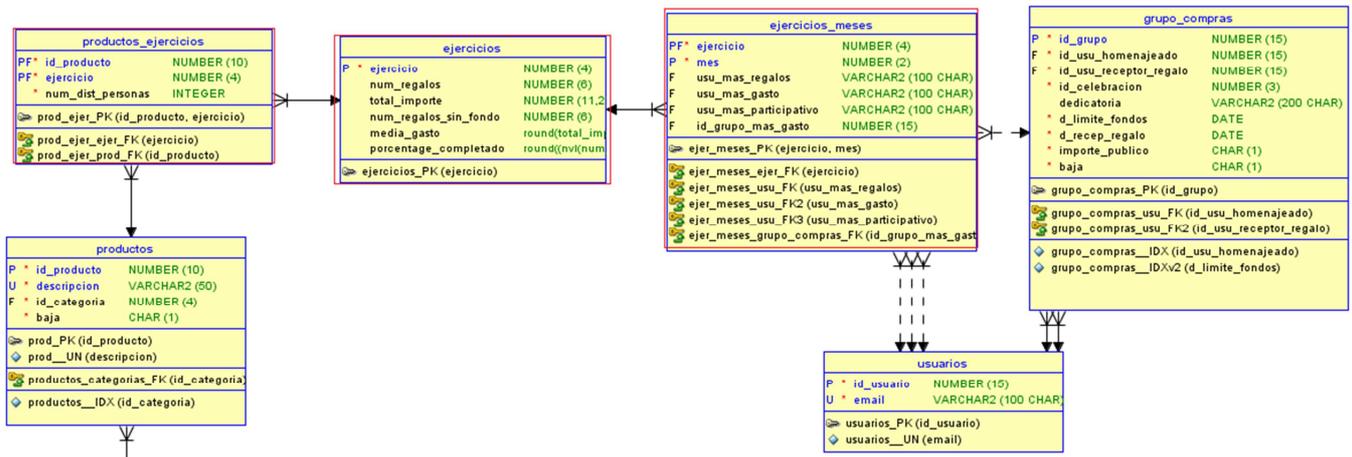


Imagen 11, diseño lógico del módulo estadístico.

Las relaciones del modelo lógico son las siguientes:

- Productos\_ejercicios** (*id\_producto*, *ejercicio*, *num\_distintas\_personas*)
  - id\_producto* → *id\_producto*.productos.
  - ejercicio* → *ejercicio*.ejercicios.

2. **Ejercicios**(*ejercicio, num\_regalos, total\_importe, media\_gasto, porcentaje\_completado*)

3. **Ejercicios\_meses**(*ejercicio, mes, usu\_mas\_regalos, usu\_mas\_gasto, usu\_mas\_participativos, id\_grupo\_mas\_gasto*)

- ejercicio=>ejercicio.ejercicios.
- Usu\_mas\_regalos→id\_usuario.usuarios.
- Usu\_mas\_gasto→id\_usuario.usuarios.
- Usu\_mas\_participativo→id\_usuario.usuarios.
- id\_grupo\_mas\_gasto→ id\_grupo.grupo\_compras.

## 6.2.- Procedimientos.

Los procedimientos que devuelven los datos del modelo estadísticos son los siguientes:

Procedimiento	pkg_administrador.E1_num_productos_año.		
<b>Funcionalidad</b>	Devuelve el número de personas diferentes que han recibido un producto en un ejercicio concreto.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
	p_id_producto	Number	Identificador del producto. El valor se obtiene del listado ListadoProductos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_num_dist_personas	Number	Indica el número de personas distintas que han recibido un producto determinado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio	ERROR: Falta informar el parámetro: p_ejercicio	
	exc_no_id_producto	ERROR: Falta informar el parámetro: p_id_producto.	
	NO_DATA_FOUND	ERROR: No existe registro con la información de entrada: ejercicio:'p_ejercicio id_producto: p_id_producto	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER:= 2014; P_ID_PRODUCTO NUMBER:= 1; P_NUM_DIST_PERSONAS NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E1_NUM_PRODUCTOS_AÑO(P_EJERCICIO, P_ID_PRODUCTO, P_NUM_DIST_PERSONAS,</pre>		

P\_RSP);END;

Tabla 56, Procedimiento: pkg\_administrador.E1\_num\_productos\_año.

Procedimiento	pkg_administrador.E2_num_regalos.		
<b>Funcionalidad</b>	Devuelve el número de regalos que se han comprado en el año que se informa por parámetro. El número de regalos se contabiliza cuando el regalo pase por el estado "Pago completado".		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_num_regalos	Number	Número de regalos que se han comprado en el ejercicio filtrado
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER := 2014; P_NUM_REGALOS NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E2_NUM_REGALOS(P_EJERCICIO, P_NUM_REGALOS, P_RSP); END;</pre>		

Tabla 57, Procedimiento: pkg\_administrador.E2\_num\_regalos.

Procedimiento	pkg_administrador.E3_importe_total_regalos.		
<b>Funcionalidad</b>	Devuelve el importe total gastado en regalos en el año que se informa por parámetro. El importe de los regalos se contabiliza cuando el regalo pase por el estado "Pago completado".		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_importe_regalos	Number	Importe total en regalos que se han gastado en el ejercicio filtrado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio

	Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER:= 2014; P_IMPORTE_REGALOS NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E3_IMPORTE_TOTAL_REGALOS(P_EJERCICIO, P_IMPORTE_REGALOS, P_RSP); END;</pre>	

Tabla 58, Procedimiento: pkg\_administrador.E3\_importe\_total\_regalos.

