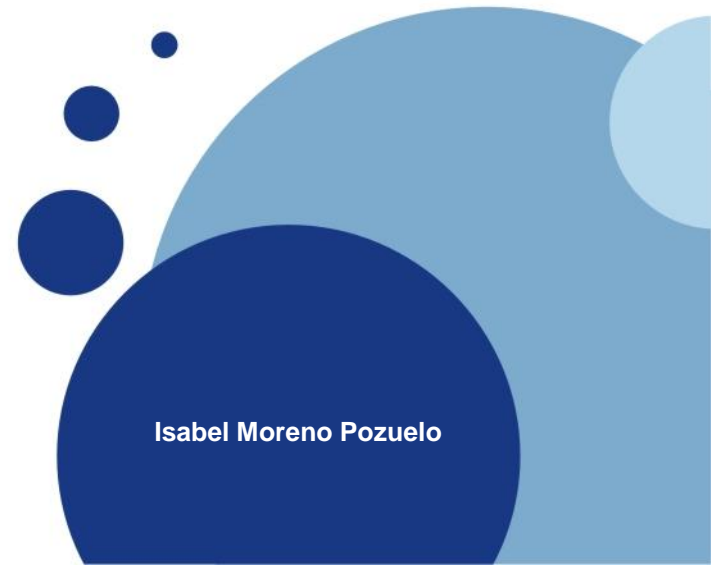


TFC

Base de Datos Relacionales





Objetivos del Proyecto

Página

1

- Desarrollo de una Base de Datos para la Gestión de Carteras de Inversión
- Permitirá almacenar:
 - Las órdenes de compra/venta de acciones, así como su estado de ejecución
 - Los datos asociados a los clientes junto con la cartera de inversión actual e histórica
 - Información estadística que es consultada por los clientes y analistas del *broker*
- La metodología a utilizar es el ciclo de vida clásico que comprende las siguientes etapas:
 - Análisis, Diseño, Codificación e implementación, pruebas y mantenimiento

- **R1:** Almacenar los datos de los clientes y la composición de la cartera de acciones
- **R2:** Las órdenes de compra/venta podrán ser dadas de alta, de baja o modificadas por parte de los clientes o por sistemas automáticos a través de los procedimientos almacenados que hay que implementar en este proyecto.
- **R3:** Se ha de almacenar los datos de las acciones.



Requisitos

- **R4:** Las funcionalidades que tiene que ejecutar este sistema son las siguientes:
 - Procedimientos de Alta, Baja y Modificación (ABM) de las órdenes de compra/venta, de los clientes y de las carteras de inversión
 - Procedimiento actualización del valor de cierre diario de las acciones
 - Procedimientos de consulta que permiten obtener la información bajo diversos criterios
 - Fecha concreta, Identificador de cliente, Operaciones, etc..

