



TFC BD RELACIONALES

Isabel Moreno Pozuelo

15/06/2015

“Como ya no estaba matriculado en la universidad y no tenía clases obligatorias, decidí atender al curso de caligrafía para aprender cómo se hacía. Aprendí cosas sobre el Serif y tipografías Sans Serif, sobre los espacios variables entre letras, sobre qué hace realmente grande a una gran tipografía. Era sutilmente bello, histórica y artísticamente, de una forma que la ciencia no puede capturar, y lo encontré fascinante. Nada de esto tenía ni la más mínima esperanza de aplicación práctica en mi vida. Pero diez años más tarde, cuando estábamos diseñando Macintosh, todo eso volvió a mí.

Y diseñamos el Mac con eso en su esencia. Fue el primer ordenador con tipografías bellas. Si nunca me hubiera dejado caer por aquél curso concreto en la universidad, el Mac jamás habría tenido múltiples tipografías, ni caracteres con espaciado proporcional. Y como Windows no hizo más que copiar el Mac, es probable que ningún ordenador personal los tuviera ahora.

Por supuesto, era imposible conectar los puntos mirando hacia el futuro cuando estaba en clase, pero fue muy, muy claro al mirar atrás diez años más tarde. Lo diré otra vez: no puedes conectar los puntos hacía adelante, sólo puedes hacerlo hacía atrás. Así que tenéis que confiar en que los puntos se conectarán alguna vez en el futuro. Tienes que confiar en algo, tu instinto, el destino, la vida, el karma, lo que sea.

Esta forma de actuar nunca me ha dejado tirado, y ha marcado la diferencia en mi vida.”

Steve Jobs, Discurso en la Universidad de Stanford (2005).

Tabla de Contenido:

Tabla de Contenido:	3
1. Alcance del proyecto, objetivos, descripción	4
2. Planificación, Diagrama de Gantt.....	9
3. Explicación PAC2 y PAC3	12
4. Recursos Humanos, Técnicos y Materiales	12
4.1. Recursos Humanos	12
4.2. Recursos Técnicos y Materiales	15
5. Análisis de riesgos y controles	15
6. Valoración Inicial de las horas y coste total del proyecto	18
7. Preparación Entorno Software necesario: Oracle Express, SGBD.....	21
8. Diseño Conceptual:	22
8.1. Diagrama Entidad Relación:	22
9. Diseño Lógico:	26
9.1. Diagrama Lógico Relacional:	26
10. Diseño Físico:	31
10.1. Tablas:	31
10.2. Paquetes, procedimientos y funciones:	31
10.3. Funciones y procedimientos para confeccionar listados:	37
10.4. Disparadores:.....	38
11. IMPLEMENTACIÓN:	39
11.1. Creación tablas de la base de datos:	39
11.2. Inserción de Datos:.....	41
11.3. Procedimientos de Alta, Baja y Modificación:	41
11.4. Procedimientos de Consulta:	47
11.5. Módulo Estadístico:.....	49
12. PRUEBAS:	52
12.1. Preparación de Datos:.....	52
12.2. Pruebas en el compilador de Oracle Developer:	53
12.3. Pruebas a través de consultas	53
12.4. Análisis de Resultados:	53
13. ANEXO	55
14. BIBLIOGRAFÍA	56

1. Alcance del proyecto, objetivos, descripción

ALCANCE DEL PROYECTO:

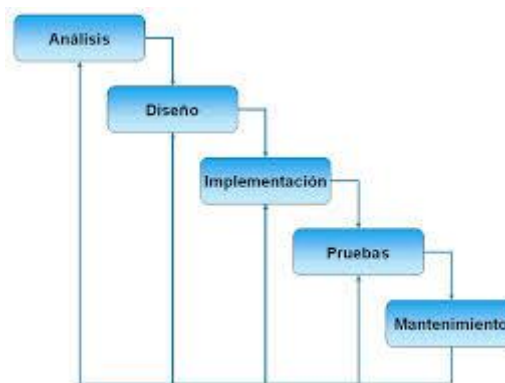
El Trabajo fin de carrera (TFC) es el resultado final del aprendizaje progresivo de las asignaturas que engloban la carrera de Ingeniería informática de Gestión. Los conocimientos adquiridos a lo largo de todos estos años y la experiencia profesional hacen que la alumna tenga la capacidad de desarrollar este proyecto con eficacia e ilusión.

Dentro del conjunto de las áreas propuestas para realizar el TFC, la alumna elige este proyecto por la gran importancia que existe hoy en día en el diseño de una base de datos (BD), que es fundamental para cualquier empresa existente.

Las tareas imprescindibles son el análisis, diseño e implementación de una BD relacional que almacene datos referentes a la compra y venta de acciones.

La metodología a utilizar es el ciclo de vida clásico que comprende las siguientes etapas:

Análisis, Diseño, Codificación e implementación, pruebas y mantenimiento.



El análisis de requerimientos consiste en reunir las necesidades del producto mediante conversaciones y reuniones con el cliente.

El diseño describe la estructura interna del producto y suele representarse con diagramas y texto explicativos con el objetivo.

La implementación equivale a la programación, la generación del código que se necesita para cumplir con los requisitos.

Las pruebas se confeccionan para detectar y controlar posibles errores antes de la entrega al cliente del producto.

El mantenimiento es una fase que puede ser constante y que permite conseguir que el producto permanezca actualizado.

OBJETIVOS:

El objetivo de este proyecto es desarrollar una base de datos relacional que contenga los datos necesarios para realizar la gestión de carteras de inversión. Esta BD se

La BD 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*.

DESCRIPCIÓN:

Se ha de realizar el diseño e implementación de la base de datos para dar respuesta a la necesidad de la gestión de un proyecto de carteras de inversión.

Un importante *broker* dedicado a la gestión de activos financieros ha decidido abrir un concurso público con el objetivo de recibir propuestas para el diseño de la BD.

La BD diseñada tiene que almacenar principalmente: órdenes de compra/venta de acciones y su estado de ejecución, datos asociados a los clientes y sus carteras de inversiones actuales e históricas e información estadística que será consultada por los clientes y por los analistas del *broker*.

Hay que destacar que los requisitos funcionales del sistema son una pieza clave en el proyecto debido a que a través de éstos el cliente expresa sus necesidades.

La base de datos tiene que:

R1: Almacenar los datos de los clientes y la composición de la cartera de acciones.

De los clientes es necesario reflejar los datos personales como:

- Nombre
- apellidos
- NIF, etc.

También se debe poder acceder a los datos de la cartera de acciones actual e histórica.

En principio un cliente tendrá una única cartera de acciones que se modificará en función de las órdenes de compra/venta que se ejecuten.

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.

Por cada orden de compra/venta de acciones se deberá almacenar los siguientes datos:

- Identificador único de la orden.

- Identificador del cliente al que pertenece la orden.
- Identificador de la acción sobre la cual se está ejecutando la orden.
- Tipo de orden: órdenes de *venta* y órdenes de *compra*.
- Tipos de origen: *manual*, en el caso en el que el cliente haya ejecutado el alta de manera manual, y *automática*, en el caso en el que el sistema haya ejecutado el alta. En este último caso, habrá que identificar el sistema de señales que ha realizado la orden.
- Fecha en la que la orden ha entrado en el sistema el *broker*.
- Estado en el que se encuentra la orden: *pendiente*, *ejecutada* y *cancelada*.
- Fecha en la que se ha ejecutado realmente la orden en el mercado.
- Cantidad de acciones a adquirir o a vender.
- Precio de la acción mínima en la que se quiere realizar la operación de compra y venta (opcional)
- Precio en el que se ha realizado a operación una vez que se ha ejecutado la orden en el mercado.

Una vez que la orden se selecciona como ejecutada no se podrá modificar ni borrar en ningún caso.

R3: Se ha de almacenar los datos de las acciones.

Para cada acción es necesario reflejar como mínimo lo siguiente:

- Identificador o *ticker* de la acción: GRFS, SAB, etc.
- Identificador del mercado financiero en el cual se negocia: IBEX35, SP500, NASDAQDAX, FTSE 100, etc.
- Nombre completo de la acción.
- Histórico de precios de la acción y el precio de cierre diario de la acción.

Se supondrá que las carteras de inversión estarán formadas solamente por acciones. Una posible mejora del proyecto es que el sistema gestione otros productos financieros como por ejemplo fondos de inversión, ETF, compra de divisas, etc.

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 siguiente:
 1. Dada una fecha concreta y el identificador de cliente como parámetros de entrada se genera un listado con la composición de la cartera de acciones

vigente, por la fecha concreta, del cliente y con el valor en euros de las acciones que tengan en ese momento.

2. Dado un año concreto como parámetro de entrada se genera un listado de todos los clientes que han realizado operaciones durante este año, y se tiene que incluir para cada cliente los datos siguientes:
 - Identificador del cliente
 - Nombre el cliente
 - Número de órdenes de compra cargadas durante el año.
 - Número de órdenes de venta cargadas durante el año.
 - Número de órdenes de compra ejecutadas durante el año.
 - Sumatorio en euros de todas las órdenes de compra ejecutadas durante el año.
 - Valor medio en euros de cada operación de compra ejecutada por el cliente durante el año.
 - Número de órdenes de venta ejecutadas durante el año.
 - Sumatorio en euros de todas las órdenes de venta ejecutadas durante el año.
 - Valor medio en euros de cada operación de venta ejecutada por el cliente durante el año.
 - Número de órdenes de compra canceladas durante el año.
 - Número de órdenes de venta canceladas durante el año.

3. Dado un año concreto como parámetro de entrada se genera un listado de todas las acciones sobre las cuales se han de realizar alguna operación, y se tienen que incluir los datos siguientes para cada acción:
 - Identificador de la acción
 - Nombre completo de la acción
 - Número de operaciones de compra ejecutadas durante el año.
 - Suma total en euros de las órdenes de compra ejecutadas durante el año sobre la acción.
 - Valor medio en euros de las órdenes de compra ejecutadas durante el año sobre la acción.
 - Número de operaciones de venta ejecutadas durante el año sobre la acción.
 - Suma total en euros de las órdenes de venta ejecutadas sobre la acción.
 - Revalorización de la acción en porcentaje desde el uno de enero del año indicado.

4. Listado de las diez acciones más adquiridas por los sistemas automáticos, ordenadas de forma decreciente según su volumen.

Estos listados se construirán dinámicamente en tiempo de ejecución de la consulta. Hay que tener en cuenta que se valorarán también otros procedimientos y funcionalidades documentados y consensuados.

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 que podemos denominar "*taula_modul_estadistic*".

Los campos de esta tabla estarán pre calculados y no se deberá calcular en el momento con funciones agregadas. Una manera de realizar estos cálculos es a través de *triggers* o llamadas a los procedimientos que actualicen las estadísticas al final de cada procedimiento de alta, baja y modificación. En estos *triggers* o procedimientos de actualización de estadísticas se pueden utilizar funciones agregadas.

El módulo estadístico tiene que dar respuesta inmediata a las siguientes consultas:

1. Dado un identificador de cliente y un mes concreto, anterior al actual, se visualizará el beneficio neto en euros y el porcentaje de su cartera de acciones durante ese mes.
2. Dado el identificador de la acción se visualizará el número de acciones disponibles dentro del *broker* en ese mismo momento; es decir, a suma de todas las acciones que están en órdenes de venta que no están ejecutadas ni canceladas. También el nombre de los clientes que la tienen actualmente en su cartera de forma activa.
3. El cliente con la cartera de acciones con un valor más grande junto con el valor en euros de su cartera.
4. La acción más negociada, comprada y vendida, en el *broker* de forma histórica junto con su volumen.

Las respuestas del módulo estadístico han de ser inmediatas y deben estar siempre actualizadas con la última información de la BD.

Los procedimientos almacenados tienen que cumplir las condiciones siguientes:

- ❖ Como mínimo dispondrán de un parámetro de salida denominado RSP de tipo *string* que indicará si la ejecución ha finalizado con éxito (OK) o ha fracasado ("ERROR + TIPO").
- ❖ Dispondrán de tratamiento de excepciones.
- ❖ Se almacenarán todas las llamadas a procedimientos que se hacen en la tabla *log*, el procedimiento ejecutado y los parámetros de entrada y de salida.

Se han de documentar los procedimientos a través de los correspondientes comentarios y de:

- ❖ Descripción que incluya las acciones que realiza.
- ❖ Describir los tipos y valores posibles de cada parámetro de entrada.
- ❖ Describir los tipos y valores posibles de cada parámetro de salida, y además los diferentes códigos de error que se pueden haber generado.

2. Planificación, Diagrama de Gantt

La planificación detallada del proyecto se muestra en el listado de tareas que forman parte de cada fase del proyecto y también en el Diagrama de Gantt.

LISTADO DE TAREAS:

Inicio del proyecto

Lectura y comprensión del enunciado del proyecto

Definición y estudio del proyecto

Análisis

Identificación de requisitos

Preparación entorno software necesario para esta etapa: Gantt Project, Microsoft Office

Estudio previo fases ciclo de vida.