<b>Procedimiento</b>	<b>pkg_administrador.E4_porcentaje_completado.</b>		
<b>Funcionalidad</b>	Devuelve el porcentaje de los regalos completados, respecto a los que se han quedado sin fondos por superar la fecha máxima de recaudación. Cada vez que los regalos lleguen al estado "Pago_completado" y "Sin fondos suficientes" se actualiza la información.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_porcentaje_completado	Decimal	Porcentaje entre los regalos completados y los que no se han quedado sin fondos por superar la fecha máxima de recaudación.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio	ERROR: Falta informar el parámetro: p_ejercicio	
	NO_DATA_FOUND	ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER:= 2014; P_PORCENTAJE_COMPLETADO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E4_PORCENTAJE_COMPLETADO(P_EJERCICIO, P_PORCENTAJE_COMPLETADO , P_RSP); END;</pre>		

Tabla 59, Procedimiento: pkg\_administrador.E4\_porcentaje\_completado.

<b>Procedimiento</b>	<b>pkg_administrador.E5_media_gasto_regalos.</b>		
<b>Funcionalidad</b>	Devuelve la media del gasto en regalos. Para contabilizarlo, solo se tienen en cuenta los regalos que hayan llegado al estado "pago_completado".		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_media_gasto	Decimal	Gasto medio de los participantes

	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER:= 2014; P_MEDIA_GASTO NUMBER; P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E5_MEDIA_GASTO_REGALOS(P_EJERCICIO, P_MEDIA_GASTO, P_RSP); END;</pre>		

Tabla 60, Procedimiento: pkg\_administrador.E5\_media\_gasto\_regalos.

<b>Procedimiento</b>	<b>pkg_administrador.E6_usuario_mas_regalos.</b>		
<b>Funcionalidad</b>	Devuelve el email del usuario que más regalos ha recibido en un ejercicio/mes. La contabilización de esta estadística se produce en el momento que son entregados los regalos (estado "Recibido").		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
	p_mes	Number	Indica el mes por el cual se van a filtrar los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_usuario	Varchar2	Email del usuario que más regalos ha recibido
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	exc_no_mes		ERROR: Falta informar el parámetro: p_mes.
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio mes:p_mes
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_EJERCICIO NUMBER := 2014; P_MES NUMBER:= 5; P_USUARIO VARCHAR2(100); P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.E6_USUARIO_MAS_REGALOS(P_EJERCICIO, P_MES,P_USUARIO, P_RSP); END;</pre>		

Tabla 61, Procedimiento: pkg\_administrador.E6\_usuario\_mas\_regalos.

Procedimiento <b>pkg_administrador.E7_usuario_mas_gasto.</b>			
<b>Funcionalidad</b>	Devuelve el email del usuario que más gasto ha realizado en un ejercicio/mes. Para contabilizarlo, solo se tienen en cuenta los regalos que hayan llegado al estado "Pago_completado"		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
	p_mes	Number	Indica el mes por el cual se van a filtrar los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_usuario	Varchar2	Email del usuario que más gasto ha realizado.
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	exc_no_mes		ERROR: Falta informar el parámetro: p_mes.
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio mes:p_mes
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_EJERCICIO NUMBER:= 2014;   P_MES NUMBER:= 3;   P_USUARIO VARCHAR2(100);   P_RSP VARCHAR2(1000); BEGIN   PKG_ADMINISTRADOR.E7_USUARIO_MAS_GASTO(P_EJERCICIO, P_MES, P_USUARIO, P_RSP); END; </pre>		

Tabla 62, Procedimiento: **pkg\_administrador.E7\_usuario\_mas\_gasto.**

Procedimiento <b>pkg_administrador.E8_usuario_mas_participativo.</b>			
<b>Funcionalidad</b>	Devuelve el email del usuario que más ha participado en un ejercicio/mes. Para contabilizarlo, solo se tienen en cuenta los regalos que hayan llegado al estado "pago_completado". Los pagos positivos que tengan su correspondiente devolución, no serán contabilizados como "participación".		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
	p_mes	Number	Indica el mes por el cual se van a filtrar los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_usuario	Varchar2	Email del usuario que más gasto ha realizado.

	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	exc_no_mes		ERROR: Falta informar el parámetro: p_mes.
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio mes:p_mes
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_EJERCICIO NUMBER := 2014;   P_MES NUMBER := 4;   P_USUARIO VARCHAR2(100);   P_RSP VARCHAR2(1000); BEGIN   PKG_ADMINISTRADOR.E8_USUARIO_MAS_PARTICIPATIVO(P_EJERCICIO, P_MES,   P_USUARIO, P_RSP); END; </pre>		

Tabla 63, Procedimiento: pkg\_administrador.E8\_usuario\_mas\_participativo.

<b>Procedimiento</b>	<b>pkg_administrador.E9_grupo_compra_mas_gasto.</b>		
<b>Funcionalidad</b>	<p>Devuelve el identificador del grupo de compra (id_grupo) con más gasto realizado en un ejercicio/mes filtrado por parámetro.</p> <p>La contabilización del gasto se realiza cuando los regalos alcanzan el estado "Pago completado"</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_ejercicio	Number	Ejercicio por el cual se filtran los datos. Obligatorio.
	p_mes	Number	Indica el mes por el cual se van a filtrar los datos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	P_id_grupo	Number	Identificador único del grupo de compra con más gasto realizado
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_ejercicio		ERROR: Falta informar el parámetro: p_ejercicio
	exc_no_mes		ERROR: Falta informar el parámetro: p_mes.
	NO_DATA_FOUND		ERROR: No existe registro con la información de entrada: ejercicio: p_ejercicio mes:p_mes
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_EJERCICIO NUMBER:= 2014; </pre>		

```
P_MES NUMBER:= 7;
P_ID_GRUPO NUMBER;
P_RSP VARCHAR2(1000);
BEGIN
PKG_ADMINISTRADOR.E9_GRUPO_COMPRA_MAS_GASTO(P_EJERCICIO, P_MES, P_ID_GRUPO, P_RSP);
END;
```

Tabla 64, Procedimiento: pkg\_administrador.E9\_grupo\_compra\_mas\_gasto.