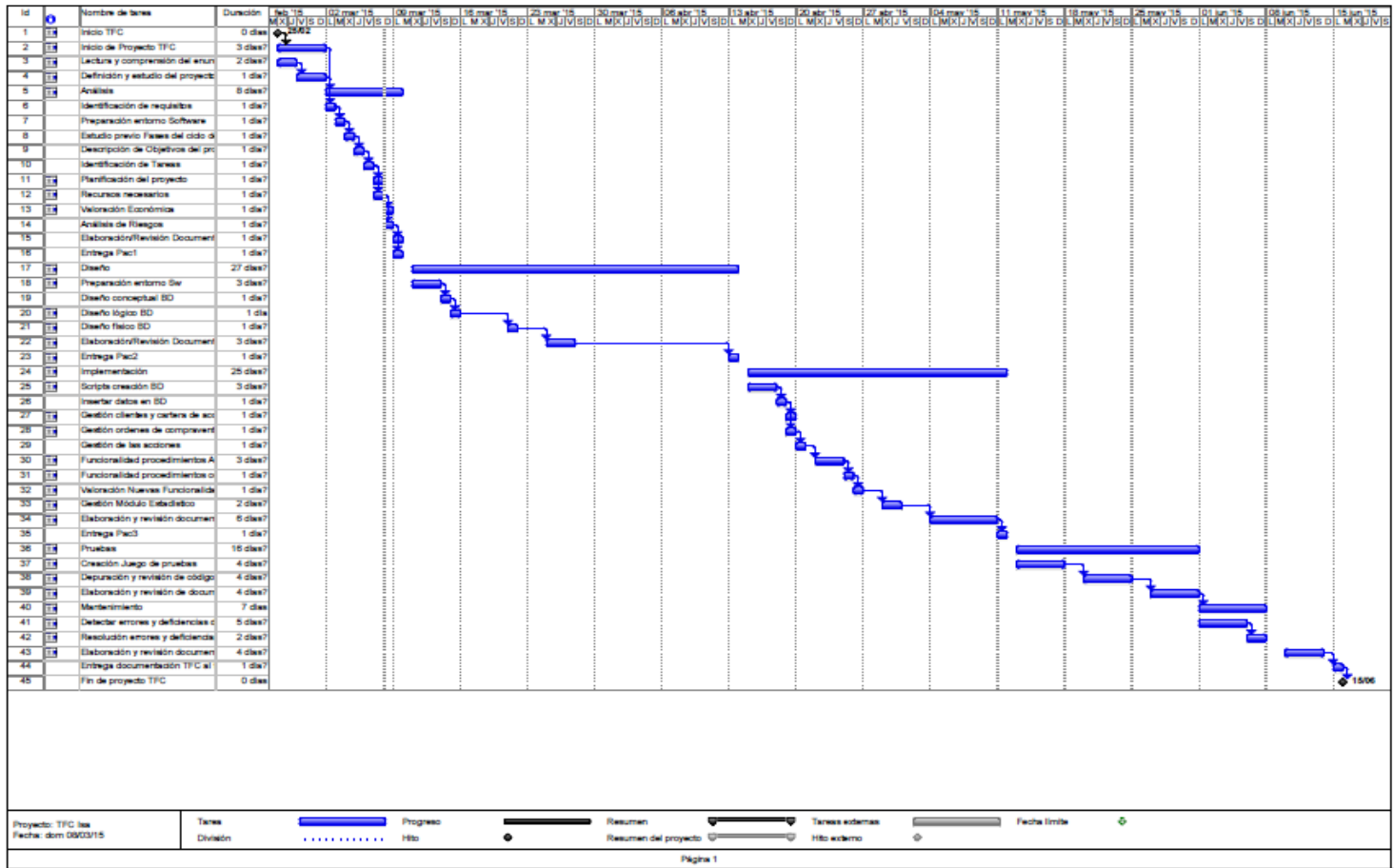


Requisitos

- **R5:** La implementación del módulo estadístico se debe realizar a través de los procedimientos que implementan las funcionalidades de este proyecto y a través de una *select* sobre un registro de una tabla
- Una manera de realizar estos cálculos es a través de *triggers* que actualicen las estadísticas:
 - Beneficio neto en euros y el porcentaje de su cartera.
 - Número de acciones disponibles dentro del *broker*.
 - El cliente con la cartera de acciones con un valor más grande junto con el valor en euros de su cartera.
 - La acción más negociada, comprada y vendida, en el *broker* de forma histórica junto con su volumen



Diagrama de Gantt





Planificación

FECHA	ENTREGA	DESCRIPCIÓN
25/02/2015	Inicio TFC	Inicio curso UOC 2014/2015 segundo cuatrimestre
09/03/2015	PAC1	Descripción del sistema, alcance y objetivos. Planificación detallada. Explicación PAC y PAC3. Recursos humanos, técnicos y materiales. Análisis de riesgos. Valoración inicial de las horas y coste del proyecto.
13/04/2015	PAC2	Fase de Diseño del proyecto
11/05/2015	PAC3	Fase de Implementación, Pruebas y Mantenimiento
15/06/2015	Proyecto final	Memoria
22/06/2015	Inicio Tribunal virtual	
26/06/2015	Final Tribunal virtual	

Riesgos

Tablas resumen de los riesgos del proyecto

PRINCIPALES RIESGOS			
CAUSA1	Requisitos no estén bien definidos	ACCIÓN1	Mayor esfuerzo en esta fase. Realizar preguntas al cliente/profesor
CAUSA2	Jefe de proyecto no sea muy experto	ACCIÓN2	Formación adecuada
CAUSA3	El equipo de proyecto no esté bien formado en el entorno de desarrollo.	ACCIÓN3	Buscar cursos de formación

RIESGOS TÉCNICOS			
CAUSA1	Pérdida de datos	ACCIÓN1	Copias de seguridad diarias en otro dispositivo
CAUSA2	Ordenador de desarrollo defectuoso o se produzca una avería	ACCIÓN2	Ordenador de sustitución actualizado
CAUSA3	Caducidad en las licencias de software	ACCIÓN3	Actualización de versiones y licencias. Uso software freeware

RIESGOS POR EL DESARROLLO DE OTRAS ACTIVIDADES			
CAUSA1	Obligatoriedad de cumplir la jornada laboral	ACCIÓN1	Aprovechar las horas libres
CAUSA2	Realizar prácticas de otra asignatura	ACCIÓN2	Disponibilidad de días de vacaciones
CAUSA3	Enfermedad o accidentes	ACCIÓN3	Modificar a posteriori las fechas de la planificación



Costes del Proyecto

	IMPORTE		
	HORAS	€	TOTAL
Jefe de Proyecto	40	109,73	4.389,20
Analista	67	102,72	6.882,24
Administrador BD	5	80,02	400,10
Programador	75	77,80	5.835,00
<u>TOTALES</u>	187	370,27	17.506,54





Entorno Software Necesario

Oracle Express, Oracle Developer

ORACLE Oracle Database XE 11.2 Welcome: SYSTEM Logout

Home Storage Sessions Parameters **Application Express**

Home > Oracle Application Express

Create Application Express Workspace

Cancel **Create Workspace**

Database User Create New Use Existing

* Database Username

* Application Express Username

* Password

* Confirm Password

Getting Started

[Already have an account? Login Here](#)

To get started with Oracle Application Express, create a workspace. You will need to specify:

- Database Username - Name of the database user to be created
- Application Express Username - Your login name for the Application Express Workspace
- Password - Password of both your database user and Application Express user

Once created, you will be able to [login to your Application Express workspace](#) using these credentials