Descripción y objetivos del proyecto

Identificación de tareas

Planificación del proyecto

Recursos necesarios

Valoración económica

Análisis de riesgos

Elaboración y revisión de documentación PAC1

Entrega PAC1

Diseño

Preparación entorno software necesario para esta etapa: Oracle Express, SGBD

Diseño conceptual BD

Diseño lógico BD

Diseño físico BD

Elaboración y revisión de documentación PAC2

Entrega PAC2

Implementación

Scripts de creación de BD, creación de tablas, índices, disparadores

Insertar datos en la BD

Gestión de clientes y cartera de acciones

Gestión de órdenes de compra/venta

Gestión de las acciones

Funcionalidad de procedimientos de Alta, Baja y Modificación (ABM)

Funcionalidad de procedimientos de consulta

Valoración de otros procedimientos y funcionalidades nuevas

Gestión módulo estadístico

Elaboración y revisión de documentación PAC3

Entrega PAC3

Pruebas

Creación juego de pruebas

Depuración y revisión código

Elaboración y revisión de documentación

Mantenimiento

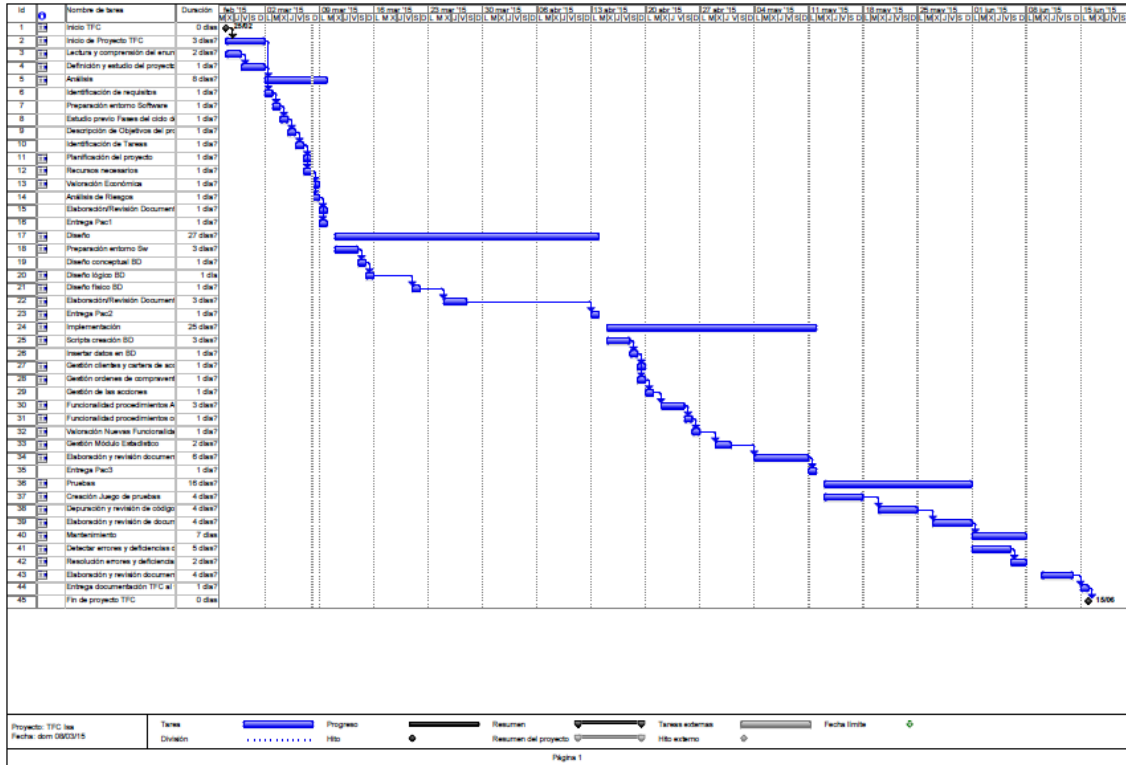
Detectar errores y deficiencias por el uso del programa

Resolución de errores y deficiencias por el uso del programa

Elaboración y revisión de documentación definitiva

Entrega documentación TFC al tribunal

Fin del proyecto



<u>Nombre de Tarea</u>	<u>Comienzo</u>	<u>Fin</u>	<u>Predecesora</u>
Inicio TFC	25/02/2015	25/02/2015	
Inicio de Proyecto TFC	25/02/2015	01/03/2015	1
Lectura y comprensión del enunciado del proyecto	25/02/2015	26/02/2015	
Definición y estudio del proyecto	27/02/2015	01/03/2015	3
Análisis	02/03/2015	09/03/2015	4
Identificación de requisitos	02/03/2015	02/03/2015	2
Preparación entorno Software	03/03/2015	03/03/2015	6
Estudio previo Fases del ciclo de Vida	04/03/2015	04/03/2015	7
Descripción de Objetivos del proyecto	05/03/2015	05/03/2015	8
Identificación de Tareas	06/03/2015	06/03/2015	9
Planificación del proyecto	07/03/2015	07/03/2015	10
Recursos necesarios	07/03/2015	07/03/2015	10
Valoración Económica	08/03/2015	08/03/2015	12
Análisis de Riesgos	08/03/2015	08/03/2015	12
Elaboración/Revisión Documentación PAC1	09/03/2015	09/03/2015	14
Entrega Pac1	09/03/2015	09/03/2015	14
Diseño	11/03/2015	13/04/2015	
Preparación entorno Sw	11/03/2015	13/03/2015	
Diseño conceptual BD	14/03/2015	14/03/2015	18
Diseño lógico BD	15/03/2015	15/03/2015	19
Diseño físico BD	21/03/2015	21/03/2015	20
Elaboración/Revisión Documentación PAC2	25/03/2015	27/03/2015	21
Entrega Pac2	13/04/2015	13/04/2015	22
Implementación	15/04/2015	11/05/2015	
Scripts creación BD	15/04/2015	17/04/2015	
Insertar datos en BD	18/04/2015	18/04/2015	25
Gestión clientes y cartera de acciones	19/04/2015	19/04/2015	26
Gestión ordenes de compraventa	19/04/2015	19/04/2015	26
Gestión de las acciones	20/04/2015	20/04/2015	28
Funcionalidad procedimientos ABM	22/04/2015	24/04/2015	29
Funcionalidad procedimientos consulta	25/04/2015	25/04/2015	30
Valoración Nuevas Funcionalidades	26/04/2015	26/04/2015	31
Gestión Módulo Estadístico	29/04/2015	30/04/2015	32
Elaboración y revisión documentación Pac3	04/05/2015	10/05/2015	33
Entrega Pac3	11/05/2015	11/05/2015	34
Pruebas	13/05/2015	31/05/2015	
Creación Juego de pruebas	13/05/2015	17/05/2015	
Depuración y revisión de código	20/05/2015	24/05/2015	37
Elaboración y revisión de documentación	27/05/2015	31/05/2015	38
Mantenimiento	01/06/2015	07/06/2015	39
Detectar errores y deficiencias del proyecto	01/06/2015	05/06/2015	
Resolución errores y deficiencias	06/06/2015	07/06/2015	41
Elaboración y revisión documentación definitiva	10/06/2015	13/06/2015	
Entrega documentación TFC al tribunal	15/06/2015	15/06/2015	43
Fin de proyecto TFC	15/06/2015	15/06/2015	44

3. Explicación PAC2 y PAC3

En las siguientes entregas del proyecto se realizarán las siguientes tareas:

PAC2: Análisis y Diseño de la base de datos: análisis de requisitos, diseño de la BD a través del Diagrama Entidad Relación, diseño lógico y diseño físico. Documentación y entrega.

PAC3: Implementación de la BD, scripts y pruebas. Documentación y entrega.

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	

4. Recursos Humanos, Técnicos y Materiales

Los recursos necesarios para desarrollar este proyecto son los siguientes:

4.1. Recursos Humanos

Jefe de proyecto:

- Ejecutar los proyectos informáticos asignados por el Director de Proyecto, dirigiendo y coordinando el proyecto de desarrollo y mantenimiento de aplicaciones, supervisando las funciones y recursos de análisis funcional, orgánico y programación, asegurando la adecuada explotación de las aplicaciones
- Capaz de dominar todas las etapas de la vida de un proyecto (análisis de concepción, análisis técnico, programación, pruebas, documentación y formación de usuarios).
- Dirigir el equipo de trabajo compuesto por Analistas Funcionales, Analistas de aplicaciones, Programadores. Gestión de los RRHH de los componentes del proyecto (evaluaciones, desempeño, motivación)
- Control y seguimiento de plazos, indicadores económicos y de calidad.

- Supervisar y coordinar el desarrollo completo de aplicaciones y administrar la introducción de los sistemas de gestión.
- Controlar las aplicaciones en explotación, minimizando las consecuencias negativas sobre las operaciones en producción y desarrollo de aplicaciones
- Analizar y recoger nuevas técnicas y herramientas del mercado estudiando su viabilidad y necesidad. Posibilidad de contratar recursos externos
- Control y Gestión del Desarrollo del Proyecto Informático.
- Redacción, para la Dirección de Informática y para la Dirección del Proyecto de los informes que se precisan para el seguimiento del proyecto.
- Responsabilidad de la gestión económica del proyecto. Elaboración de propuestas, relación con el cliente a nivel de proyecto.

Analista:

- Modelar el software a desarrollar, comprendiendo y considerando los requisitos.
- Coordinará el trabajo con expertos en el área específica que cubra el desarrollo.
- Participará en el proceso de estimación y planificación.
- Análisis de los Nuevos Sistemas Informáticos y de los Cambios en los existentes.
- Diseño de las soluciones informáticas relacionadas con los cambios en los sistemas existentes o con los Nuevos Sistemas
- Dirección y asesoramiento a los Técnicos de Programación en la realización de los programas.
- Creación de los tests de pruebas para verificar que los Sistemas Informáticos cumplen los requisitos y especificaciones de Análisis y Diseño.
- Verificación de la documentación, tanto de los cambios en los sistemas existentes, como la de los nuevos sistemas para garantizar que está completa y al día.
- Asesoramiento a Usuarios, Programadores y Jefe de Estudios en la redacción de la Documentación de Usuario, Instalación y Explotación.
- Dirección del arranque ó "lanzamiento" de un nuevo sistema.
- Asesora al Responsable de Estudios en la elaboración de los criterios que permiten la mejor explotación de los nuevos sistemas.
- Ayuda al Área de Estudios en la resolución de los fallos que se producen en los Sistemas en Producción.

- Evalúa nuevos productos informáticos que pueden aportar mejoras tanto en los sistemas existentes, como para el desarrollo de nuevos sistemas.
- Asesora a los Usuarios para mejor utilizar los Sistemas Existentes.
- Dirige y Coordina el Desarrollo de Reuniones relacionadas con temas que afectan a los Sistemas Informáticos.
- Estudio de Métodos, Técnicas y Herramientas de Análisis y Diseño.
- Estudio de la evolución de las nuevas tecnologías, sobre todo de aquellas que pueden aportar mejoras importantes en los sistemas utilizados en la empresa.

Programador:

- Desempeñará toda tarea de desarrollo que no requiera de la cualificación de un ingeniero de desarrollo.
- Responsabilidad de la correcta funcionalidad de los componentes de negocio que desarrolle
- Desarrollo de aplicaciones. Construir el código que dará lugar al producto resultante en base al diseño realizado, utilizando lenguajes y bases de datos . Generando el código asociado a los procedimientos de migración y carga inicial de datos.
- Realización de las pruebas unitarias y participa en las pruebas de conjunto de la aplicación.
- Configurar y explotar sistemas informáticos.
- Programar bases de datos relacionales no corporativas.
- Desarrollar componentes software en lenguajes de programación.

Administrador de BD:

- Administrar un sistema de bases de datos, interpretando su diseño y estructura, y realizando la adaptación del modelo a los requerimientos del sistema gestor de bases de datos (SGBD), así como la configuración y administración del mismo a nivel físico y lógico, a fin de asegurar la integridad, disponibilidad y confidencialidad de la información almacenada.
- Desarrollo y construcción de las bases de datos. Asegurar la coherencia y la adaptación a las necesidades de la empresa.
- Gestionar las autorizaciones de acceso para los usuarios.
- Asegurar del buen funcionamiento de SGBD y hacer un seguimiento de la utilización de los usuarios a través de las tareas de *mirroring*, *tunning*, desdoblamiento y cualquier técnica futura.

- Participar en la instalación de las herramientas de *Datawarehouse*, herramientas de SIAD, *Data Mining* y cualquiera futura.
- Responsabilidad e de la integridad de los datos y de la existencia de Back-ups.
- Estimación de volúmenes de las estructuras de datos, definiendo mecanismos de migración y carga inicial de datos.
- En producción se ocupa de la gestión y operativa asociada a las bases de datos y al software en el que están implementadas.
- Este perfil es independiente de la tecnología de Base de Datos, jerárquica, relacional, orientada a objetos, nativa XML, o cualquier otra.
- Implantación de las medidas de seguridad (ejemplo reglamentos de desarrollo de la LOPD).
- Este perfil normalmente existen en las grandes empresas y en la Administración, y su responsabilidad es muy alta, ya que la seguridad de los datos es imprescindible. Ejemplo: Hacienda, Hospitales, Defensa, Interior, Banca, etc. En una pequeña y mediana empresa, no suele tener mucho sentido y normalmente no existe como tal

4.2. Recursos Técnicos y Materiales

RECURSOS TÉCNICOS Y MATERIALES:

Ordenador portátil: TOSHIBA con sistema operativo Windows Vista, 150 Gb capacidad disco, 2 Gb de RAM.

Ordenador de sustitución: Portátil HP con 80 Gb de capacidad. Sistema operativo Windows Vista.

El software necesario es:

- ✓ Gantt Project
- ✓ Microsoft Office: Word, Power Point, Excel
- ✓ Adobe Acrobat
- ✓ ArgoUML 0.26.2 - UML Modelling tool
- ✓ Oracle Express: Sistema Gestor Base de Datos Relacionales
- ✓ UltraEdit-32, Toad

5. Análisis de riesgos y controles

En este apartado se analizarán los posibles riesgos que puedan poner en peligro el desarrollo del proyecto y la planificación, así como los controles que se tendrán que aplicar para reducir el impacto negativo en cada fase del ciclo de vida del proyecto.