Procedimiento pkg_administrador.ListadoUsuariosSuperanMedia.			
<b>Funcionalidad</b>	<p>Listado por pantalla de los participantes que superan la media de gasto. Para calcular la media solo se tienen en cuenta los pagos que hagan referencia a los regalos que estén completados.</p> <p>No existe un intervalo de tiempo definido, con lo cual, tendrá en cuenta la media de todos los ejercicios.</p> <p>El listado indica la media global en la cabecera y como detalle, el email del usuario y su media</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_datos	ERROR: No se han realizado pagos en el sistema, con lo cual, no hay participantes que superen la media de gasto.	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre>DECLARE P_RSP VARCHAR2(1000); BEGIN PKG_ADMINISTRADOR.LISTADOUSUARIOSSUPERANMEDIA(P_RSP); END;</pre>		

Tabla 65, Procedimiento: pkg\_administrador.ListadoUsuariosSuperanMedia.

### 6.3.- Plan de pruebas.

El correcto funcionamiento de las estadísticas es uno de los puntos más importantes del proyecto. Para garantizar su correcto funcionamiento se ha construido un procedimiento donde puedes definir las condiciones del sistema.

Una vez definido todos los parámetros, ejecutará las pruebas justificando todas las estadísticas y los listados de consulta más importantes. De esta manera el cliente puede realizar todas las pruebas y situaciones que considere oportunos.

Además de las pruebas que pueda realizar el cliente, también se ha predefinido un conjunto de pruebas preestablecidas con las situaciones más representativas.

Existen 2 maneras de ejecutar las pruebas:

1. Ejecutando el fichero “prueba\_generador\_datos\_estadisticos.bat” del fichero de instalación. Si se desea cambiar los parámetros de la llamada hay que cambiar el fichero “16\_Generador\_datos\_estadisticos.sql”.
2. Ejecutar los procedimientos definidos en el paquete “pkg\_test”. La definición de su interfaz y funcionamiento se explica detalladamente en el anexo: [12.3.1.- Procedimientos de pruebas del módulo estadístico.](#)

## 7.- Valoración económica y recursos utilizados.

Antes de entrar a valorar el coste del proyecto, debemos enumerar los recursos utilizados que se dividen en recursos humanos y recursos técnicos/materiales.

### 7.1.- Recursos humanos.

El grupo de trabajo será formado por los siguientes perfiles profesionales:

- **Jefe de proyectos:** Máximo responsable del proyecto del éxito/fracaso del proyecto. Deberá planificar y coordinar el resto de recursos. Será la única persona que se contacte con el cliente y revisará la información que se le entregue. Interviene principalmente en el plan de trabajo.
- **Analista:** Se encarga del análisis y diseño de la base de datos a partir de los requerimientos del cliente. Define los casos de uso y define el comportamiento de los procedimientos a programar. Interviene principalmente en la fase de análisis y diseño.
- **Programador:** Persona encargada de crear los objetos necesarios en la base de datos y el código de los procedimientos a partir del análisis realizado. Interviene en la fase de implementación y pruebas.

### 7.2.- Recursos técnicos y materiales.

Los integrantes del proyecto utilizarán los siguientes recursos:

- **Hardware:** Ordenador de sobremesa con las principales características son:
  - Procesador Intel I5 2750K quad-core.
  - 16gb memoria DDR3.
  - Disco duro 256GB SSD.
  - Conexión a internet 30/1MB.
- **Software:**
  - Sistema operativo Windows 7 64bits.

- SGBD: Base de datos Oracle 11g donde se implementará el proyecto.
- Microsoft Office 2010: Paquete ofimático para la generación de los informes.
- Microsoft Project 2010: Utilizado para generar el diagrama de Gantt.
- Magic Drawn V17: Implementación de los esquemas de análisis (casos de uso, diagrama entidad-relación).
- Oracle SQL developer: Permite interactuar con la base de datos con un entorno gráfico. Utilizado para la implementación del código.
- Oracle SQL Data Modeler v4.0.1.836: Herramienta para definir el diseño lógico de la base de datos y automatizar la creación de los objetos mediante scripts.

### 7.3.- Coste del proyecto.

A la hora de calcular el coste del proyecto necesitamos saber el tiempo que ha invertido cada profesional en realizar las tareas asignadas.

La siguiente tabla muestra la relación del tiempo/técnico planificado.

Nombre de tarea	Duración	Nombres de los recursos
<b>Inicio proyecto</b>	0 días	
<b>Plan de trabajo (PEC1)</b>	<b>17,5 días</b>	
Lectura requerimientos TFC	3 hrs	Jefe del proyecto
Análisis preliminar del enunciado	5 hrs	Jefe del proyecto
Consultas al cliente	3 hrs	Jefe del proyecto
Consulta otros proyectos	5 hrs	Jefe del proyecto
Lectura redacción de textos	4 hrs	Jefe del proyecto
Instalación software necesario ( MS project)	1 hr	programador
Generar documentación PEC1	10 hrs	Jefe del proyecto
Revisión documentación	4 hrs	Jefe del proyecto
Entrega PEC1	0 hrs	Jefe del proyecto
<b>Análisis y diseño (PEC2)</b>	<b>28 días</b>	
Instalacion software necesario (magic draw uml)	1 hr	programador
Análisis requerimientos	15 hrs	Analista
Preguntas cliente	5 hrs	Analista
Generación casos de uso	3 hrs	Analista
Diseño conceptual	10 hrs	Analista
Diseño lógico	4 hrs	Analista
Diseño físico	4 hrs	Analista
Generar documentación PEC2	10 hrs	Analista
Revisión documentación	4 hrs	Jefe del proyecto
Entrega PEC2	0 días	Analista
<b>Implementación y pruebas (PEC3)</b>	<b>27 días</b>	
Instalación software (Oracle 11/ sql Developer)	3 hrs	programador
Creación entorno (usuario, tablespace,tablas, constraints..)	4 hrs	programador