Oracle

Diagrama Entidad Relación

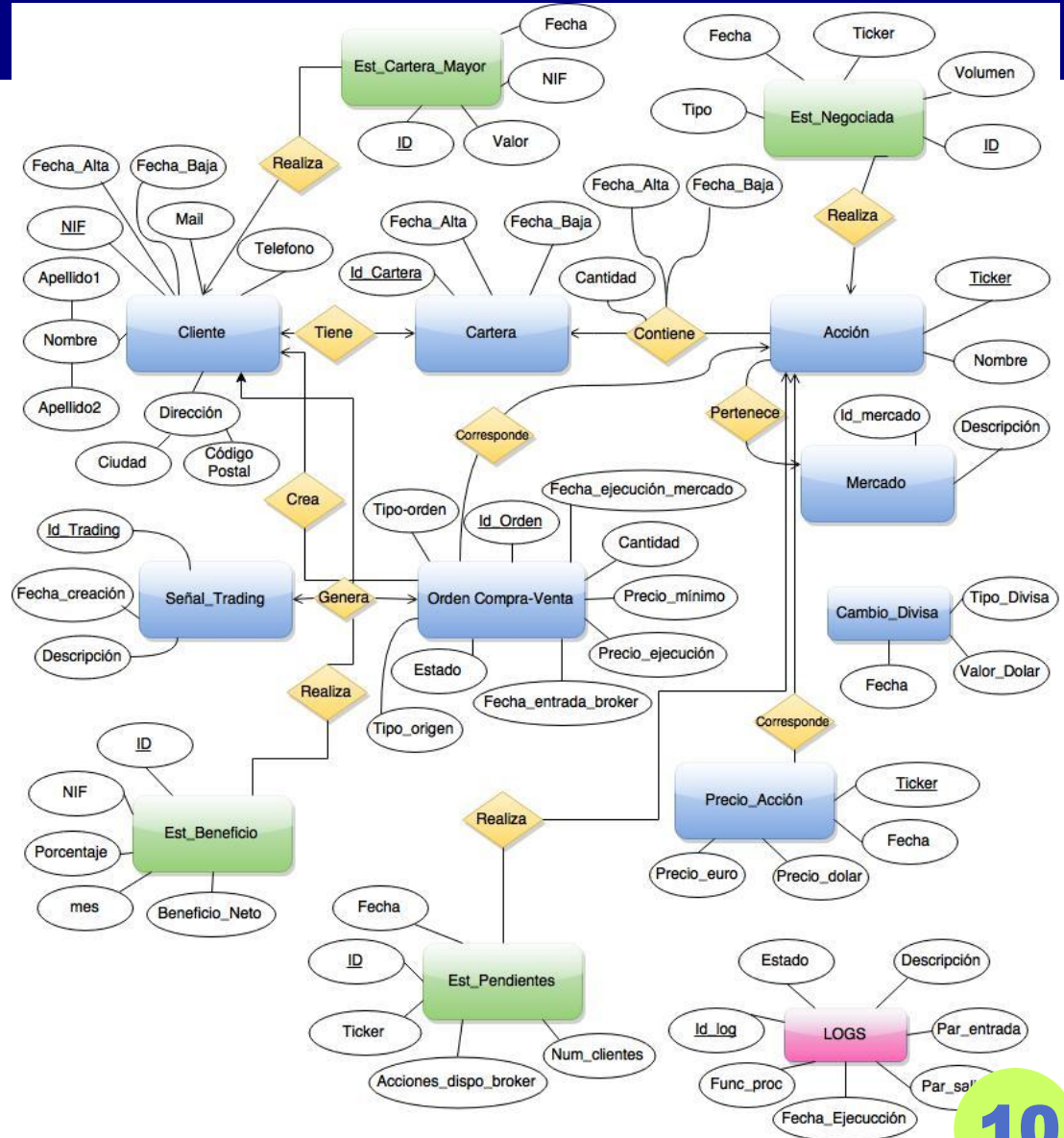