La definición de riesgo explica que es un evento o condición que cuando ocurre varía los objetivos iniciales del proyecto.

Las fases que forman parte de la gestión del riesgo son: Planificación de riesgos, Identificación de riesgos, Análisis de riesgos y Seguimiento y control de los riesgos.

En general, los principales riesgos para conseguir el objetivo de este proyecto son:

- Causa1: que los requisitos no estén bien definidos.
- Causa2: que el jefe de proyecto no sea muy experto
- Causa3: que el equipo de proyecto no esté bien formado en el entorno de desarrollo del proyecto.

Para prevenir con antelación estos riesgos hay que desarrollar un plan de acciones que se tienen que cumplir antes de iniciar el proyecto. Por lo tanto para las causas anteriores se pueden definir las acciones siguientes:

- Acción1: Si los requisitos no están bien definidos, hay que poner un mayor esfuerzo en esta fase para asegurarse de que son comprensibles y se sabe lo que hay que hacer. Además, se pueden realizar preguntas al cliente/profesor a través de correo electrónico y el foro de la asignatura TFC.
- Acción2: Si el jefe de proyecto no es muy experto se debe buscar una formación adecuada para que obtenga los conocimientos necesarios e incluso la participación anterior en otros proyectos similares.
- Acción3: Si el equipo no está bien formado hay que buscar cursos de formación para que conozca el entorno tecnológico de desarrollo.

Los principales riesgos técnicos para conseguir el objetivo de este proyecto son:

- Causa1: que se produzca una pérdida de datos.
- Causa2: que el ordenador de desarrollo esté defectuoso o se produzca una avería.
- Causa3: caducidad en las licencias de los programas instalados.

Para prevenir con antelación estos riesgos técnicos hay que desarrollar un plan de acciones que se tienen que cumplir antes de iniciar el proyecto. Por lo tanto para las causas anteriores se pueden definir las acciones siguientes:

- Acción1: Para evitar esta pérdida se deben realizar copias de seguridad diarias en otro dispositivo disponible para el proyecto.
- Acción2: Es necesario disponer de un ordenador de sustitución actualizado.
- Acción3: Actualización y chequeo de versiones y licencias del software. Garantizar y controlar un soporte adecuado por parte del proveedor.

También hay que tener en cuenta los riesgos por el desarrollo en paralelo de otras actividades por el mismo recurso que son:

- Causa1: Obligatoriedad de cumplir la jornada laboral de lunes a viernes, de ocho a cinco y media, además de realizar el TFC.
- Causa2: Necesidad de realizar otras prácticas de otra asignatura.
- Causa3: Enfermedad o accidentes.

Para prever estos riesgos se pueden definir las acciones siguientes:

- Acción1: Organizar y gestionar el tiempo a lo largo del día y aprovechar las horas libres de la jornada laboral en la empresa.
- Acción2: Disponibilidad de días de vacaciones de la empresa para desarrollar las prácticas de la asignatura y el TFC.
- Acción3: Posibilidad de modificar a posteriori las fechas de la planificación del proyecto con tiempo suficiente para cumplir el objetivo.

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 del proyecto.	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 los programas	ACCIÓN3	Actualización y chequeo de versiones y licencias

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

6. Valoración Inicial de las horas y coste total del proyecto

Según el Convenio Colectivo Estatal de Empresas de Consultoría y Estudios de mercado y de la opinión pública el importe a aplicar por categorías es el siguiente:

Categorías	Mes (× 14) – Euros	Año – Euros
Grupo I. Titulados		
Titulado/a de grado superior.	109,73	1.536,22
Titulado/a de grado medio.	80,02	1.120,28
Grupo II. Administrativos		
Jefe/a Superior.	82,25	1.151,50
Jefe/a de primera y Cajero/a con firma.	77,80	1.089,20
Jefe/a de segunda y Cajero/a sin firma.	69,16	968,24
Oficial/a de primera.	60,32	844,48
Oficial/a de segunda y Telefonista-recepcionista.	54,83	767,62
Auxiliar, Auxiliar de caja y Telefonista.	48,12	673,68
Aspirantes de dieciséis y diecisiete años.	29,62	414,68
Grupo III. Técnicos de oficina		
Analista y Analista de sistemas.	109,73	1.536,22
Analista-programador/a y Diseñador/a de página Web.	102,72	1.438,08
Programador/a senior, Jefe/a de operación y Programador/a en Internet. .	77,80	1.089,20
Delineante-proyectista.	71,32	998,48
Programador/a junior, Operador/a ordenador, Programador/a máquina auxiliar, Monitor/a de grabación y Técnico/a mantenimiento página Web. .	69,50	973,00
Delineante.	62,39	873,46

http://www.fesugt.es/documentos/pdf/seguros-oficinas/convenios/consultorias/convenio_consultorias_2007_2009_boe.pdf

Esta tabla salarial por categorías se va a utilizar para realizar una valoración inicial de las horas y del coste total del proyecto y según los perfiles necesarios.

	HORAS	IMPORTE €	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

De tal manera que el total del coste inicial detallado queda reflejado en la siguiente tabla:

<u>TAREAS:</u>	<u>RECURSO</u>	<u>HORAS</u>	<u>PRECIO HORA</u>	<u>TOTAL €</u>
Inicio del proyecto				
Lectura y comprensión del enunciado del proyecto	Jefe de Proyecto	1	109,73	109,73
Definición y estudio del proyecto	Jefe de Proyecto	5	109,73	548,65
Análisis				
Identificación de requisitos	Analista	1	102,72	102,72
Preparación entorno software necesario para esta etapa: Gantt Project, Microsoft Office	Analista	1	102,72	102,72
Estudio previo fases ciclo de vida.	Jefe de Proyecto	1	109,73	109,73
Descripción y objetivos del proyecto	Jefe de Proyecto	2	109,73	219,46
Identificación de tareas	Jefe de Proyecto	2	109,73	219,46
Planificación del proyecto	Jefe de Proyecto	3	109,73	329,19
Recursos necesarios	Jefe de Proyecto	2	109,73	219,46
Valoración económica	Jefe de Proyecto	2	109,73	219,46
Análisis de riesgos	Jefe de Proyecto	2	109,73	219,46
Elaboración y revisión de documentación PAC1	Jefe de Proyecto	3	109,73	329,19
Entrega PAC1	Jefe de Proyecto	1	109,73	109,73
Diseño				
Preparación software necesario para esta etapa: Oracle Express, SGBD	Administrador BD	5	80,02	400,10
Diseño conceptual BD	Analista	7	102,72	719,04
Diseño lógico BD	Analista	6	102,72	616,32
Diseño físico BD	Analista	7	102,72	719,04
Elaboración y revisión de documentación PAC2	Analista	3	102,72	308,16
Entrega PAC2	Jefe de Proyecto	1	109,73	109,73
Implementación				
Scripts de creación de BD, creación de tablas, índices, disparadores	Programador	7	77,8	544,60

Insertar datos en la BD	Programador	5	77,8	389,00
Gestión de clientes y cartera de acciones	Programador	2	77,8	155,60
Gestión de órdenes de compra/venta	Programador	2	77,8	155,60
Gestión de las acciones	Programador	2	77,8	155,60
Funcionalidad de procedimientos de Alta, Baja y Modificación (ABM)	Programador	6	77,8	466,80
Funcionalidad de procedimientos de consulta	Programador	6	77,8	466,80
Valoración de otros procedimientos y funcionalidades nuevas	Programador	9	77,8	700,20
Gestión módulo estadístico	Programador	9	77,8	700,20
Elaboración y revisión de documentación PAC3	Analista	5	102,72	513,60
Entrega PAC3	Jefe de Proyecto	1	109,73	109,73
Pruebas				
Creación juego de pruebas	Analista	12	102,72	1.232,64
Depuración y revisión código	Programador	15	77,8	1.167,00
Elaboración y revisión de documentación	Analista	15	102,72	1.540,80
Mantenimiento				
Detectar errores y deficiencias por el uso del programa	Analista	10	102,72	1.027,20
Resolución de errores y deficiencias por el uso del programa	Programador	12	77,8	933,60
Elaboración y revisión de documentación definitiva				
Entrega documentación TFC al tribunal	Jefe de Proyecto	13	109,73	1.426,49
Fin del proyecto	Jefe de Proyecto	1	109,73	109,73
TOTALES HORAS/EUROS		187		17.506,54

7. Preparación Entorno Software necesario: Oracle Express, SGBD

ORACLE EXPRESS:

Para la instalación de Oracle Express y Oracle Developer se ha realizado el registro en la Web de Oracle para realizar la descarga del software.

Se tiene en cuenta que el precio de las acciones se almacena en euros pero puede añadirse una nueva funcionalidad y almacenar este valor también en dólares, de tal manera que en la web del Banco de España aparece el cambio.

Actualización automática 6x									
0 : 10									
	USD	EUR	GBP	INR	AUD	CAD	ZAR	NZD	JPY
1 USD	1.00000	0.94313	0.68315	62.2720	1.30234	1.25720	11.9956	1.32811	120.175
Inverso:	1.00000	1.06030	1.46380	0.01606	0.76785	0.79542	0.08336	0.75295	0.00832
1 EUR	1.06030	1.00000	0.72435	66.0269	1.38087	1.33301	12.7189	1.40819	127.421
Inverso:	0.94313	1.00000	1.38055	0.01515	0.72418	0.75018	0.07862	0.71013	0.00785

Tipos de cambio intermedio: 2015-04-12 10:28 UTC

<http://www.bde.es/webbde/es/estadis/infoest/tipos/tipos.html>

1 Tipos de cambio (datos diarios)

1.1 Cambios oficiales del euro del Banco Central Europeo (1)

Los datos mensuales corresponden al último día del mes

Unidades monetarias por euro

	1	2	3	4	5	6	7	8	9	10	11
	Dólar estadounidense (USD)	Yen japonés (JPY)	Franco suizo (CHF)	Libra esterlina (GBP)	Corona sueca (SEK)	Corona danesa (DKK)	Corona noruega (NOK)	Dólar canadiense (CAD)	Dólar australiano (AUD)	Dólar neozelandés (NZD)	Pro memoria DEG (XDR)
14 Nov	1,2483	147,69	1,2018	0,79530	9,2660	7,4409	8,6975	1,4227	1,4647	1,5879	0,8520
Dic	1,2141	145,23	1,2024	0,77890	9,3930	7,4453	9,0420	1,4063	1,4829	1,5525	0,8386
15 Ene	1,1305	133,08	1,0468	0,75110	9,3612	7,4440	8,8335	1,4323	1,4535	1,5568	0,8019
Feb	1,1240	134,05	1,0636	0,72780	9,3693	7,4660	8,5740	1,3995	1,4358	1,4849	0,7986
Mar	1,0759	128,95	1,0463	0,72730	9,2901	7,4697	8,7035	1,3738	1,4154	1,4388	0,7794
Abr	1,0774	129,22	1,0447	0,72680	9,3421	7,4717	8,6620	1,3493	1,3951	1,4188	0,7832
15 Mar 20	1,0776	130,26	1,0552	0,72550	9,3006	7,4516	8,7125	1,3655	1,3988	1,4401	0,7825
21	-	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-	-
23	1,0912	130,80	1,0561	0,73180	9,2804	7,4555	8,6275	1,3667	1,3938	1,4314	0,7865
24	1,0950	130,89	1,0491	0,73440	9,2889	7,4603	8,5935	1,3649	1,3843	1,4301	0,7870
25	1,0985	131,13	1,0517	0,73650	9,3026	7,4688	8,5930	1,3711	1,3925	1,4322	0,7893
26	1,0973	130,67	1,0498	0,73710	9,3250	7,4678	8,6005	1,3653	1,3990	1,4429	0,7879
27	1,0856	129,39	1,0476	0,72980	9,3178	7,4672	8,6450	1,3580	1,3940	1,4342	0,7831
28	-	-	-	-	-	-	-	-	-	-	-
29	-	-	-	-	-	-	-	-	-	-	-
30	1,0845	130,02	1,0439	0,73280	9,3207	7,4690	8,6545	1,3709	1,4158	1,4417	0,7839
31	1,0759	128,95	1,0463	0,72730	9,2901	7,4697	8,7035	1,3738	1,4154	1,4388	0,7794
Abr 1	1,0755	129,29	1,0426	0,72850	9,2541	7,4708	8,6695	1,3632	1,4153	1,4472	0,7795
2	1,0830	129,59	1,0400	0,73160	9,3445	7,4708	8,6506	1,3693	1,4343	1,4499	0,7831
3	-	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	-	-	-	-	-	-	-	-
7	1,0847	130,33	1,0438	0,72860	9,3740	7,4714	8,7300	1,3563	1,4192	1,4423	0,7829
8	1,0862	130,09	1,0438	0,72640	9,3595	7,4717	8,6766	1,3487	1,4078	1,4315	0,7832
9	1,0774	129,22	1,0447	0,72680	9,3421	7,4717	8,6620	1,3493	1,3951	1,4188	-

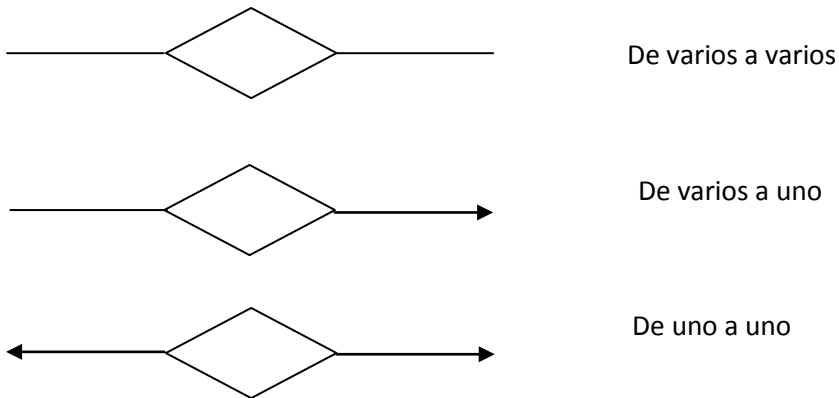
8. Diseño Conceptual:

8.1. Diagrama Entidad Relación:

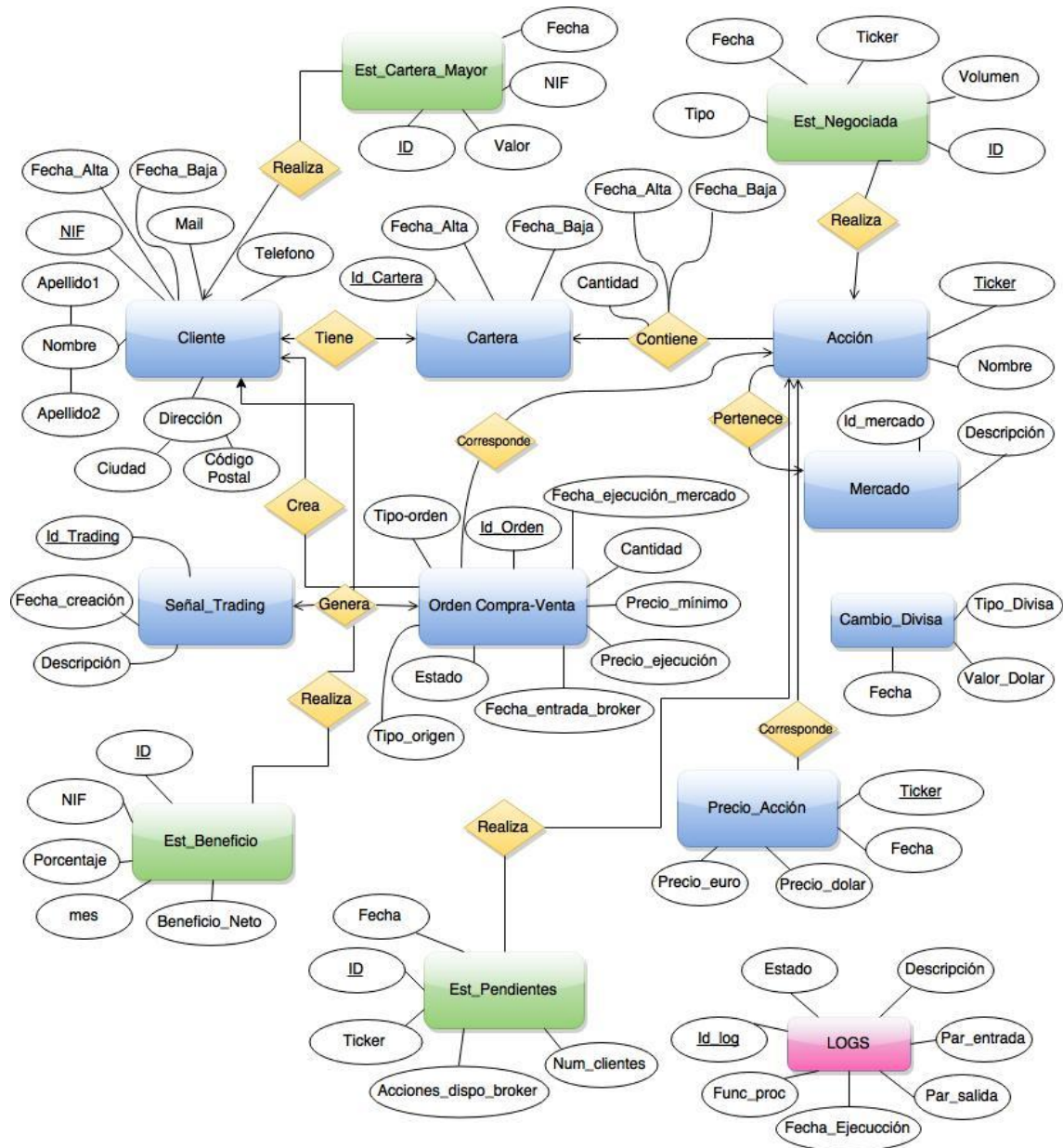
En el modelo de datos (E/R) se parte de una situación real a partir de la cual se definen entidades y relaciones entre dichas entidades que representa un objeto del que se quiere almacenar información. La simbología de entidades y atributos del diagrama son las siguientes:

Atributos: simples, compuestos, monovalorado, multivalorados, derivados

Relaciones:



A continuación se muestra el Diagrama Entidad/Relación de nuestro sistema:



Entidad: CLIENTE

Nombre atributo	Tipo
NIF	Clave primaria
Nombre	Compuesto
Apellido1	
Apellido2	
Dirección	Compuesto
Código Postal	
Ciudad	
Teléfono	Simple
Email	Simple
Fecha_alta	Simple
Fecha_baja	Simple

Entidad: CARTERA

Nombre atributo	Tipo
Id_cartera	Clave primaria
Fecha_Alta	Simple
Fecha_baja	Simple

Entidad: MERCADO

Nombre atributo	Tipo
Id_mercado	Simple
Descripción	Simple

Entidad: ACCION

Nombre atributo	Tipo
Ticker	Clave primaria
Nombre	Simple

Entidad: PRECIO_ACCION

Nombre atributo	Tipo
Ticker	Clave primaria
Precio_euro	Simple
Precio_dolar	Simple
Fecha	Simple

Entidad: CAMBIO_DIVISA

Nombre atributo	Tipo
Tipo_divisa	Clave primaria

Fecha	Clave primaria
Valor_Dólar	Simple

Entidad: ORDEN_COMPRA_VENTA

Nombre atributo	Tipo
Id_orden	Clave primaria
Tipo_orden	Compra, venta
Tipo_origen	Manual, automática
Fecha_entrada_broker	Simple
Estado	Pendiente, ejecutada, cancelada
Fecha_ejec_mercado	Simple
Cantidad	Simple
Precio_mínimo	Simple
Precio_ejecución	Simple

Entidad: SEÑALES_TRADING

Nombre atributo	Tipo
Id_trading	Clave primaria
Descripción	Simple
Fecha_creación	Simple

Entidad: EST_BENEFICIO

Nombre atributo	Tipo
ID	Clave primaria
Beneficio_Neto	Calculado
Porcentaje	Calculado
Mes	Simple
NIF	Simple

Entidad: EST_PENDIENTES

Nombre atributo	Tipo
ID	Clave primaria
Acciones_dispo_broker	Simple
Num_Clientes	Simple
Fecha	Simple
Ticker	Simple

Entidad: EST_NEGOCIADA

Nombre atributo	Tipo
ID	Clave primaria
Volumen	Simple

Fecha	Simple
Tipo	Simple
Ticker	Simple

Entidad: EST_CARTERA_MAYOR:

Nombre atributo	Tipo
ID	Clave primaria
Fecha	Simple
Valor	Simple
NIF	Simple

Entidad: LOGS:

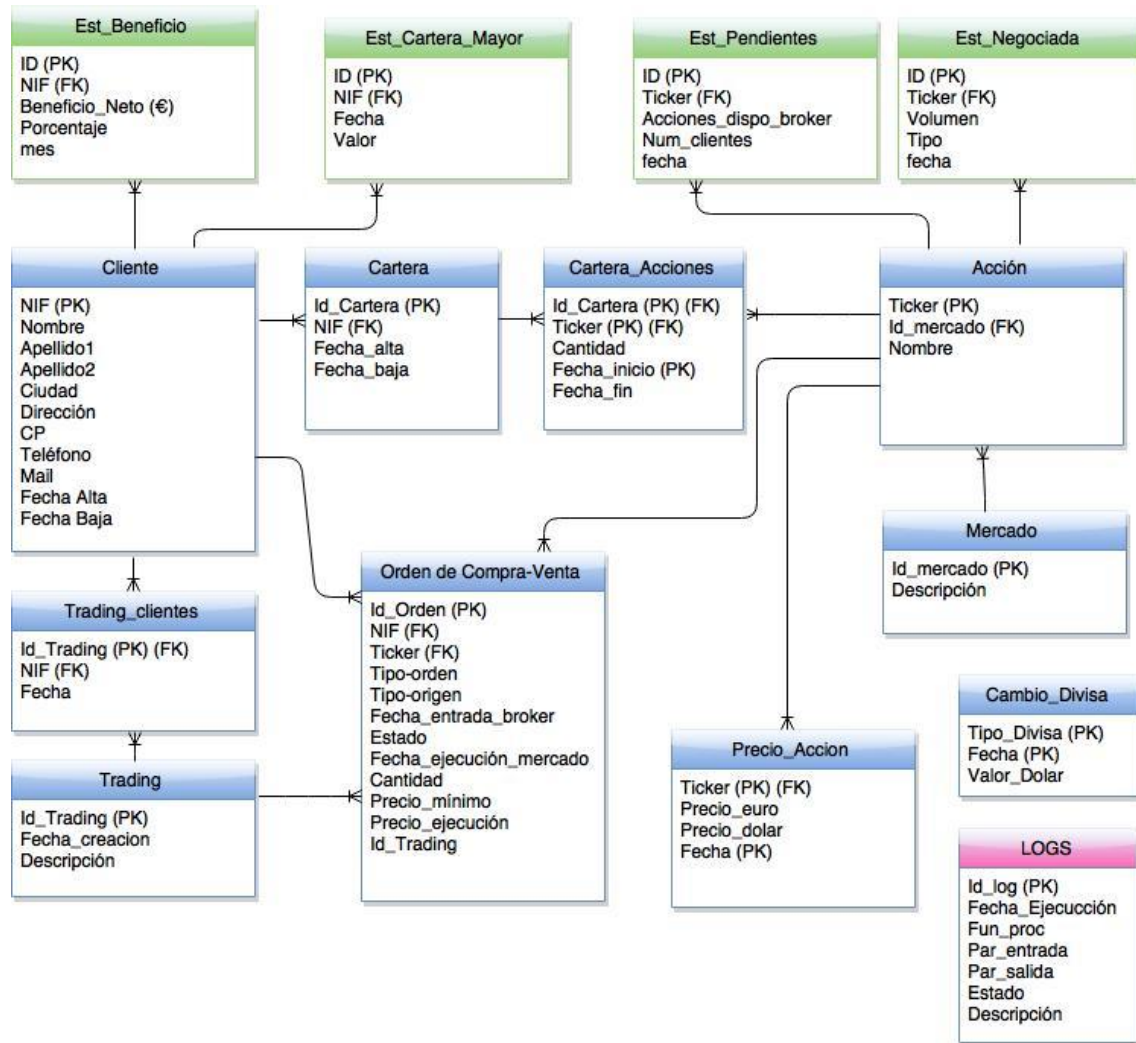
Nombre atributo	Tipo
ID_log	Clave primaria
Fecha_ejecución	Simple
Func_proc	Simple
Par_entrada	Simple
Par_salida	Simple
Estado	Simple
descripción	Simple

9. Diseño Lógico:

El diseño lógico parte del diseño conceptual y su objetivo es conseguir el conjunto de tablas que forman la base de datos.

9.1. Diagrama Lógico Relacional:

A continuación se muestra el Diagrama Relacional formado por las tablas de la base de datos de nuestro sistema:



- Tabla: **CLIENTE**

Nombre atributo	Tipo	Observaciones
NIF	Varchar2(9)	Clave primaria
Nombre	Varchar2(20)	
Apellido1	Varchar2(30)	
Apellido2	Varchar2(30)	
Ciudad	Varchar2(30)	
Dirección	Varchar2(30)	
CP	Number (6)	
Teléfono	Number(12)	
Email	Varchar2(15)	
Fecha_alta	Date	
Fecha_baja	Date	

- Tabla: **CARTERA**

Nombre atributo	Tipo	Observaciones
Id_cartera	Varchar2(9)	Clave primaria
NIF	Varchar2(9)	Clave ajena (CLIENTE)
Fecha_Alta	Date	
Fecha_baja	Date	

- Tabla: **MERCADO**

Nombre atributo	Tipo	Observaciones
Id_mercado	VARCHAR2(15)	Clave primaria
Descripción	Varchar2(100)	

- Tabla: **ACCION**

Nombre atributo	Tipo	Observaciones
Ticker	Varchar2(15)	Clave primaria
Id_mercado	Number(4)	Clave ajena (MERCADO)
Nombre	Varchar2(15)	

- Tabla: **PRECIO_ACCION**

Nombre atributo	Tipo	Observaciones
Ticker	Varchar2(15)	Clave primaria, Clave ajena (ACCION)
Fecha	Date	Clave primaria
Precio_euro	Number(6,2)	
Precio_dolar	Number(6,2)	

- Tabla: **CARTERA_ACCION**

Nombre atributo	Tipo	Observaciones
Id_cartera	Varchar2(9)	Clave primaria, Clave ajena (CARTERA)
Ticker	Varchar2(15)	Clave primaria, Clave ajena (ACCIÓN)
Cantidad	Number(10)	
Fecha_inicio	Date	Clave primaria
Fecha_fin	Date	

- Tabla: **CAMBIO_DIVISA**

Nombre atributo	Tipo	Observaciones
Tipo_divisa	Varchar2(10)	Clave primaria
Fecha	Date	Clave primaria
Valor_Dólar	Number(6,2)	

La tabla CAMBIO_DIVISA se actualiza con los importes de cierre diario según el Banco de España. Para poder implementar una funcionalidad adicional o mejora se ha decidido almacenar en la tabla el importe necesario para realizar la correspondencia del valor euro y dólar de las acciones.