Creación procedimientos ABM	14 hrs	programador
Creación procedimientos de consulta	4 hrs	programador
Creación procedimientos de estadísticos	5 hrs	programador
Creación procedimiento de logs	5 hrs	programador
Pruebas unitarias	4 hrs	programador
Generar documentación PEC3	10 hrs	programador
Revisar documentación	2 hrs	Jefe del proyecto
Entrega PAC3	0 días	programador
<b>Entrega del proyecto</b>	<b>34 días</b>	
Pruebas finales	10 hrs	programador
Scripts carga de datos y pruebas	10 hrs	programador
Generar memoria	20 hrs	Jefe del proyecto
Generar presentación	20 hrs	Jefe del proyecto
Entrega final (producto, memoria y presentación)	0 días	Jefe del proyecto
<b>Tribunal virtual</b>	<b>3 días</b>	
Defensa del proyecto	3 días	Jefe del proyecto

Tabla 66, relación de tareas/tiempo/recurso.

A partir de la tabla anterior ya podemos agrupar las horas realizadas por los diferentes técnicos como muestra la siguiente tabla:

	Jefe del proyecto	Analista	Programador
<b>Plan de trabajo</b>	34	0	1
<b>Análisis y diseño</b>	4	51	1
<b>Implementación y pruebas</b>	2	0	49
<b>Entrega del proyecto</b>	40	0	10
<b>Total</b>	80	51	61

Tabla 67, resumen de horas por técnicos.

En el sumatorio de horas totales, no se ha contabilizado la última tarea (defensa del proyecto) porque consideramos que no es ético cobrar por justificar el trabajo realizado.

Siguiendo con el cálculo del proyecto, solo nos queda multiplicar las horas de cada técnico por el precio que tiene cada perfil cómo podemos observar en la siguiente tabla:

Recurso	Horas	Precio/hora	Total
<b>Jefe del proyecto</b>	80h	55€	4400€
<b>Analista</b>	51h	40€	2040€
<b>Programador</b>	61h	25€	1525€
		<b>Total</b>	<b>7965€</b>

Tabla 68, coste total del proyecto.

El precio final del proyecto será de **7.965€** al que habrá que sumarle el IVA actual (21%).

Durante los 3 primeros meses de la puesta en producción, las posibles incidencias que se puedan detectar quedan cubiertas por la garantía y no se facturarán.

A partir de esa fecha ya entraría la fase del mantenimiento que incluye la evolución del producto y su mantenimiento.

## 8.- Mejoras futuras.

Las mejoras 2 mejoras que proponemos se explican en los siguientes apartados.

### 8.1.- Cálculo de los gastos de envío.

Existe un aspecto que no se ha comentado como requerimiento del nuevo sistema a crear y que no podemos olvidar que es un parte importante en las compras por internet; los gastos de envío.

Proponemos el siguiente funcionamiento:

- Los gastos de envío se calculan a partir de 2 variables:
  - Distancia del envío: No es lo mismo comprar un producto en tu mismo país que el comprarlo en otro continente.
  - Volumen/peso: Las empresas de transportes facturan más o menos importe en función de las dimensiones y pesos del producto a transportar.
- Promociones: Para promocionar el comercio por internet siempre es recomendable aplicar estímulos como envíos gratuitos cuando superas un importe determinado.

El análisis preliminar de este requerimiento implica ampliar nuevos atributos en las entidades de productos, usuarios, grupos de compra y realizar la programación/adaptación del código.

### 8.2.- Envío de notificaciones de productos afines.

Otra opción para estimular el consumo a través de nuestra red social es avisar a los usuarios de la incorporación de nuevos productos que son afines con sus preferencias.

Proponemos el siguiente funcionamiento:

- Cuando un proveedor introduzca un nuevo producto en el sistema, se enviará un email a todas las personas que tengan productos en su lista de deseos que compartan la misma categoría del producto insertado. De esta manera estamos proponiendo que los usuarios lo añadan a su lista de deseos y esto puede implicar un aumento en las ventas de productos.

Este desarrollo sería muy fácil y rápido ya que el propio sistema ya tiene todos los datos implicados y no hay que ampliar la estructura de la base de datos.

## 9.- Conclusiones.

Una vez acabado el proyecto siento gran satisfacción al ver que cumple con los requerimientos exigidos y además se han incorporado mejoras que algunas de ellas se han considerado indispensables.

Referente a la gestión del proyecto he aprendido lo difícil que resulta cuantificar el tiempo en las tareas, teniendo que ampliar la media de horas hasta el último día.

A nivel técnico he profundizado la programación en pl/sql donde no tenía muchos conocimientos y me irá bien de cara a promocionarme laboralmente.

## 10.- Glosario.

**Diagrama de Gantt:** Herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado.

**Clave primaria:** Campo o a una combinación de campos que identifica de forma única a cada fila de una tabla. Una clave primaria comprende de esta manera una columna o conjunto de columnas. No puede haber dos filas en una tabla que tengan la misma clave primaria.

**Clave candidata:** Una clave candidata es aquella clave que no ha sido seleccionada como clave primaria, pero que también puede identificar de forma única a una fila dentro de una tabla.

**Trigger:** Un trigger (o disparador) en una Base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (INSERT), actualización (UPDATE) o borrado (DELETE). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos.