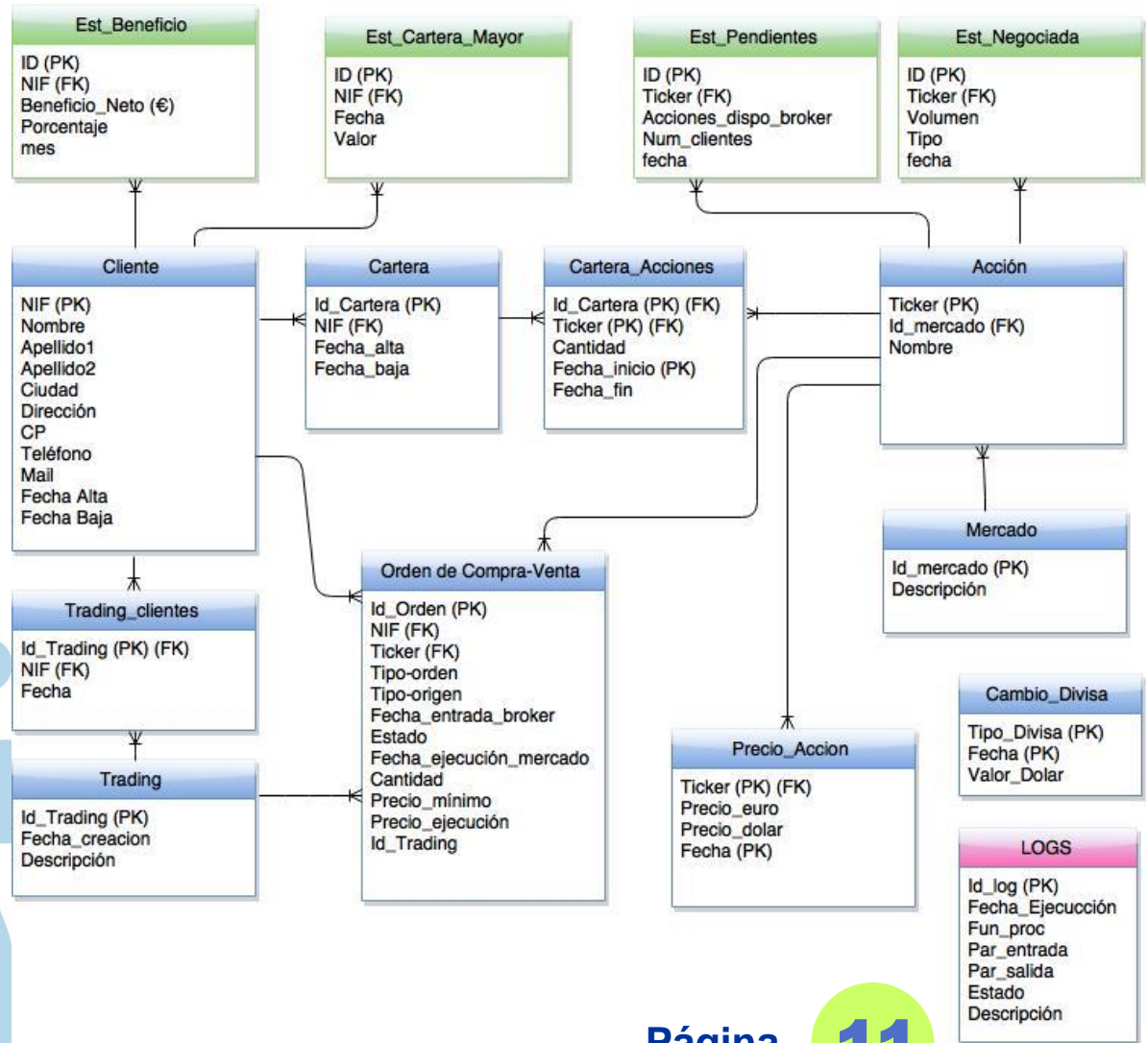
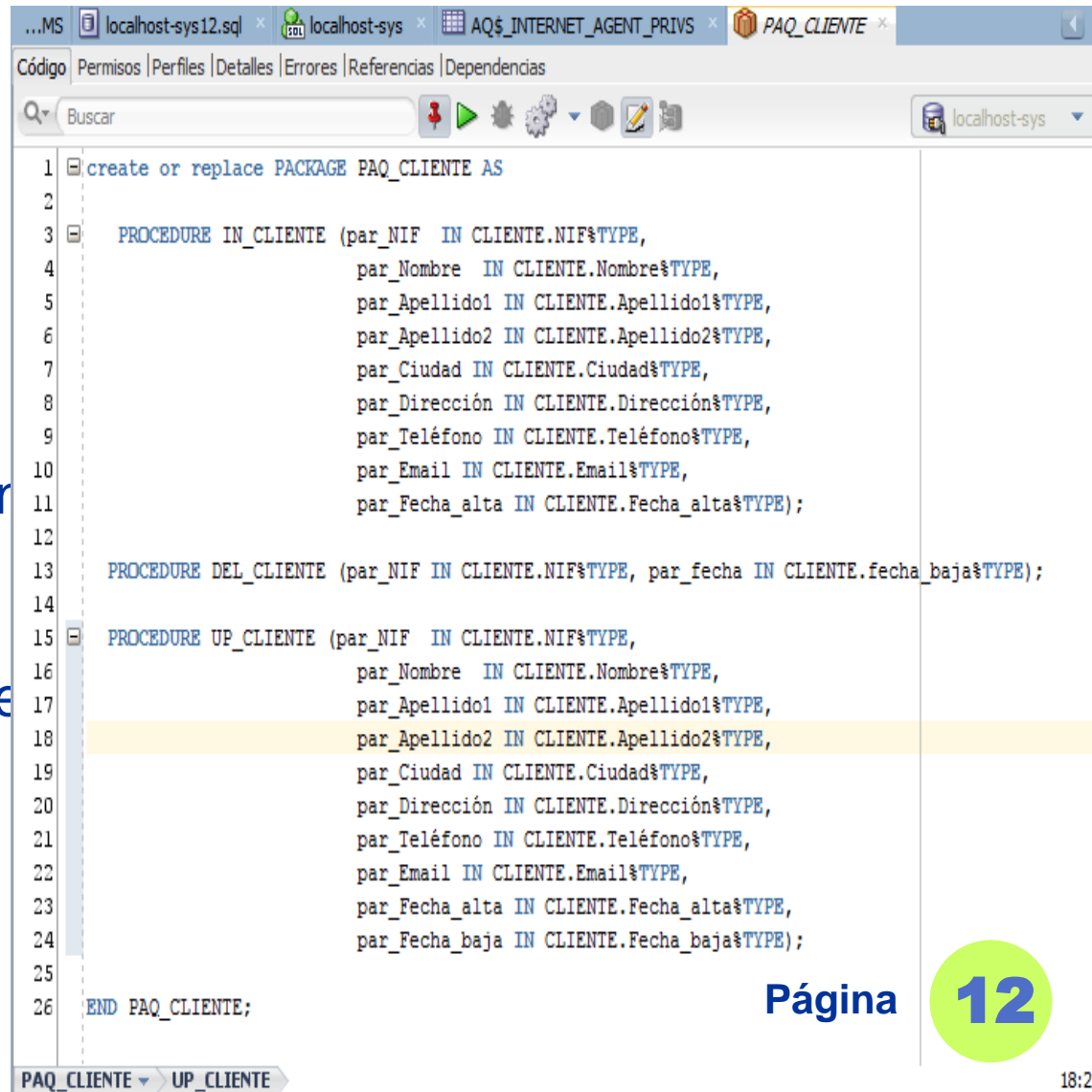