- Tabla: **ORDEN_COMPRA_VENTA**

Nombre atributo	Tipo	Observaciones
Id_orden	Varchar2(9)	Clave primaria
NIF	Varchar2(9)	Clave ajena (CLIENTE)
Ticker	Varchar2(15)	Clave ajena (ACCION)
Id_trading	Varchar2(9)	Clave ajena (TRADING)
Tipo_orden	Varchar2(10)	'VENTA', 'COMPRA'
Tipo_origen	Varchar2(12)	'MANUAL', 'AUTOMÁTICA'
Fecha_entrada_broker	Date	
Estado	VARCHAR2(10)	'EJECUTADA', 'CANCELADA', 'PENDIENTE'
Fecha_ejec_mercado	Date	
Cantidad	Number(10)	
Precio_mínimo	Number(4,5)	Opcional
Precio_ejecución	Number(4,5)	

- Tabla: **TRADING**

Nombre atributo	Tipo	Observaciones
Id_trading	VARCHAR2(9)	Clave primaria
Descripción	Varchar2(35)	
Fecha_creación	Date	

- Tabla: **TRADING_CLIENTES**

Nombre atributo	Tipo	Observaciones
Id_trading	VARCHAR2(9)	Clave primaria
NIF	Varchar2(9)	Clave ajena (CLIENTE)
Fecha	Date	

- Tabla: **EST_BENEFICIO**

Nombre atributo	Tipo	Observaciones
ID	Number	Clave primaria
NIF	Varchar2(9)	Clave ajena (CLIENTE)
Beneficio_neto	Number(10,2)	
Porcentaje	Number(6,2)	
Mes	Date	

La tabla EST_BENEFICIO, que pertenece al “módulo estadístico”, se actualiza el último día del mes según la fecha de cierre del mercado para almacenar el beneficio neto en euros y el porcentaje de la cartera de acciones de un cliente.

- Tabla: **EST_PENDIENTES**

Nombre atributo	Tipo	Observaciones
ID	Number	Clave primaria
Fecha	Date	
Ticker	Varchar2(10)	Clave ajena (ACCIÓN)
Acciones_dispo_broker	Number(9)	
Num_clientes	Number(9)	

La tabla EST_PENDIENTES, que pertenece al “módulo estadístico”, se actualiza cuando se insertan en la tabla ORDEN_COMPRA_VENTA órdenes de venta de tipo pendiente. Contiene el número de acciones disponibles y el número de clientes que tienen esa acción en su cartera de forma activa.

- Tabla: **EST_CARTERA_MAYOR**

Nombre atributo	Tipo	Observaciones
ID	Number	Clave primaria
NIF	Varchar2(9)	Clave ajena (CLIENTE)
Valor	Varchar2(10)	
Fecha	Date	

La tabla EST_CARTERA_MAYOR, que pertenece al “módulo estadístico”, se actualiza cuando se ejecutan órdenes de compra/venta. Tiene que almacenar el cliente con la cartera de acciones con mayor valor. Se puede incluir un campo fecha para tener un histórico.

- Tabla: EST_NEGOCIADA

Nombre atributo	Tipo	Observaciones
ID		Clave primaria
Ticker	Varchar2(10)	Clave ajena (ACCION)
Volumen	Number(10)	
Tipo	Varchar2(10)	'COMPRADA','VENDIDA'
Fecha	Date	

La tabla EST_NEGOCIADA, que pertenece al "módulo estadístico", se actualiza cuando se ejecutan órdenes de compra/venta. Tiene que almacenar la acción más negociada y su volumen. Se puede incluir un campo fecha para tener un histórico.

- Tabla: LOGS

Nombre atributo	Tipo	Observaciones
Id_log	Varchar2(10)	Clave primaria
Fecha_ejecución	Date	
Func_proc	Varchar2(30)	
Par_entrada	Varchar2(50)	
Par_salida	Varchar2(50)	
Estado	Varchar2(10)	'ERROR', 'OK'
Descripción	Varchar2(500)	

La tabla LOGS se actualiza cuando se ejecutan procedimientos en la base de datos del sistema para poder realizar un control posterior.

10. Diseño Físico:

10.1. Tablas:

Se realiza el script de creación de las tablas de la base de datos según los requisitos y el diseño lógico obtenido como aplicación de la normalización al diagrama entidad/relación. Se debe respetar un orden estricto para evitar errores en las claves ajenas que relacionan cada tabla.

10.2. Paquetes, procedimientos y funciones:

Los procedimientos a realizar de ABM (Alta, Baja, Modificaciones) son de las órdenes de compra/venta, de los clientes, de las carteras de inversión y el valor de cierre diario de las acciones. También se realiza un procedimiento que actualice el valor del dólar/euro para almacenar el precio en otra divisa.

En general cada tabla debe tener los procedimientos adecuados para su actualización:

- PAQ_CLIENTE: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CLIENTE.

- F_ALTA_CLIENTE:
Descripción: Inserta en la tabla CLIENTE los datos correspondientes
Parámetros de entrada: NIF, Nombre, Apellido1, Apellido2, Domicilio, Teléfono, Mail.
Parámetros de salida:
Validaciones: Comprobación de la letra del NIF correcto.
- F_BAJA_CLIENTE:
Descripción: Realiza la baja en la tabla CLIENTE los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.
Parámetros de entrada: NIF
Parámetros de salida: Datos borrados
Validaciones:
- F_MODIFIC_CLIENTE:
Descripción: Realiza las modificaciones en los datos de la tabla CLIENTE.
Parámetros de entrada: Nombre, Apellido1, Apellido2, Domicilio, Teléfono, Mail.
Parámetros de salida: Indica al usuario los datos que se han modificado.
Validaciones: Que exista el NIF
- PAQ_CARTERA_ACCIONES: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CARTERA_ACCIONES.
 - F_ALTA_CARTERA:
Descripción: Inserta en la tabla CARTERA_ACCIONES los datos correspondientes
Parámetros de entrada: Id_cartera, NIF, Fecha_alta, Fecha_baja
Parámetros de salida:
Validaciones: Que exista el NIF.
 - F_BAJA_CARTERA:
Descripción: Realiza la baja en la tabla CARTERA_ACCIONES los datos correspondientes.
Parámetros de entrada: Id_cartera, fecha_baja
Parámetros de salida: Datos borrados
Validaciones: Fecha de baja mayor que la fecha de alta. Que exista el Id_cartera.
 - F_MODIFIC_CARTERA:
Descripción: Realiza las modificaciones en los datos de la tabla CARTERA_ACCIONES.

Parámetros de entrada: Id_cartera, NIF, fecha_alta, fecha_baja

Parámetros de salida: Indica al usuario los datos que se han modificado.

Validaciones: Que exista el NIF, Id_cartera, fecha_baja mayor o igual que la fecha de alta.

- PAQ_ACCIÓN: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla ACCIÓN.

- F_ALTA_ACCIÓN:

Descripción: Inserta en la tabla ACCIÓN los datos correspondientes

Parámetros de entrada: Ticker, Id_mercado, Nombre, Id_cartera, cantidad

Parámetros de salida:

Validaciones: Que exista el id_mercado, id_cartera.

- F_BAJA_ACCIÓN:

Descripción: Realiza la baja en la tabla ACCIÓN los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.

Parámetros de entrada: Ticker

Parámetros de salida: Datos borrados

Validaciones:

- F_MODIFIC_ACCIÓN:

Descripción: Realiza las modificaciones en los datos de la tabla ACCIÓN.

Parámetros de entrada: Ticker, Id_mercado, Nombre, Id_cartera, cantidad

Parámetros de salida: Indica al usuario los datos que se han modificado.

Validaciones: Que exista el Ticker, Id_mercado, Nombre, Id_cartera

- F_CONSULT_ACCIÓN:

Descripción: Realiza las consultas necesarias en los datos de la tabla ACCIÓN.

Parámetros de entrada: Ticker, Id_mercado, Nombre, Id_cartera, cantidad

Parámetros de salida: Se visualizan los datos de las carteras

Validaciones: Que exista el Ticker, Id_mercado

- PAQ_MERCADO: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla MERCADO.

- F_ALTA_MERCADO:

Descripción: Inserta en la tabla MERCADO los datos correspondientes

Parámetros de entrada: Id_mercado, descripción

Parámetros de salida:

Validaciones:

- F_BAJA_MERCADO:
Descripción: Realiza la baja en la tabla MERCADO los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.
Parámetros de entrada: Id_mercado
Parámetros de salida: Datos borrados
Validaciones: Que no exista mercado vigente en la tabla ACCIÓN
- F_MODIFIC_MERCADO:
Descripción: Realiza las modificaciones en los datos de la tabla MERCADO.
Parámetros de entrada: Id_mercado
Parámetros de salida: Indica al usuario los datos que se han modificado.
Validaciones: Que exista id_mercado
- PAQ_PRECIO: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla PRECIO_ACCIÓN.
 - F_ALTA_PRECIO:
Descripción: Inserta en la tabla PRECIO_ACCIÓN los datos correspondientes
Parámetros de entrada: Ticker, precio_euro, precio_dólar, fecha
Parámetros de salida:
Validaciones: Que exista el id_ticker, se inserta una fila con el precio de cierre del día en el mercado. Se actualiza precio_dólar con la tabla CAMBIO_DIVISA.
 - F_BAJA_PRECIO:
Descripción: Realiza la baja en la tabla PRECIO_ACCIÓN los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.
Parámetros de entrada: Ticker
Parámetros de salida: Datos borrados
Validaciones:
 - F_MODIFIC_PRECIO:
Descripción: Realiza las modificaciones en los datos de la tabla PRECIO_ACCIÓN.
Parámetros de entrada: Ticker, Fecha, precio_euro, precio_dólar
Parámetros de salida: Indica al usuario los datos que se han modificado.
Validaciones: Que exista el Ticker

- PAQ_DIVISA: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CAMBIO_DIVISA.
 - F_ALTA_DIVISA:
Descripción: Inserta en la tabla CAMBIO_DIVISA los datos correspondientes
Parámetros de entrada: Tipo_divisa, Fecha, Valor_Dólar.
Parámetros de salida:
Validaciones:
 - F_BAJA_DIVISA:
Descripción: Realiza la baja en la tabla CAMBIO_DIVISA los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.
Parámetros de entrada: Tipo_divisa
Parámetros de salida: Datos borrados
Validaciones:
 - F_MODIFIC_DIVISA:
Descripción: Realiza las modificaciones en los datos de la tabla CAMBIO_DIVISA.
Parámetros de entrada: Tipo_divisa, Fecha, Valor_Dólar.
Parámetros de salida: Indica al usuario los datos que se han modificado.
Validaciones: La modificación de la divisa es diaria y se guarda el valor correspondiente para realizar la conversión del precio_dólar de la tabla de PRECIO_ACCIÓN.
- PAQ_ORDEN_CV: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla ORDEN_COMPRA_VENTA.
 - F_ALTA_ORDEN_CV:
Descripción: Inserta en la tabla ORDEN_COMPRA_VENTA los datos correspondientes
Parámetros de entrada: Id_orden, NIF, Ticker, Tipo_orden, Tipo_origen, Fecha_entrada_broker, Estado, Fecha_ejecución_mercado, Cantidad, Precio mínimo, Precio_ejecución
Parámetros de salida:
Validaciones: Que exista NIF, Ticker
 - F_BAJA_ORDEN_CV:
Descripción: Realiza la baja en la tabla ORDEN_COMPRA_VENTA los datos correspondientes. Realmente es poner el campo estado en "cancelada".
Parámetros de entrada: Id_orden, NIF, Ticker

Parámetros de salida: Datos borrados

Validaciones: Que exista NIF, Ticker, id_orden

- F_CONSULT_ORDEN_CV:

Descripción: Realiza las consultas necesarias en los datos de la tabla ORDEN_COMPRA_VENTA.

Parámetros de entrada: Id_orden, NIF, Ticker.

Parámetros de salida: Se visualizan los datos de las órdenes compra/venta

Validaciones: Que exista NIF, Ticker, id_orden

- PAQ_TRADING: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla SEÑALES_TRADING.

- F_ALTA_TRADING:

Descripción: Inserta en la tabla SEÑALES_TRADING los datos correspondientes

Parámetros de entrada: Id_trading, Id_orden, Fecha_creación, descripción

Parámetros de salida:

Validaciones:

- F_BAJA_TRADING:

Descripción: Realiza la baja en la tabla SEÑALES_TRADING los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja.

Parámetros de entrada: Id_trading

Parámetros de salida: Datos borrados

Validaciones:

- F_MODIFIC_TRADING:

Descripción: Realiza las modificaciones en los datos de la tabla SEÑALES_TRADING.

Parámetros de entrada: Id_trading

Parámetros de salida: Indica al usuario los datos que se han modificado.

Validaciones: Que exista id_trading

- PAQ_LOG: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla LOGS.

- F_ALTA_LOG:

Descripción: Inserta en la tabla LOGS los datos correspondientes

Parámetros de entrada: Id_log, Fecha_ejecución, Fun_proc, Par_entrada, Par_salida, Estado

Parámetros de salida:

Validaciones: El sistema de de alta los datos de la tabla LOG cada vez que se ejecuta un procedimiento o función en el sistema.

- F_BAJA_LOG:
Descripción: Realiza la baja en la tabla LOGS los datos correspondientes. En estudio el procedimiento adecuado que implique insertar un campo adicional de fecha de baja o se borre la tabla cada cierto tiempo.
Parámetros de entrada: Id_log
Parámetros de salida: Datos borrados
Validaciones: Que exista id_log

- F_MODIFIC_LOG:
Descripción: Realiza las modificaciones en los datos de la tabla LOGS.
Parámetros de entrada: Id_log
Parámetros de salida: Indica al usuario los datos que se han modificado.
Validaciones: Que exista id_log

10.3. Funciones y procedimientos para confeccionar listados:

Para cumplir con el requisito de construir ciertos listados dinámicamente, es necesaria la creación de varias funciones que utilicen la sentencia SELECT y sus funciones de agregados cuando sea necesario.