**Constraints:** Es una condición que obliga el cumplimiento de ciertas condiciones en la base de datos.

**Tablespace:** Un tablespace es una unidad lógica de almacenamiento dentro de una base de datos oracle. Es un puente entre el sistema de ficheros del sistema operativo y la base de datos.

**Uml:** Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

**Pl/sql:** PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación incrustado en Oracle.

**Scripts:** Programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. El uso habitual de los scripts es realizar diversas tareas como combinar componentes,

interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de este tipo de programas.

**Casos de uso:** Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso.

## 11.- Bibliografía.

**Beneito, M; Roser.** “Presentació de documents i elaboració de presentacions”. Material TFC. UOC.  
[Fecha de consulta: 10-03-2014]

<[http://materials.cv.uoc.edu/cdocent/HKZB5UG6XS130\\_6R5O43.pdf?ajax=true](http://materials.cv.uoc.edu/cdocent/HKZB5UG6XS130_6R5O43.pdf?ajax=true)>

**Saézn, H; Nita.** “Redacció de textos científicotècnics”. Material UOC.

[Fecha de consulta: 20-03-2014]

<[http://materials.cv.uoc.edu/cdocent/8QROP4G6IXT6ND3J1\\_XE.pdf?ajax=true](http://materials.cv.uoc.edu/cdocent/8QROP4G6IXT6ND3J1_XE.pdf?ajax=true)>

**Costa, Costa; D.** “Diseny de base de dades”. Material UOC

[Fecha de consulta: 01-04-2014]

<[http://materials.cv.uoc.edu/cdocent/V9\\_BJ89APY\\_48ICQTOF.pdf](http://materials.cv.uoc.edu/cdocent/V9_BJ89APY_48ICQTOF.pdf)>

**Martín, Escofet; Carme.** “El llenguatge SQL” Material UOC.

[Fecha de consulta: 15-04-2014]

<[https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDwQFjAA&url=http%3A%2F%2Focw.uoc.edu%2Finformatica-tecnologia-i-multimedia%2Fbases-de-dades%2FX92IE17U9XAMZXDMDTDDO.odt&ei=W9aSU9OUFpCy7Aa\\_oCoBg&usg=AFQjCNEs-FsXnwFbV3Rc3W\\_w0uSAz90Gog&sig2=bjlk8JXc4grx2MV3W6dgDw](https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0CDwQFjAA&url=http%3A%2F%2Focw.uoc.edu%2Finformatica-tecnologia-i-multimedia%2Fbases-de-dades%2FX92IE17U9XAMZXDMDTDDO.odt&ei=W9aSU9OUFpCy7Aa_oCoBg&usg=AFQjCNEs-FsXnwFbV3Rc3W_w0uSAz90Gog&sig2=bjlk8JXc4grx2MV3W6dgDw)>

**Amatller, Marín; Antoni.** “Exposició de continguts en vídeo”. Material UOC.

[Fecha de consulta: 01-06-2014]

<[http://materials.cv.uoc.edu/continguts/PID\\_00191577/web/main/materias/PID\\_00191569-1.pdf](http://materials.cv.uoc.edu/continguts/PID_00191577/web/main/materias/PID_00191569-1.pdf)>

## 12.- Anexos.

### 12.1- Creación de la base de datos.

La instalación de la base de datos y se ha implementado a partir de scripts.

A la hora de realizar los procesos de carga se ha detectado que existen problemas interpretando los caracteres con acento o el uso de la “ñ”.

Los problemas detectados han sido los siguientes:

- **Mensajes de texto incorrectos:** Muchos de los mensajes de error que pueden devolver los procedimientos utilizan acentos como el siguiente ejemplo: “Falta informar el par&\$·metro: xxxx”.
- **Errores de compilación:** Un procedimiento que participa en toda la aplicación es el procedimiento de log y este se llama: `AñadeLogEstructura`. Como era de esperar, todos los procedimientos daban errores de compilación.
- **Errores de importación:** El error más extraño sucedía al cargar unas vistas que daban error cuando aparentemente no había causa por caracteres especiales.

Después de probar diferentes soluciones al final recomiendo para la instalación del producto cargar los scripts con el **SQL Developer**.

Dentro de las preferencias debemos elegir el tipo de codificación UTF-8 tal y como se muestra en la siguiente imagen:

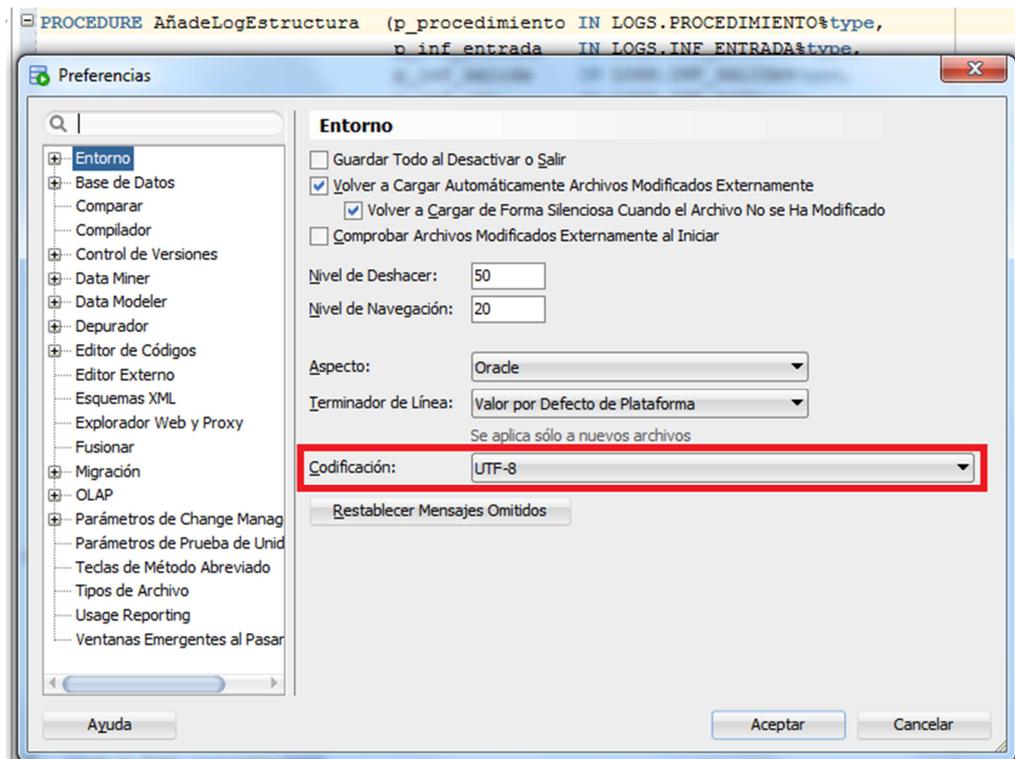


Imagen 12, Configuración correcta del sql developer.

Una vez se ha explicado el incidente, ya podemos pasar a explicar la instalación del producto.

El proceso consta de los siguientes pasos:

1. Descomprimir el fichero “producto.rar” en el directorio de trabajo de Oracle para asegurarnos que tenga permisos de escritura a la hora de generar los ficheros de log. Los ficheros descomprimidos son los que se muestran en la imagen de la derecha.
2. Iniciar el SQL Developer con la codificación “UTF-8”.
3. Introducir las credenciales de un usuario administrador (SYSTEM/SYS).
4. Ejecutar el script **00\_CREAR\_BBDD.sql** en función del directorio donde resida. Para ejecutar un script desde la consola hay que introducir la instrucción: @@ {directorio}\00\_CREAR\_BBDD.sql. Internamente el procedimiento principal ejecuta los scripts desde la creación del tablespace (01\_tablespace.sql) hasta el código del programa (11\_triggers.sql).

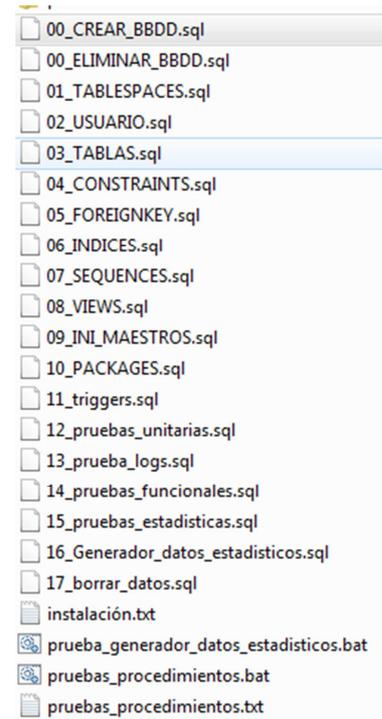


Imagen 13, ficheros del producto.

Con estos 4 pasos tenemos el producto instalado. En el caso de querer desinstalar el producto solo se tendría que ejecutar en el paso 4 el script: 00\_ELIMINAR\_BBDD.sql.

## 12.2.- Procedimientos con doble interfaz.

Procedimiento	pkg_homenajead.ListadoCategorias		
<b>Funcionalidad</b>	Genera un listado por pantalla con las categorías dadas de alta.		
<b>Parámetros de entrada</b>			
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_hay_categorias	ERROR: No existen categorías	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_RSP VARCHAR2(1000); BEGIN   PKG_HOMENAJEADO.LISTADOCATEGORIAS(P_RSP );   DBMS_OUTPUT.PUT_LINE( P_RSP); END;</pre>		

Tabla 69, Procedimiento: pkg\_homenajead.ListadoCategorias.

Procedimiento <b>pkg_homenajeado.ListadoProductos</b>			
<b>Funcionalidad</b>	Genera un listado por pantalla con los productos ofrecidos por los proveedores en función de la categoría informada a partir del procedimiento ListadoCategorias.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_categoria	Number	Identificador para filtrar los productos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_id_categoria		ERROR: Falta informar el parámetro: p_id_categoria
	exc_no_productos		ERROR: No existen productos ofrecidos por los proveedores de la categoría: p_id_categoria
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_CATEGORIA NUMBER :=1;   P_RSP VARCHAR2(1000); BEGIN   PKG_HOMENAJEADO.LISTADOPRODUCTOS( P_ID_CATEGORIA , P_RSP ); END;</pre>		

Tabla 70, Procedimiento: **pkg\_homenajeado.ListadoProductos**.

Procedimiento <b>pkg_homenajeado.ListadoDeseos</b>			
<b>Funcionalidad</b>	Genera un listado por pantalla con los deseos que el usuario ha dado de alta.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usuario	Varchar2	Identifica al usuario que queremos saber la lista de deseos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_email		ERROR: Falta informar el email
	exc_usu_no_regalos		ERROR: No existen lista de deseos del usuario: '  p_email_usuario
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_EMAIL_USUARIO VARCHAR2(100) := 'juan@gmail.com';   P_RSP VARCHAR2(1000); BEGIN   PKG_HOMENAJEADO.LISTADODESEOS(P_EMAIL_USUARIO,P_RSP); END;</pre>		

Tabla 71, Procedimiento: **pkg\_homenajeado.ListadoDeseos**.