Diagrama Relacional



- Los procedimientos Alta, Baja, Modificaciones son de órdenes de compra/venta, de clientes, carteras de inversión y el valor de cierre diario
- Los paquetes proporcionar un mecanismo para extender el lenguaje. Es una estructura PL/SQL que permite almacenar juntos una serie de objetos relacionados



```
1 create or replace PACKAGE PAQ_CLIENTE AS
2
3     PROCEDURE IN_CLIENTE (par_NIF IN CLIENTE.NIF%TYPE,
4                           par_Nombre IN CLIENTE.Nombre%TYPE,
5                           par_Apellido1 IN CLIENTE.Apellido1%TYPE,
6                           par_Apellido2 IN CLIENTE.Apellido2%TYPE,
7                           par_Ciudad IN CLIENTE.Ciudad%TYPE,
8                           par_Dirección IN CLIENTE.Dirección%TYPE,
9                           par_Teléfono IN CLIENTE.Teléfono%TYPE,
10                          par_Email IN CLIENTE.Email%TYPE,
11                          par_Fecha_alta IN CLIENTE.Fecha_alta%TYPE);
12
13     PROCEDURE DEL_CLIENTE (par_NIF IN CLIENTE.NIF%TYPE, par_fecha IN CLIENTE.fecha_baja%TYPE);
14
15     PROCEDURE UP_CLIENTE (par_NIF IN CLIENTE.NIF%TYPE,
16                          par_Nombre IN CLIENTE.Nombre%TYPE,
17                          par_Apellido1 IN CLIENTE.Apellido1%TYPE,
18                          par_Apellido2 IN CLIENTE.Apellido2%TYPE,
19                          par_Ciudad IN CLIENTE.Ciudad%TYPE,
20                          par_Dirección IN CLIENTE.Dirección%TYPE,
21                          par_Teléfono IN CLIENTE.Teléfono%TYPE,
22                          par_Email IN CLIENTE.Email%TYPE,
23                          par_Fecha_alta IN CLIENTE.Fecha_alta%TYPE,
24                          par_Fecha_baja IN CLIENTE.Fecha_baja%TYPE);
25
26 END PAQ_CLIENTE;
```



Funciones y procedimientos para confeccionar listados:

Proc_Cartera.sql
Composición de la cartera de acciones vigente del cliente.

The screenshot displays three windows from SQL Developer:

- PROC_CARTERA - Editor:** Shows the PL/SQL code for the procedure. It creates a cursor to select actions from the 'cartera' table, grouped by NIF, id_cartera, and ticker. It also declares variables for currency prices and a counter.
- localhost-sys - Editor:** Shows the execution of the procedure with the following SQL statement:

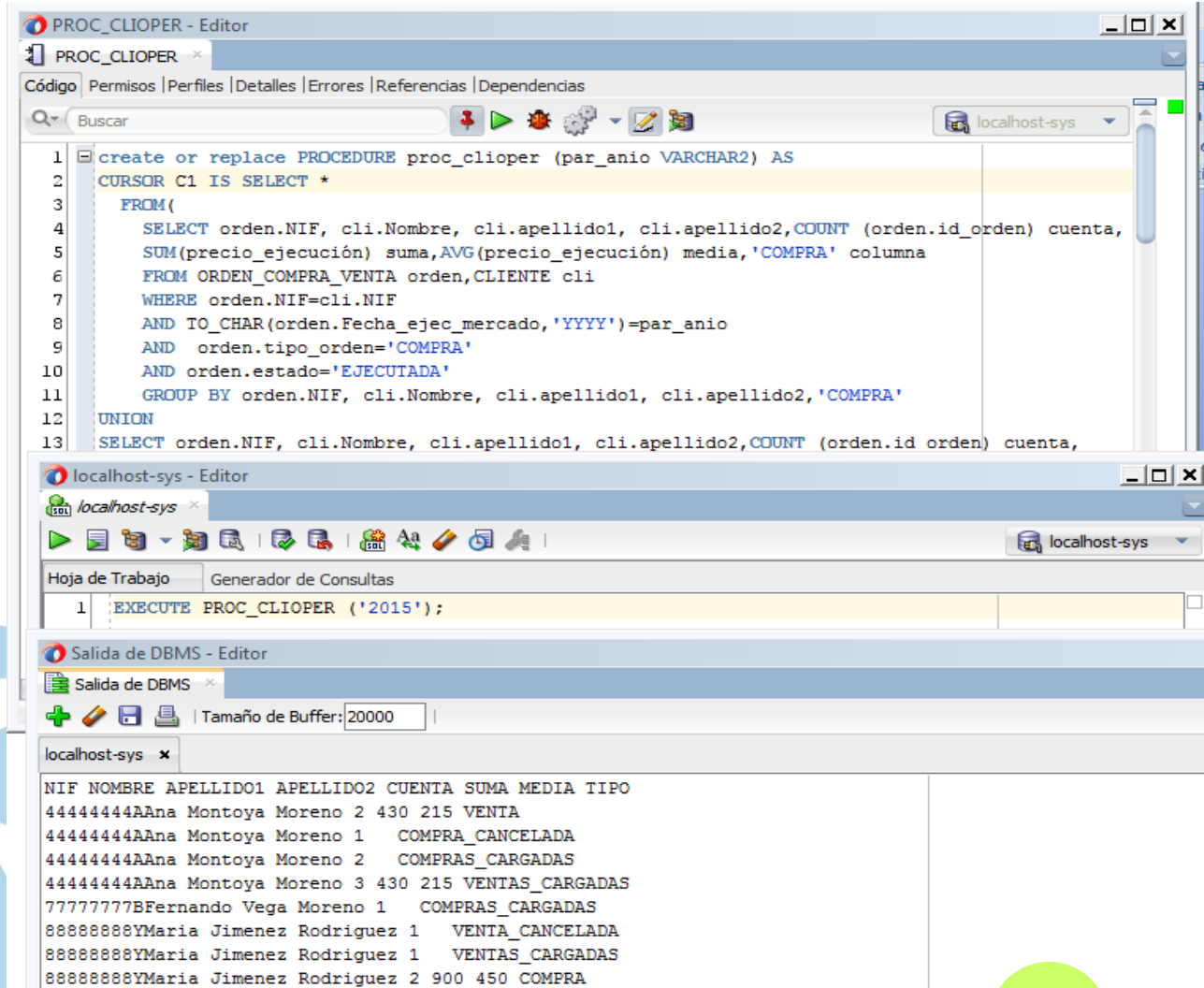
```
EXECUTE PROC_CARTERA ('01/04/2015', '99999999B');
```
- Salida de DBMS - Editor:** Shows the output of the procedure execution:

```
CARTERA, TICKER, PR.EURO, PR.DÓLAR  
CAR001 GRFS 8000 € 8604 $
```

Funciones y procedimientos para confeccionar listados:

Proc_Clioper.sql

Listado de todos los clientes que han realizado operaciones durante este año.



```

1 create or replace PROCEDURE proc_clioper (par_anio VARCHAR2) AS
2   CURSOR C1 IS SELECT *
3     FROM (
4       SELECT orden.NIF, cli.Nombre, cli.apellido1, cli.apellido2, COUNT (orden.id_orden) cuenta,
5       SUM(precio_ejecución) suma, AVG(precio_ejecución) media, 'COMPRA' columna
6       FROM ORDEN_COMPRA_VENTA orden, CLIENTE cli
7       WHERE orden.NIF=cli.NIF
8       AND TO_CHAR(orden.Fecha_ejec_mercado, 'YYYY')=par_anio
9       AND orden.tipo_orden='COMPRA'
10      AND orden.estado='EJECUTADA'
11      GROUP BY orden.NIF, cli.Nombre, cli.apellido1, cli.apellido2, 'COMPRA'
12 UNION
13 SELECT orden.NIF, cli.Nombre, cli.apellido1, cli.apellido2, COUNT (orden.id_orden) cuenta,

```

localhost-sys - Editor

Hoja de Trabajo: Generador de Consultas

```
1 EXECUTE PROC_CLIOPER ('2015');
```

Salida de DBMS - Editor

```

NIF NOMBRE APELLIDO1 APELLIDO2 CUENTA SUMA MEDIA TIPO
44444444AAna Montoya Moreno 2 430 215 VENTA
44444444AAna Montoya Moreno 1 COMPRA_CANCELADA
44444444AAna Montoya Moreno 2 COMPRAS_CARGADAS
44444444AAna Montoya Moreno 3 430 215 VENTAS_CARGADAS
77777777BFernando Vega Moreno 1 COMPRAS_CARGADAS
88888888YMaria Jimenez Rodriguez 1 VENTA_CANCELADA
88888888YMaria Jimenez Rodriguez 1 VENTAS_CARGADAS
88888888YMaria Jimenez Rodriguez 2 900 450 COMPRA

```



Funciones y procedimientos para confeccionar listados:

Proc_Acciones_Oper.sql

Listado de todas las acciones sobre las cuales se han realizado alguna operación.