- **Descripción:** Listado de la composición de la cartera de acciones vigente del cliente y su valor en euros de las acciones del momento.
- **Parámetros entrada:** Fecha, identificador del cliente (NIF).
- **Parámetros salida:** identificador del cliente (NIF), identificador de cartera (id_cartera), valor euros (precio acción * cantidad)
- **Tablas:** CARTERA_ACCION, ACCION, PRECIO_ACCION, LOGS

- **Descripción:** Listado de todos los clientes que han realizado operaciones durante este año.
- **Parámetros entrada:** Año.
- **Parámetros salida:** identificador del cliente (NIF), nombre del cliente (Nombre, apellido1, apellido2), número de órdenes de compra cargadas durante ese año, número de órdenes de venta cargadas durante ese año, número de órdenes de compra ejecutadas durante ese año, sumatorio en euros de todas las órdenes de compra ejecutadas durante ese año, valor medio en euros de cada operación de compra ejecutada por el cliente durante ese año, número de órdenes de venta ejecutadas durante ese año, sumatorio en euros de todas las órdenes de venta ejecutadas durante ese año, valor medio en euros de cada operación de venta ejecutada por el cliente durante ese año, número de órdenes de compra canceladas durante ese año, número de órdenes de venta canceladas durante ese año.
- **Tablas:** CLIENTE, ORDEN_COMPRA_VENTA, LOGS

- **Descripción:** Listado de todas las acciones sobre las cuales se han realizado alguna operación.
 - **Parámetros entrada:** Año
 - **Parámetros salida:** Identificador de la acción (ticker), nombre completo de la acción (nombre), número de las operaciones de compra ejecutadas durante ese año, suma total en euros de las órdenes de compra ejecutadas durante el año sobre la acción, valor medio en euros de las órdenes de compra ejecutadas durante ese año sobre la acción, número de las operaciones de venta ejecutadas durante ese año sobre la acción, suma total en euros de las órdenes de venta ejecutadas sobre la acción, revalorización de la acción en porcentaje desde el uno de enero del año indicado.
 - **Tablas:** ACCIÓN, ORDEN_COMPRA_VENTA, LOGS
-
- **Descripción:** Listado de las diez acciones más adquiridas por los sistemas automáticas, ordenados de forma decreciente por su volumen.
 - **Parámetros entrada:** No es necesario.
 - **Parámetros salida:** Identificador de la acción (ticker), nombre completo de la acción (nombre)
 - **Tablas:** ACCIÓN, ORDEN_COMPRA_VENTA, LOGS

10.4. Disparadores:

Las tablas que forman parte del módulo estadístico se actualizan a través de triggers:

- ❖ Para actualizar la tabla EST_CARTERA_MAYOR se crea un disparador denominado **TR_MAYOR** que pueda actualizar los campos que forman parte de esta tabla: ID, NIF, Fecha, valor. Esta tabla contiene el cliente con la cartera de acciones más grande y su valor. Por lo tanto cuando se ejecute una orden de compra/venta se tienen que actualizar los datos de la tabla EST_CARTERA_MAYOR. Se comparan todas las carteras con su valor y se insertan los datos de la mayor.
- ❖ Para actualizar la tabla EST_NEGOCIADA se crea un disparador denominado **TR_NEGOCIADA** que actualiza los campos de esta tabla: ID, Ticker, volumen, fecha, tipo. Esta tabla contiene la acción más negociada comprada y vendida en el *broker* junto con su volumen. Esto ocurre cuando se produce alguna actividad en la tabla ORDEN_COMPRA_VENTA. De esta manera, se tiene que calcular el número de veces que aparece el ticker de la acción según las órdenes ejecutadas en el *broker* y según la fecha.
- ❖ Para actualizar la tabla EST_PENDIENTES se crea un disparador denominado **TR_PENDIENTES** que actualiza los campos de esta tabla: ID, Ticker, Acciones_dispo_broker, Num_clientes. Esta tabla contiene el número de acciones disponibles dentro del broker, es decir cuando se produce algún cambio en la tabla ORDEN_COMPRA_VENTA se ejecuta este *trigger*. Para saber el número de clientes

que contienen esa acción se debe consultar en las tablas CARTERA_ACCIONES y ACCIONES.

- ❖ Para actualizar la tabla EST_BENEFICIO se crea un disparador denominado **TR_BENEFICIO** que actualiza los campos de esta tabla: ID, NIF, Beneficio_neto, porcentaje, mes. Esta tabla contiene el beneficio neto en euros y el porcentaje de su cartera de acciones durante un mes de un cliente y un mes concreto. El campo beneficio de esta tabla contiene la diferencia entre el precio de compra y el precio actual de las acciones de la cartera del cliente. El precio de la acción está en la tabla PRECIO_ACCIÓN por fecha. El campo porcentaje de la tabla EST_BENEFICIO se calcula en función del beneficio neto obtenido.

11. IMPLEMENTACIÓN:

A partir del lenguaje *SQL* y *PL/SQL* y *Oracle Developer* se ha desarrollado el código necesario para dar soluciones a los requisitos solicitados del proyecto.

Para realizar la implementación de este proyecto se va a distinguir varias fases:

- Creación de las tablas de la base de datos.
- Restricciones de las tablas
- Inserción y modificación de datos de las tablas.
- Paquetes de AMB.
- Procedimientos de consultas para generar datos solicitados.
- Módulo estadístico: disparadores y actualizaciones de las tablas estadísticas.

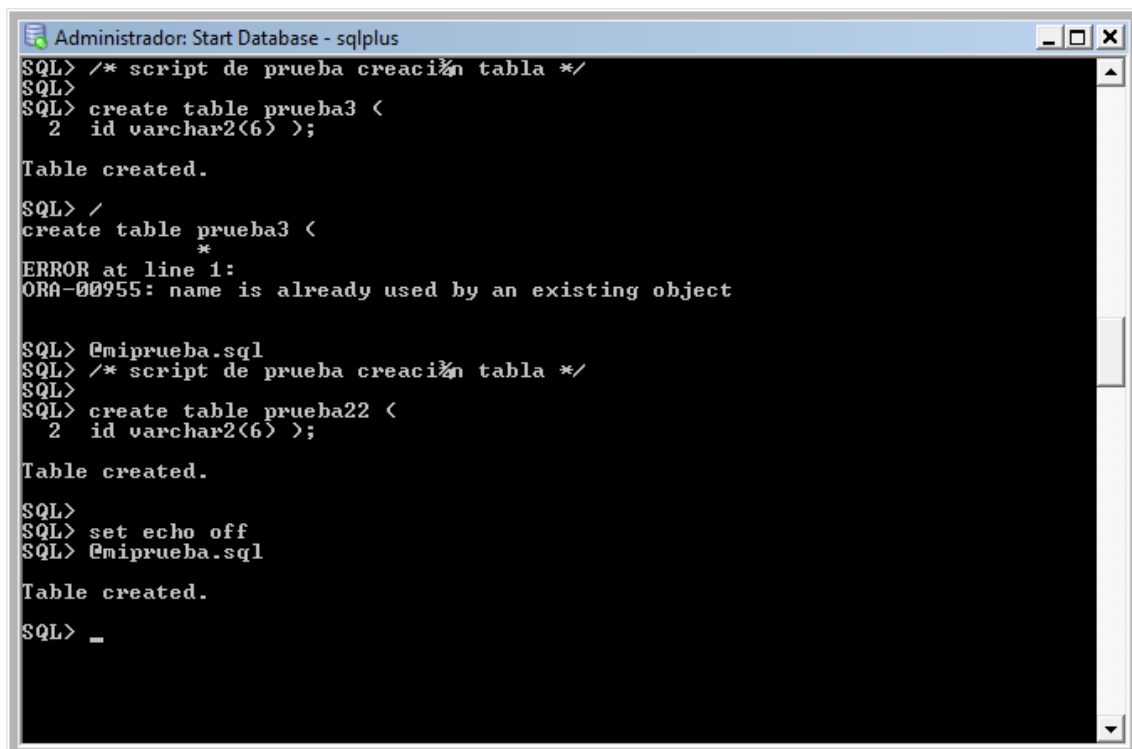
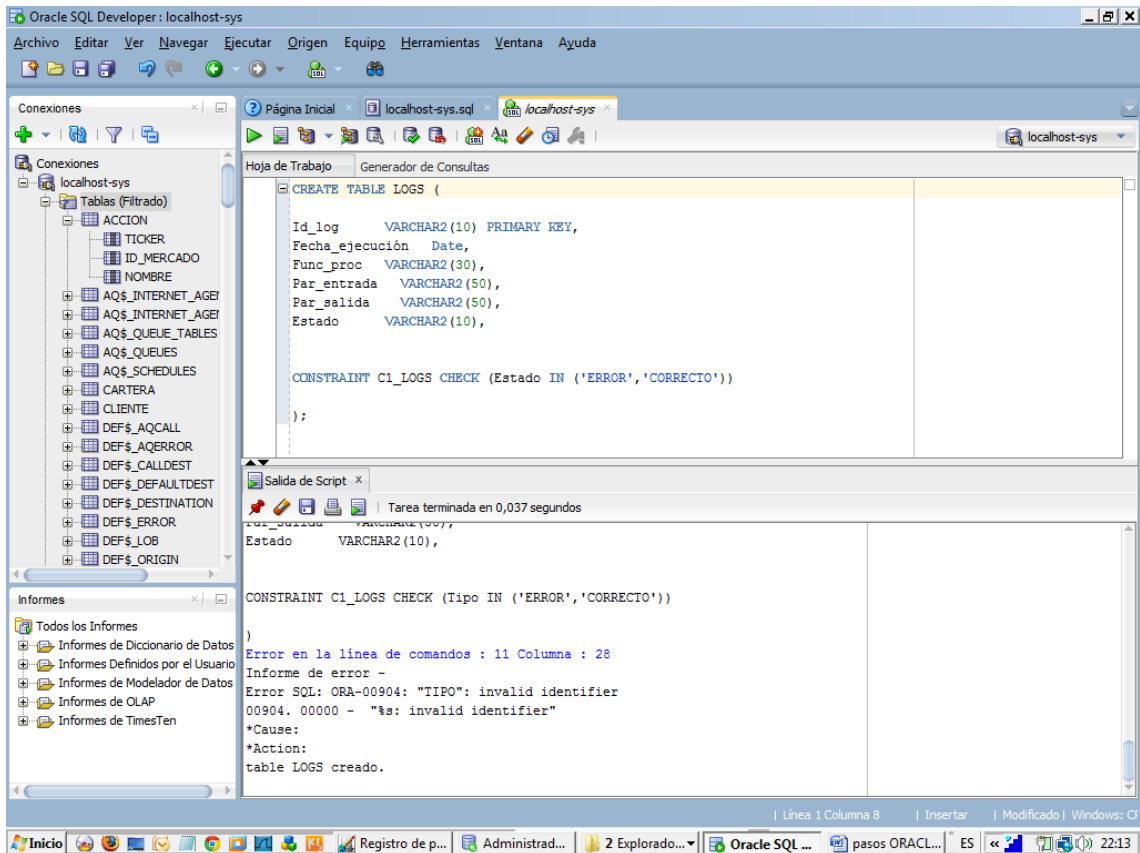
11.1. Creación tablas de la base de datos:

El script de creación de las tablas se denomina "CREAR TABLAS.sql". Contiene el nombre y las columnas de las tablas de la base de datos con las restricciones correspondientes. Cada tabla tiene la clave principal ("*primary key*") y cuando corresponda, se han creado también, las claves ajenas ("*foreign key*"). Por ejemplo: CONSTRAINT PK_ORDEN_COMPRA_VENTA PRIMARY KEY (*Id_orden*), CONSTRAINT FK1_ORDEN_COMPRA_VENTA FOREIGN KEY (*NIF*) REFERENCES CLIENTE (*NIF*).

Algunas tablas tienen la restricción "*check*" para obligar que el dato que se almacene en ese campo contenga uno en concreto y no existan valores inadecuados. Por ejemplo: CONSTRAINT C1_ORDEN_COMPRA_VENTA CHECK (*Tipo_orden* IN ('VENTA','COMPRA')).

Se ha creado además el script "BORRAR TABLAS.sql" que permite eliminar las tablas de la base de datos de este proyecto por si fuera necesario.

A continuación se visualiza la pantalla "Generador de consultas" de *Oracle Developer* donde aparecen las tablas y la pantalla de *SQLPLUS* que permite lanzar el *script* de creación de las tablas de una manera automática.



11.2. Inserción de Datos:

Para insertar datos en las tablas se ha creado un script para cada tabla, cuyo nombre está formado por el prefijo IN seguido del nombre de la tabla:

“IN_CLIENTE.sql”, “IN_CARTERA.sql”, “IN_ACCION.sql”, “IN_CARTERA_ACCION.sql”,
“IN_CAMBIO_DIVISA.sql”, “IN_ORDEN_COMPRA_VENTA.sql”, “IN_TRADING.sql”,
“IN_TRADING_CLIENTE.sql”, “IN_PRECIO_ACCION.sql”, “IN_MERCADO.sql”

11.3. Procedimientos de Alta, Baja y Modificación:

Los paquetes proporcionan un mecanismo para extender el lenguaje. Es una estructura PL/SQL que permite almacenar juntos una serie de objetos relacionados.