Procedimiento <b>pkg_amigos.ListadoGrupos</b>			
<b>Funcionalidad</b>	Genera un listado por pantalla con los grupos que contienen regalos pendientes de alcanzar la recaudación. En el caso de querer saber el grupo de una persona, se puede informar del email del usuario y devolverá los grupos donde conste como homenajeado.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_email_usu_homenajeado	Varchar2	Email utilizado para filtrar por los grupos en los cuales consta. Si no se informa devolverá los grupos con regalos pendientes de completar la recaudación
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_hay_grupos		ERROR: No existen grupos que cumplan con las condiciones
	Others		ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_EMAIL_USU_HOMENAJEADO VARCHAR2(100) := 'juan@gmail.com';   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.CONSLTAGRUPOS(P_EMAIL_USU_HOMENAJEADO, P_RSP); END;</pre>		

Tabla 72, Procedimiento: pkg\_amigos.ListadoGrupos.

Procedimiento <b>pkg_amigos.ListadoRegalosGrupo.</b>			
<b>Funcionalidad</b>	Genera un listado por pantalla con la información más representativa de los regalos de un grupo de compra. Si el grupo está de baja también devuelve los regalos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Identificador del grupo que queremos consultar los regalos. El valor se obtiene del procedimiento ListadoGrupos. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>		<b>Valor devuelto</b>
	exc_no_id_grupo		ERROR: El falta informar el parámetro: p_id_grupo
	exc_no_hay_regalos		ERROR: No existen regalos a partir del id_grupo:p_id_grupo
Others		ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_ID_GRUPO NUMBER:= 1;   P_RSP VARCHAR2(1000); BEGIN   PKG_AMIGOS.LISTADOREGALOSGRUPO(P_ID_GRUPO, P_RSP); END;</pre>		

Tabla 73, Procedimiento: pkg\_amigos.ListadoRegalosGrupo.

<b>Procedimiento</b> pkg_amigos.ListadoPagos.			
<b>Funcionalidad</b>	Genera una listado por pantalla de todos los pagos que ha realizado un participante en un grupo de compra.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Filtra los pagos a partir del id_grupo informado. El valor del grupo se obtiene del ListadoGrupos. Obligatorio
	p_email_participante	Varchar2	Filtra los datos a partir del participante indicado. Obligatorio.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_id_grupo	ERROR: Falta informar el parámetro: p_id_grupo.	
	exc_no_email_part	ERROR: Falta informar el parámetro: p_email_participante	
	exc_no_coinciden_datos_entrada	ERROR: No coinciden los parámetros p_email_participante y p_id_grupo	
	exc_no_hay_pagos	ERROR: No hay pagos del participante:p_email_participante con el id_grupo:p_id_grupo	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER:= 1; P_EMAIL_PARTICIPANTE VARCHAR2(100) :='juan@gmail.com'; P_RSP VARCHAR2(1000); BEGIN PKG_AMIGOS.LISTADOPAGOS(P_ID_GRUPO, P_EMAIL_PARTICIPANTE, P_RSP); END;</pre>		

Tabla 74, Procedimiento: pkg\_amigos.ListadoPagos.

<b>Procedimiento</b> pkg_amigos.ListadoParticipantesGrupo.			
<b>Funcionalidad</b>	Genera un listado por pantalla de los participantes de un grupo de compra. Se puede filtrar por un grupo concreto o por el usuario y nos saldrá todos los grupos donde consta.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_id_grupo	Number	Filtra los datos a partir del id_grupo informado.
	p_participante_email	Varchar2	Email del participante que queremos consultar.
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_inf_entrada	ERROR: Falta informar uno de los 2 parámetros de entrada (p_id_grupo/p_participante_email)	
	exc_existe_grupo	ERROR: No existe el grupo de compra	

	informado:p_id_grupo
exc_no_existe_participante	ERROR: No existen ningún participante con el email:p_participante_email
exc_no_datos	ERROR: No existen datos a partir de la información proporcionada.
Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_ID_GRUPO NUMBER; P_PARTICIPANTE_EMAIL VARCHAR2(100); P_RSP VARCHAR2(1000); BEGIN P_ID_GRUPO := 1; PKG_AMIGOS.LISTADOPARTICIPANTESGRUPO( P_ID_GRUPO , P_PARTICIPANTE_EMAIL , P_RSP); END;</pre>

Tabla 75, Procedimiento: pkg\_amigos.ListadoParticipantesGrupo.

### 12.3.- Procedimientos de pruebas del sistema.

<b>Procedimiento</b>	<b>pkg_test.Elimina_Datos.</b>		
<b>Funcionalidad</b>	Procedimiento que elimina todos los datos de pruebas e inicializa las secuencias de las tablas para dejar el sistema listo para su uso.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_RSP VARCHAR2(1000); BEGIN PKG_TEST.Elimina_Datos(P_RSP); END;</pre>		

Tabla 76, Procedimiento: pkg\_test.Elimina\_Datos.

<b>Procedimiento</b>	<b>pkg_test.Pruebas_Unitarias.</b>		
<b>Funcionalidad</b>	<p>Testea todos los procedimientos para comprobar su correcto funcionamiento.</p> <p>Todas las ejecuciones dejarán el resultado por pantalla pudiendo contrastar todas las acciones con sus resultados.</p>		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE P_RSP VARCHAR2(1000); BEGIN PKG_TEST.Pruebas_Unitarias(P_RSP); END;</pre>		

Tabla 77, Procedimiento: pkg\_test.Pruebas\_Unitarias.

<b>Procedimiento</b> pkg_test.Pruebas_Funcionales.			
<b>Funcionalidad</b>	Realiza un conjunto de pruebas para demostrar el funcionamiento del regalo único con todas las posibilidades. También se realizan todas las devoluciones posible (baja producto, baja producto_proveedor, baja deseo, baja regalo, baja participante, baja pago, baja grupo de compra y baja por superar fecha recaudación).		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_RSP VARCHAR2(1000); BEGIN   PKG_TEST.Pruebas_Funcionales(P_RSP); END;</pre>		

Tabla 78, Procedimiento: pkg\_test.Pruebas\_Funcionales.