The screenshot shows three windows in SQL Developer:

- PROC_ACCIONES_OPER - Editor:** Contains the SQL code for creating a procedure.


```

      create or replace PROCEDURE proc_acciones_oper (par_anio VARCHAR2) AS
      -- Se realiza el cursor con los datos de las acciones de compra y de venta.
      CURSOR C1 IS SELECT * FROM (select cv.ticker TICKER, acc.nombre NOMBRE,
      count(cv.id_orden) CONTAR, sum(cv.precio_ejecución) SUMA,
      avg(cv.precio_ejecución) MEDIA, cv.tipo_orden ORDEN
      from ACCION acc, ORDEN_COMPRA_VENTA cv
      where acc.ticker=cv.ticker
      and TO CHAR(cv.Fecha ejec mercado, 'YYYY')=par_anio
      
```
- localhost-sys - Editor:** Shows the execution of the procedure.


```

      EXECUTE PROC_ACCIONES_OPER ('2015');
      
```
- Salida de DBMS - Editor:** Shows the output of the procedure.


```

      TICKER, NOMBRE,Nº,SUMA, MEDIA, $REVALOR
      GRFS Grifols 2 320 320 COMPRA -92,347215 $
      SAB Banc Sabadell 3 1080 360 COMPRA 25,777778 $
      SAB Banc Sabadell 1 230 0 VENTA 25,777778 $
      GRFS Grifols 3 430 0 VENTA -92,347215 $
      
```




Funciones y procedimientos para confeccionar listados:

Proc_Acciones_Aut.sql

Listado de las diez acciones más adquiridas por los sistemas automáticas.

The screenshot displays three windows from SQL Developer:

- PROC_ACCIONES_AUT - Editor:** Shows the SQL code for creating a procedure. The code defines a cursor to select the top 10 actions by quantity from the 'ORDEN_COMPRA_VENTA' table, filtered by 'AUTOMATICA' type and 'EJECUTADA' status. It also declares output parameters: err_num (NUMBER(6,2)), err_msg (VARCHAR2(200)), res (VARCHAR2(200)), RSP (VARCHAR2(500)), and PAR_IN (VARCHAR2(30)).
- localhost-sys - Editor:** Shows the execution of the procedure: `EXECUTE PROC_ACCIONES_AUT ();`
- Salida de DBMS - Editor:** Shows the output of the procedure execution, listing the top 10 actions: TICKER, NOMBRE COMPLETO, Banc Sabadell 400, and Grifols 80.

Módulo Estadístico

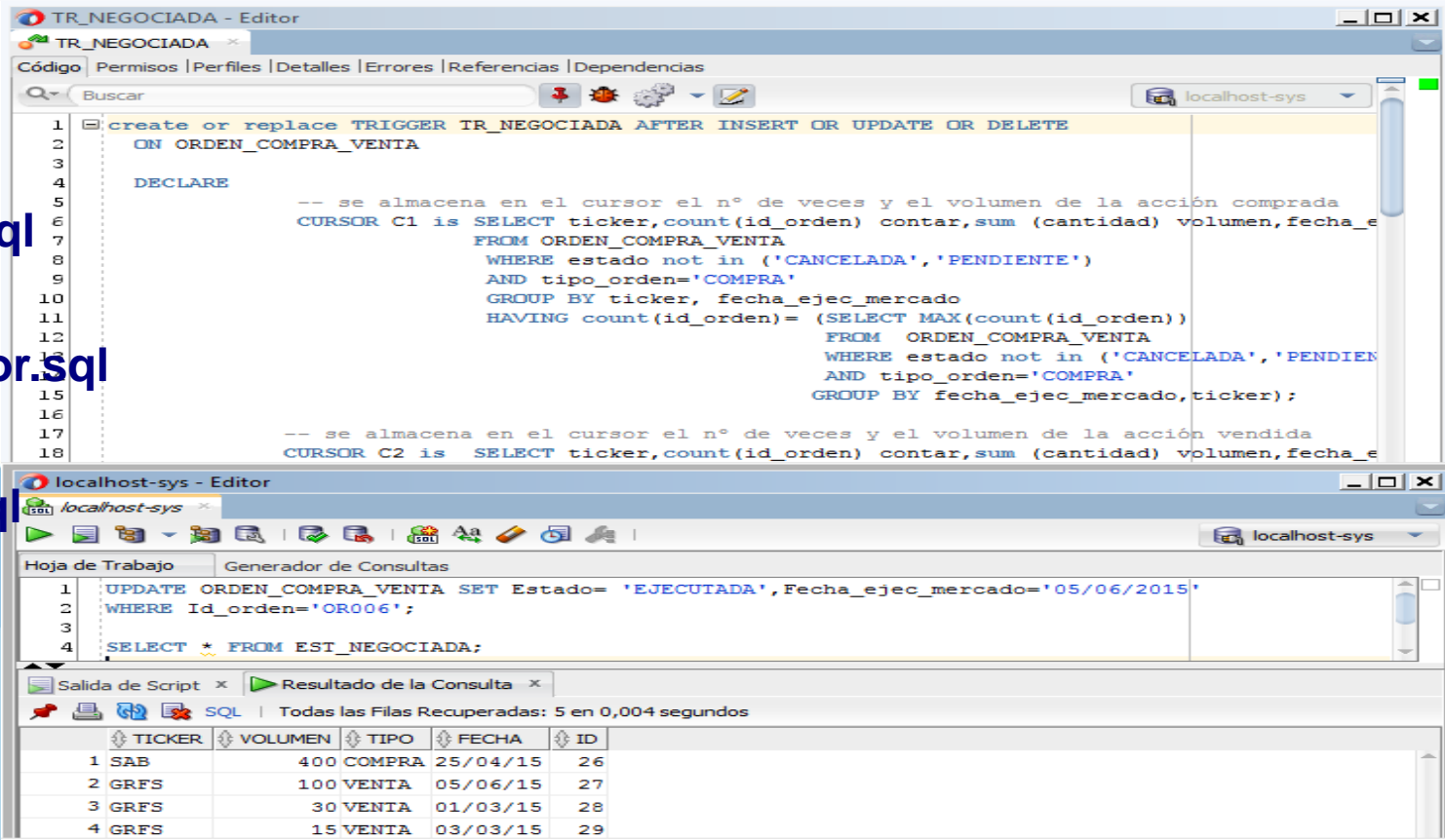
Permite dar respuesta inmediata a las consultas definidas en los requisitos. Hay que actualizar las tablas de estadísticas a través de disparadores o *triggers*.