En general cada tabla debe tener los procedimientos adecuados para las altas, bajas y modificaciones de datos. Los paquetes implementados según los requisitos son: “PAQ_CLIENTE.sql”, “PAQ_CARTERA_ACCIONES.sql”, “PAQ_ORDEN_CV.sql”

- “PAQ_CLIENTE.sql”: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CLIENTE.
 - IN_CLIENTE: Procedimiento para dar de alta a CLIENTES, hay que insertar en la tabla CLIENTE e insertar los valores de los parámetros

PARÁMETROS

INPUT: par_NIF IN CLIENTE.NIF%TYPE,
par_Nombre IN CLIENTE.Nombre%TYPE,
par_Apellido1 IN CLIENTE.Apellido1%TYPE,
par_Apellido2 IN CLIENTE.Apellido2%TYPE,
par_Ciudad IN CLIENTE.Ciudad%TYPE,
par_Dirección IN CLIENTE.Dirección%TYPE,
par_Teléfono IN CLIENTE.Teléfono%TYPE,
par_Email IN CLIENTE.Email%TYPE,
par_Fecha_alta IN CLIENTE.Fecha_alta%TYPE

OUTPUT: RSP

POSTCONDICION: Se insertan los datos del cliente.

EXCEPTION: 'ERROR: El NIF del cliente no puede ser nulo'
'ERROR: El Apellido1 del cliente no puede ser nulo'
'ERROR: El Apellido2 del cliente no puede ser nulo'
'ERROR: La fecha de alta no puede ser nula'
'ERROR: (SQLERRM)'

- DEL_CLIENTE: Realiza la baja en la tabla CLIENTE e inserta el valor en fecha_baja.

PARAMETROS:

INPUT: par_NIF IN CLIENTE.NIF%TYPE,
par_Fecha_baja IN CLIENTE.Fecha_baja%TYPE

OUTPUT: RSP

PRECONDICION: Debe existir el cliente

POSTCONDICION: Se da de baja el cliente

EXCEPTION: 'ERROR: El NIF del cliente no puede ser nulo

'ERROR: El cliente no existe

'ERROR: El cliente tiene una cartera de acciones activa.

'ERROR: (SQLERRM)'

- UP_CLIENTE: Realiza las modificaciones en los datos de la tabla CLIENTE.

PARAMETROS:

INPUT: par_NIF IN CLIENTE.NIF%TYPE,
par_Nombre IN CLIENTE.Nombre%TYPE,
par_Apellido1 IN CLIENTE.Apellido1%TYPE,
par_Apellido2 IN CLIENTE.Apellido2%TYPE,
par_Ciudad IN CLIENTE.Ciudad%TYPE,
par_Dirección IN CLIENTE.Dirección%TYPE,
par_Teléfono IN CLIENTE.Teléfono%TYPE,
par_Email IN CLIENTE.Email%TYPE,
par_Fecha_alta IN CLIENTE.Fecha_alta%TYPE,
par_Fecha_baja IN CLIENTE.Fecha_baja%TYPE

OUTPUT: RSP

PRECONDICION: Debe existir el cliente

POSTCONDICION: Se modifican los datos del cliente.

EXCEPTION: 'ERROR: El NIF del cliente no puede ser nulo

'ERROR: El cliente no existe

'ERROR: El cliente tiene una cartera de acciones activa.

'ERROR: (SQLERRM)'

- "PAQ_CARTERA.sql": Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CARTERA.

- IN_CARTERA: Inserta en la tabla CARTERA los datos correspondientes

PARÁMETROS:

INPUT: NIF, Fecha_alta

OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se da de alta la cartera de inversión del cliente
'OK' si la operación se ha realizado con éxito

EXCEPTION: 'ERROR: 'El NIF no puede ser nulo'
'ERROR: 'La fecha de alta no puede ser nula'
'ERROR: (SQLERRM)'

- DEL_CARTERA: Realiza la baja de la cartera del cliente. Se modifica la fecha de baja.

PARÁMETROS:

INPUT: par_Id_cartera, fecha_baja

OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se da de baja la cartera de inversión del cliente

RETORNA:

'OK' si la operación se ha realizado con éxito

'ERROR: 'El NIF no puede ser nulo'

'ERROR: 'La fecha de baja no puede ser nula'

'ERROR: (SQLERRM)'

- UP_CARTERA: Realiza las modificaciones en los datos de la tabla CARTERA.

PARAMETROS:

INPUT: par_Id_cartera IN CARTERA.Id_cartera%TYPE,
par_Fecha_Alta IN CARTERA.Fecha_Alta%TYPE,
par_Fecha_baja IN CARTERA.Fecha_baja%TYPE

OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se modifican los datos de la cartera del cliente.

EXCEPTION: 'ERROR: El id_cartera del cliente no puede ser nulo'
'ERROR: (SQLERRM)'

- “PAQ_CARTERA_ACCION.sql”: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla CARTERA_ACCIONES.

- IN_CARTERA_ACCIONES: Inserta en la tabla CARTERA_ACCIONES los datos correspondientes.

PARÁMETROS:

INPUT: par_Id_cartera, par_ticker, par_cantidad, par_fecha_inicio

OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se da de alta las acciones en la cartera de inversión del cliente.

EXCEPTION:

'ERROR: 'El Id_cartera no puede ser nulo'
'ERROR: 'El ticker no puede ser nulo'
'ERROR: 'La cantidad no puede ser nula'
'ERROR: 'La fecha_inicio no puede ser nula'
'ERROR: (SQLERRM);

- DEL_CARTERA_ACCION: Procedimiento para dar baja carteras de clientes, hay que modificar la tabla e insertar fecha_baja.

PARÁMETROS:

INPUT: par_Id_cartera,
OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se da de baja el ticker de la cartera del cliente.

RETORNA:

- 'OK' si la operación se ha realizado con éxito
- 'ERROR' Si:

'ERROR: 'El identificador de cartera no puede ser nulo';
'ERROR: 'EL ticker no puede ser nulo';
'ERROR: (SQLERRM)'

- UP_CARTERA_ACCION: Realiza las modificaciones en los datos de la tabla CARTERA_ACCIONES.

PARAMETROS:

INPUT: par_Id_cartera IN CARTERA_ACCION.Id_cartera%TYPE,
par_ticker IN CARTERA_ACCION.ticker%TYPE,
par_cantidad IN CARTERA_ACCION.cantidad%TYPE,
par_Fecha_inicio IN CARTERA_ACCION.Fecha_inicio%TYPE,
par_Fecha_fin IN CARTERA_ACCION.Fecha_fin%TYPE
OUTPUT: RSP

PRECONDICION: Los campos obligatorios tengan valores correctos.

POSTCONDICION: Se modifican los datos de las acciones de la cartera.

EXCEPTION: 'ERROR: El id_cartera del cliente no puede ser nulo'
'ERROR: (SQLERRM)'

- “PAQ_ ORDEN_COMPRA_VENTA.sql”: Contiene las funciones y procedimientos para actualizar los datos correspondientes a la tabla ORDEN_COMPRA_VENTA.

- IN_ ORDEN_COMPRA_VENTA: Procedimiento para dar de alta órdenes de compra/venta.

PARAMETROS:

INPUT: par_NIF IN ORDEN_COMPRA_VENTA.NIF%TYPE,
par_Ticker IN ORDEN_COMPRA_VENTA.Ticker%TYPE,
par_id_trading IN ORDEN_COMPRA_VENTA.id_trading%TYPE,
par_Tipo_orden IN ORDEN_COMPRA_VENTA.Tipo_orden%TYPE,
par_Tipo_origen IN ORDEN_COMPRA_VENTA.Tipo_origen%TYPE,
par_Fecha_entrada_broker IN
ORDEN_COMPRA_VENTA.Fecha_entrada_broker%TYPE,
par_Estado IN ORDEN_COMPRA_VENTA.Estado%TYPE,
Par_Fecha_ejec_mercado
ORDEN_COMPRA_VENTA.Fecha_ejec_mercado%TYPE,
par_Cantidad IN ORDEN_COMPRA_VENTA.Cantidad%TYPE,
par_Precio_minimo IN ORDEN_COMPRA_VENTA.Precio_mínimo%TYPE,
par_Precio_ejecucion IN ORDEN_COMPRA_VENTA.Precio_ejecución%TYPE);

OUTPUT: RSP

PRECONDICION: El valor de los parámetros es correcto.

POSTCONDICION: Se inserta la orden de compra/venta

EXCEPTION: 'ERROR: 'El NIF no puede ser nulo';
'ERROR: 'El ticker no puede ser nulo';
'ERROR: 'El tipo de orden no puede ser nulo';
'ERROR: 'El tipo de origen no puede ser nulo';
'ERROR: 'El estado no puede ser nulo';
'ERROR: 'La cantidad no puede ser nula';
'ERROR: (SQLERRM)'

- DEL_ ORDEN_COMPRA_VENTA:
Descripción: Realiza la baja en la tabla ORDEN_COMPRA_VENTA los datos correspondientes. Realmente es poner el campo estado en “cancelada”.

PARAMETROS:

INPUT: par_Id_orden IN ORDEN_COMPRA_VENTA.Id_orden%TYPE);

OUTPUT: RSP

PRECONDICION: El valor de los parámetros es correcto.

POSTCONDICION: Se modifica el estado de la orden de compra/venta a cancelada.

EXCEPTION: 'ERROR: El id_orden no puede ser nulo'
'ERROR: (SQLERRM)'

- UP_ORDEN_COMPRA_VENTA: Realiza las modificaciones en los datos de la tabla ORDEN_COMPRA_VENTA.

PARAMETROS:

INPUT: par_Id_orden IN ORDEN_COMPRA_VENTA.Id_orden%TYPE

OUTPUT: RSP

PRECONDICION: El valor de los parámetros es correcto.

POSTCONDICION: Se modifica el estado de la orden de compra/venta a cancelada.

EXCEPTION: 'ERROR: El id_orden no puede ser nulo';
'ERROR: (SQLERRM)'

- “PR_VALOR_ACCION.sql”: Procedimiento que actualiza el valor de cierre de la acción.

- **Descripción:** Procedimiento que actualiza el valor de cierre de la acción.
- **Parámetros entrada:** Ticker, precio en euros, fecha
- **Parámetros salida:** se actualiza la tabla PRECIO_ACCION y se muestra por pantalla DBMS_OUTPUT.PUT_LINE('Se actualiza el ticker: ' || par_ticker);
- **Tablas:** PRECIO_ACCION, ACCION, CAMBIO_DIVISA, LOGS
- **Control:** A través de la tabla LOGS con INSERT:

```
secuencia_logs.nextval, SYSDATE, 'PROC_CLIOPER', 'PAR_IN', 'RSP: ' || RSP,'OK',
v_usuario
```

```
secuencia_logs.nextval, SYSDATE,' PROC_CLIOPER','PAR_IN', RSP,'ERROR'
```

Comprobar que el ticker y el valor dólar existen

- **Exception:** WHEN NO_DATA_FOUND, WHEN OTHERS, WHEN ERROR
- **Instrucciones de ejecución:** EXECUTE PR_CIERRE_DIARIO ('SAB',2.83,'12/06/2015');

11.4. Procedimientos de Consulta:

Los procedimientos implementados para obtener la información necesaria según los requisitos son: "PROC_CARTERA.sql", "PRC_CLIENTE.sql", "PROC_ACCIONES_OPER.sql", "PROC_ACCIONES_AUT.sql"

"PROC_CARTERA.sql"

- **Descripción:** Listado de la composición de la cartera de acciones vigente del cliente y su valor en euros de las acciones del momento.
- **Parámetros entrada:** Fecha, identificador del cliente (NIF).
- **Parámetros salida:**), identificador de cartera (id_cartera), acción (ticker), valor euros (precio acción * cantidad), valor dólares (precio acción * cantidad)
- **Tablas:** CARTERA_ACCION, ACCION, PRECIO_ACCION, LOGS
- **Control:** A través de la tabla LOGS con INSERT:

```
secuencia_logs.nextval, SYSDATE, 'PROC_CLIOPER', 'PAR_IN', 'RSP: '||RSP,'OK',  
v_usuario
```

```
secuencia_logs.nextval, SYSDATE, 'PROC_CLIOPER', 'PAR_IN', RSP, 'ERROR'
```

- **Exception:** WHEN NO_DATA_FOUND, WHEN OTHERS
- **Instrucciones de ejecución:** EXECUTE PROC_CARTERA ('01/04/2015','99999999B');

"PROC_CLIOPER.sql"

- **Descripción:** Listado de todos los clientes que han realizado operaciones durante este año.
- **Parámetros entrada:** Año.
- **Parámetros salida:** identificador del cliente (NIF), nombre del cliente (Nombre, apellido1, apellido2), número de órdenes de compra cargadas durante ese año, número de órdenes de venta cargadas durante ese año, número de órdenes de compra ejecutadas durante ese año, sumatorio en euros de todas las órdenes de compra ejecutadas durante ese año, valor medio en euros de cada operación de compra ejecutada por el cliente durante ese año, número de órdenes de venta ejecutadas durante ese año, sumatorio en euros de todas las órdenes de venta ejecutadas durante ese año, valor medio en euros de cada operación de venta ejecutada por el cliente durante ese año, número de órdenes de compra canceladas durante ese año, número de órdenes de venta canceladas durante ese año.
- **Tablas:** CLIENTE, ORDEN_COMPRA_VENTA, ACCION, LOGS
- **Control:** A través de la tabla LOGS con INSERT:

```

    secuencia_logs.nextval, SYSDATE, 'PROC_CLIOPER', 'PAR_IN', 'RSP: '||RSP,'OK',
v_usuario

```

```

    secuencia_logs.nextval, SYSDATE, 'PROC_CLIOPER', 'PAR_IN', RSP, 'ERROR'