### 12.3.1.- Procedimientos de pruebas del módulo estadístico.

<b>Procedimiento</b> pkg_test.Pruebas_Estadisticas.			
<b>Funcionalidad</b>	Conjunto de situaciones predefinidas que justifican el correcto funcionamiento de los módulos estadísticos. Simplemente llama con uno valores concretos a los procedimientos que utiliza el generador_datos_estadisticos.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	Others	ERROR: +sqlcode:+sqlerrm	
<b>Ejemplo de invocación</b>	<pre> DECLARE   P_RSP VARCHAR2(1000); BEGIN   PKG_TEST.Pruebas_Estadisticas (P_RSP); END;</pre>		

Tabla 79, Procedimineto: pkg\_test.Pruebas\_Estadisticas.

<b>Procedimiento</b>	<b>pkg_test.Generador_datos_estadisticos.</b>		
<b>Funcionalidad</b>	Procedimiento que genera un conjunto de pruebas a partir de los parámetros de entrada y justifica el correcto funcionamiento de las estadísticas.		
<b>Parámetros de entrada</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_num_personas	Number	Número de personas homenajeadas. Obligatorio.
	p_nim_deseos	Number	Define el intervalo mínimo de regalos que puede tener la persona. Obligatorio.
	p_max_deseos	Number	Define el intervalo máximo de regalos que puede tener la persona. Obligatorio.
	p_num_grupos	Number	Define el número de grupos a crear. El usuario homenajeadado de cada grupo será aleatorio y el municipio de entrega también será aleatorio. Obligatorio.
	p_num_participantes	Number	Define el número de participantes que crea en cada uno de los grupos. Obligatorio.
	p_importe_maximo	Number	Define el intervalo máximo que puede alcanzar cada pago. El participante y regalo que realiza cada pago es aleatorio. Obligatorio.
	p_num_reg_pag	Number	Define el número de regalos que deben quedar pagados en cada grupo. Admite el valor 0.
	p_num_devoluciones	Number	Define el número de devoluciones que se deben realizar de manera aleatoria por cada grupo de compra. Admite el valor 0.
	p_num_meses_cobros	Number	Define el número de meses máximo que debe atrasar los cobros para comprobar que las estadísticas reflejan la fecha de compra. Obligatorio.
p_num_meses_envio	Number	Define el intervalo máximo que debe atrasar la fecha de recepción del envío para jugar con las estadísticas de las entregas. Obligatorio.	
<b>Parámetros de salida</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Descripción</b>
	p_rsp	Varchar2	Indica si el procedimiento se ha ejecutado correctamente con el valor 'OK' o en caso contrario 'ERROR: + tipo de error'.
<b>Errores controlados</b>	<b>Nombre de la excepción</b>	<b>Valor devuelto</b>	
	exc_no_num_personas	ERROR: Falta informar el número de personas a dar de alta.	
	exc_no_min_deseos	ERROR: Falta informar el número mínimo de deseos a dar de alta	
	exc_no_max_deseos	ERROR: Falta informar el número máximo de deseos a dar de alta	
	exc_min_mayor_max	ERROR: el número mínimo de deseos es mayor que el máximo	
	exc_no_num_grupos	ERROR: Falta informar el número de grupos de	

	compra a dar de alta
exc_no_num_participantes	ERROR: Falta informar el número de participantes a dar de alta en cada grupo
exc_no_importe_maximo	ERROR: Falta informar el importe máximo que puede alcanzar los pagos
exc_no_num_reg_pag	ERROR: Falta informar el número de regalos pagados que tiene que quedar en cada grupo
exc_neg_num_devoluciones	ERROR: Falta informar el número de devoluciones que se realizarán en cada grupo de compra
exc_neg_num_meses_cobros	ERROR: Falta informar el número de meses que retrocederá la fecha de los cobros.
exc_no_num_meses_envio	ERROR: Falta informar el número de meses de meses que se aplaza el envío de los pedidos. Valor >=0.
Others	ERROR: +sqlcode:+sqlerrm
<b>Ejemplo de invocación</b>	<pre> DECLARE P_NUM_PERSONAS NUMBER; P_MIN_DESEOS NUMBER; P_MAX_DESEOS NUMBER; P_NUM_GRUPOS NUMBER; P_NUM_PARTICIPANTES NUMBER; P_IMPORTE_MAXIMO NUMBER; P_NUM_REG_PAG NUMBER; P_NUM_DEVOLUCIONES NUMBER; P_NUM_MESES_COBROS NUMBER; P_NUM_MESES_ENVIO NUMBER; P_RSP VARCHAR2(1000); BEGIN P_NUM_PERSONAS := 4; P_MIN_DESEOS := 1; P_MAX_DESEOS := 10; P_NUM_GRUPOS := 10; P_NUM_PARTICIPANTES := 5; P_IMPORTE_MAXIMO := 300; P_NUM_REG_PAG := 5; P_NUM_DEVOLUCIONES := 0; P_NUM_MESES_COBROS := 5; P_NUM_MESES_ENVIO := 3;  PKG_TEST.GENERADOR_DATOS_ESTADISTICOS( P_NUM_PERSONAS, P_MIN_DESEOS, P_MAX_DESEOS, P_NUM_GRUPOS, P_NUM_PARTICIPANTES, P_IMPORTE_MAXIMO, P_NUM_REG_PAG, P_NUM_DEVOLUCIONES, P_NUM_MESES_COBROS, P_NUM_MESES_ENVIO, P_RSP ); END; </pre>

Tabla 80, Procedimiento: pkg\_test.Generador\_datos\_estadisticos.