Est_Beneficio.sql

Est_Pendientes.sql

Est_Cartera_Mayor.sql

Est_Negociada.sql



The screenshot shows two windows from a database editor. The top window, titled 'TR_NEGOCIADA - Editor', displays the following SQL code:

```

1 create or replace TRIGGER TR_NEGOCIADA AFTER INSERT OR UPDATE OR DELETE
2 ON ORDEN_COMPRA_VENTA
3
4 DECLARE
5
6     -- se almacena en el cursor el n° de veces y el volumen de la acción comprada
7     CURSOR C1 is SELECT ticker,count(id_orden) contar,sum (cantidad) volumen,fecha_e
8 FROM ORDEN_COMPRA_VENTA
9 WHERE estado not in ('CANCELADA','PENDIENTE')
10 AND tipo_orden='COMPRA'
11 GROUP BY ticker, fecha_ejec_mercado
12 HAVING count(id_orden)=(SELECT MAX(count(id_orden))
13 FROM ORDEN_COMPRA_VENTA
14 WHERE estado not in ('CANCELADA','PENDIEN
15 AND tipo_orden='COMPRA'
16 GROUP BY fecha_ejec_mercado,ticker);
17
18     -- se almacena en el cursor el n° de veces y el volumen de la acción vendida
19     CURSOR C2 is SELECT ticker,count(id_orden) contar,sum (cantidad) volumen,fecha_e

```

The bottom window, titled 'localhost-sys - Editor', shows a query being executed. The SQL code is:

```

1 UPDATE ORDEN_COMPRA_VENTA SET Estado= 'EJECUTADA',Fecha_ejec_mercado='05/06/2015'
2 WHERE Id_orden='OR006';
3
4 SELECT * FROM EST_NEGOCIADA;

```

The result of the query is displayed in a table with the following data:

TICKER	VOLUMEN	TIPO	FECHA	ID
1 SAB	400	COMPRA	25/04/15	26
2 GRFS	100	VENTA	05/06/15	27
3 GRFS	30	VENTA	01/03/15	28
4 GRFS	15	VENTA	03/03/15	29



Logs

Para comprobar las ejecuciones del producto y los posibles errores.

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
1 ID_LOG	VARCHAR2 (10 BYTE)	No	(null)	1	(null)
2 FECHA_EJECUCIÓN	DATE	Yes	(null)	2	(null)
3 FUNC_PROC	VARCHAR2 (50 BYTE)	Yes	(null)	3	(null)
4 PAR_ENTRADA	VARCHAR2 (200 BYTE)	Yes	(null)	4	(null)
5 PAR_SALIDA	VARCHAR2 (200 BYTE)	Yes	(null)	5	(null)
6 ESTADO	VARCHAR2 (20 BYTE)	Yes	(null)	6	(null)
7 DESCRIPCION	VARCHAR2 (500 BYTE)	Yes	(null)	7	(null)

Hoja de Trabajo | Generador de Consultas

```
SELECT * FROM LOGS;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 11 en 0,003 segundos

ID_LOG	FECHA_EJECUCIÓN	FUNC_PROC	PAR_ENTRADA	PAR_SALIDA	ESTADO	DESCRIPCION
1 301	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
2 303	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
3 302	11/06/15	PROC_CARTERA	(null)	RSP: ERROR 100: ORA-01403: no data found	OK	SYSTEM
4 304	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
5 305	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
6 306	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
7 307	11/06/15	PROC_CARTERA	(null)	(null)	OK	SYSTEM
8 321	12/06/15	PR_CIERRE_DIARIO	SAB 2,83 12/06/15	ERROR -1: ORA-00001: unique constraint...	ERROR	SYSTEM
9 322	12/06/15	PR_CIERRE_DIARIO	SAB 2,83 12/06/15	ERROR -1: ORA-00001: unique constraint...	ERROR	SYSTEM
10 323	12/06/15	PR_CIERRE_DIARIO	SAB 2,83 12/06/15	ERROR -1: ORA-00001: unique constraint...	ERROR	SYSTEM
11 324	12/06/15	PR_CIERRE_DIARIO	SAB 2,83 12/06/15	OK	OK	SYSTEM



Muchas gracias