```

- **Exception:** WHEN NO_DATA_FOUND, WHEN OTHERS
- **Instrucciones de ejecución:** EXECUTE PROC_CLIOPER ('2015');

“PROC_ACCIONES_OPER.sql”

- **Descripción:** Listado de todas las acciones sobre las cuales se han realizado alguna operación.
- **Parámetros entrada:** Año
- **Parámetros salida:** Identificador de la acción (ticker), nombre completo de la acción (nombre), número de las operaciones de compra ejecutadas durante ese año, suma total en euros de las órdenes de compra ejecutadas durante el año sobre la acción, valor medio en euros de las órdenes de compra ejecutadas durante ese año sobre la acción, número de las operaciones de venta ejecutadas durante ese año sobre la acción, suma total en euros de las órdenes de venta ejecutadas sobre la acción, revalorización de la acción en porcentaje desde el uno de enero del año indicado. Se muestran por pantalla DBMS_OUTPUT.PUT_LINE(fila.TICKER||' '||fila.NOMBRE||' '||fila.CONTAR||' '|| fila.SUMA||' '|| fila.MEDIA ||' '|| fila.ORDEN ||' '|| v_reval ||' '|| '%');
- **Tablas relacionadas:** ACCIÓN, ORDEN_COMPRA_VENTA, PRECIO_ACCION, LOGS
- **Control:** A través de la tabla LOGS con INSERT:

```

    secuencia_logs.nextval, SYSDATE, ' PROC_ACCIONES_OPER ', 'PAR_IN', 'RSP:
' ||RSP,'OK', v_usuario

```

```

    secuencia_logs.nextval, SYSDATE, 'PROC_ACCIONES_OPER ', 'PAR_IN', RSP, 'ERROR'

```

- **Exception:** WHEN NO_DATA_FOUND, WHEN OTHERS
- **Instrucciones de ejecución:** EXECUTE PROC_ACCIONES_OPER ('2015');

“PROC_ACCIONES_AUT.sql”

- **Descripción:** Listado de las diez acciones más adquiridas por los sistemas automáticas, ordenados de forma decreciente por su volumen.
- **Parámetros entrada:** No es necesario.
- **Parámetros salida:** nombre completo de la acción y la cantidad, que se leen a través de un cursor que se muestra por pantalla: (DBMS_OUTPUT.PUT_LINE(fila.nombre ||' '||fila.cantidad)
- **Tablas relacionadas:** ACCIÓN, ORDEN_COMPRA_VENTA, LOGS

- **Control:** A través de la tabla LOGS con INSERT:

```
secuencia_logs.nextval, SYSDATE, 'TR_MAYOR', 'PAR_IN', 'RSP: '||RSP,'OK', v_usuario
```

```
secuencia_logs.nextval, SYSDATE,'TR_MAYOR','PAR_IN', RSP,'ERROR'
```

- **Exception:** WHEN NO_DATA_FOUND, WHEN OTHERS
- **Instrucciones de ejecución:** EXECUTE PROC_ACCIONES_AUT ();

11.5. Módulo Estadístico:

Permite dar respuesta inmediata a las consultas definidas en los requisitos. Como hay que utilizar las tablas correspondientes a través de una SELECT, hay que actualizar las tablas de este módulo a través de disparadores o *triggers*.

Disparadores:

Un disparador se asemeja a los procedimientos y funciones pero se ejecuta de manera implícita cada vez que tiene lugar el suceso del disparo. El acto de ejecutar un disparador se conoce como disparo. El suceso del disparo es una operación DML (INSERT, UPDATE o DELETE) sobre una tabla de la base de datos.

Las tablas que forman parte del módulo estadístico se actualizan a través de disparadores o *triggers*:

“TR_BENEFICIO.sql”

- ❖ Para actualizar la tabla EST_BENEFICIO se crea un disparador denominado TR_BENEFICIO que actualiza los campos de esta tabla: NIF, Beneficio_neto, porcentaje, mes. Esta tabla contiene el beneficio neto en euros y el porcentaje de su cartera de acciones durante un mes de un cliente y un mes concreto. El campo beneficio de esta tabla contiene la diferencia entre el precio de compra y el precio actual de las acciones de la cartera del cliente. El precio de la acción está en la tabla PRECIO_ACCIÓN por fecha. El campo porcentaje de la tabla EST_BENEFICIO se calcula en función del beneficio neto obtenido.

NOMBRE TRIGGER	TR_BENEFICIO
DESCRIPCIÓN	Actualiza tabla EST_BENEFICIO
TABLA IMPLICADAS	PRECIO_ACCION, EST_BENEFICIO, CARTERA, CLIENTES, CARTERA_ACCIONES, LOGS
PARÁMETROS ENTRADA	Se dispara cuando se detecta que es el día uno del mes y se ha insertado un nuevo valor en la tabla PRECIO_ACCION
PARÁMETROS SALIDA	RSP:= 'ERROR ' TO_CHAR(err_num) ': ' err_msg;
CONTROL	A través de la tabla LOGS: secuencia_logs.nextval, SYSDATE, 'TR_MAYOR', 'PAR_IN', 'RSP:

	' RSP,'OK', v_usuario secuencia_logs.nextval,SYSDATE,'TR_BENEFICIO','PAR_IN',RSP,'ERROR' ' v_usuario
EXCEPTION	WHEN NO_DATA_FOUND WHEN OTHERS

“TR_PENDIENTES.sql”

- ❖ Para actualizar la tabla EST_PENDIENTES se crea un disparador denominado TR_PENDIENTES que actualiza los campos de esta tabla: Ticker, Acciones_dispo_broker, Num_clientes. Esta tabla contiene el número de acciones disponibles dentro del broker, es decir donde el valor del campo estado de la tabla ORDEN_COMPRA_VENTA sea igual a “pendiente” y el campo Tipo_orden de la tabla ORDEN_COMPRA_VENTA sea igual a “venta”. Para saber el número de clientes que contienen esa acción se debe consultar en las tablas CARTERA_ACCIONES y ACCIONES.

NOMBRE TRIGGER	TR_PENDIENTES
DESCRIPCIÓN	Actualiza tabla EST_PENDIENTES
TABLA IMPLICADAS	ORDEN_COMPRA_VENTA, CARTERA, CARTERA_ACCIONES, EST_PENDIENTES, LOGS
PARÁMETROS ENTRADA	Ticker de la acción
PARÁMETROS SALIDA	RSP:= 'ERROR ' TO_CHAR(err_num) ': ' err_msg;
CONTROL	A través de la tabla LOGS: secuencia_logs.nextval, SYSDATE, 'TR_PENDIENTES', 'PAR_IN', 'RSP: ' RSP,'OK', v_usuario secuencia_logs.nextval,SYSDATE,'TR_PENDIENTES','PAR_IN',RSP,'ERR OR',v_usuario
EXCEPTION	WHEN NO_DATA_FOUND WHEN OTHERS

El procedimiento PR_PENDIENTES que se ha creado para ofrecer al usuario la visualización de los datos de la acción pendiente que soliciten.

De esta manera, el usuario ejecuta la llamada al procedimiento y le pasa como parámetro el ticker de la acción. Se realiza una SELECT a la tabla EST_PENDIENTES y se visualizan los datos por pantalla.

```
EXECUTE PR_PENDIENTES ('SAB');
```

NOMBRE TRIGGER	PR_PENDIENTES
DESCRIPCIÓN	Visualiza los datos de la tabla EST_PENDIENTES según el parámetro de entrada Ticker
TABLA IMPLICADAS	EST_PENDIENTES, LOGS
PARÁMETROS ENTRADA	Ticker de la acción
PARÁMETROS SALIDA	RSP:= 'ERROR ' TO_CHAR(err_num) ': ' err_msg;
CONTROL	A través de la tabla LOGS: secuencia_logs.nextval, SYSDATE, 'TR_PENDIENTES', 'PAR_IN', 'RSP: ' RSP,'OK', v_usuario secuencia_logs.nextval,SYSDATE,'TR_PENDIENTES','PAR_IN',RSP,'ERR OR',v_usuario
EXCEPTION	WHEN NO_DATA_FOUND WHEN OTHERS

“TR_MAYOR.sql”

- ❖ Para actualizar la tabla EST_CARTERA_MAYOR se crea un disparador denominado TR_MAYOR que pueda actualizar los campos que forman parte de esta tabla: NIF, Fecha, valor y descripción. Esta tabla contiene el cliente con la cartera de acciones más grande. Por lo tanto cuando se ejecute una orden en la tabla PRECIO_ACCION se tienen que actualizar los datos de la tabla EST_CARTERA_MAYOR. Esto ocurre cuando al cierre de la bolsa se actualizan los precios de las acciones. Se comparan todas las carteras con su valor y se insertan los datos de la mayor.

NOMBRE TRIGGER	TR_MAYOR
DESCRIPCIÓN	Actualiza tabla EST_CARTERA_MAYOR
TABLA IMPLICADAS	PRECIO_ACCION, EST_CARTERA_MAYOR, LOGS
PARÁMETROS ENTRADA	Se dispara el trigger
PARÁMETROS SALIDA	RSP:= 'ERROR ' TO_CHAR(err_num) ': ' err_msg;
CONTROL	A través de la tabla LOGS: secuencia_logs.nextval, SYSDATE, 'TR_MAYOR', 'PAR_IN', 'RSP: ' RSP,'OK', v_usuario secuencia_logs.nextval,SYSDATE,'TR_MAYOR','PAR_IN',RSP,'ERROR', v_usuario
EXCEPTION	WHEN NO_DATA_FOUND WHEN OTHERS

“TR_NEGOCIADA.sql”

- ❖ Para actualizar la tabla EST_NEGOCIADA se crea un disparador denominado TR_NEGOCIADA que actualiza los campos de esta tabla: Ticker, volumen, fecha, tipo. Esta tabla contiene la acción más negociada comprada y vendida en el broker junto con su volumen. Esto ocurre cuando el campo estado de la tabla ORDEN_COMPRA_VENTA pasa a tener el valor “ejecutada”. De esta manera, se tiene que calcular el número de veces que aparece el ticker de la acción según las órdenes ejecutadas en el broker y según la fecha.

NOMBRE TRIGGER	TR_NEGOCIADA
DESCRIPCIÓN	Actualiza tabla EST_NEGOCIADA
TABLA IMPLICADAS	ORDEN_COMPRA_VENTA, EST_NEGOCIADA, LOGS
PARÁMETROS ENTRADA	Se dispara el trigger
PARÁMETROS SALIDA	RSP:= 'ERROR ' TO_CHAR(err_num) ': ' err_msg;
CONTROL	A través de la tabla LOGS: secuencia_logs.nextval,SYSDATE, 'TR_NEGOCIADA', 'PAR_IN', 'RSP: ' RSP,'OK', v_usuario secuencia_logs.nextval,SYSDATE, 'TR_NEGOCIADA', 'PAR_IN', RSP,'ERROR', v_usuario
EXCEPTION	WHEN NO_DATA_FOUND WHEN OTHERS

12. PRUEBAS:

Para realizar las pruebas se han seguido los siguientes pasos: preparación de datos, pruebas en el compilador Oracle Developer, pruebas a través de consultas y análisis de resultados.

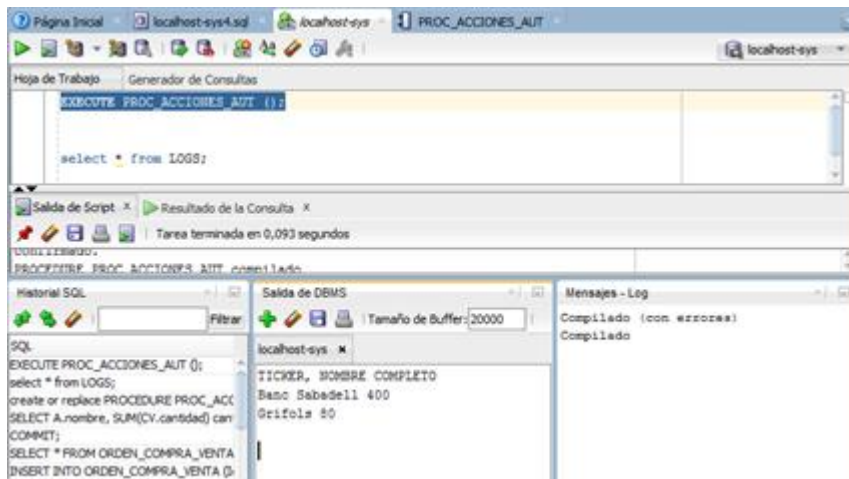
12.1. Preparación de Datos:

Para realizar las pruebas del producto realizado se ha preparado un conjunto de valores para cada tabla a través de la instrucción *insert* en los ficheros “IN_nombre de tabla.sql”. Son los siguientes: *IN_ACCION.sql*, *IN_CAMBIO_DIVISA.sql*, *IN_CARTERA.sql*, *IN_CARTERA_ACCION.sql*, *IN_CLIENTE.sql*, *IN_MERCADO.sql*, *IN_ORDEN_COMPRA_VENTA.sql*, *IN_PRECIO_ACCION.sql*, *IN_TRADING.sql*, *IN_TRADING_CLIENTE.sql*. Estos ficheros también se han utilizado en la parte de desarrollo del producto.

12.2. Pruebas en el compilador de Oracle Developer:

La herramienta *Oracle Developer* permite compilar los procedimientos que se realicen y a través de la ventana *logs* se muestran los errores que se han ido depurando para conseguir un correcto funcionamiento.

A través del documento “ANEXO PRUEBAS.docx”, no incluido en este documento, se pueden consultar más imágenes del compilador y diversas pruebas realizadas.



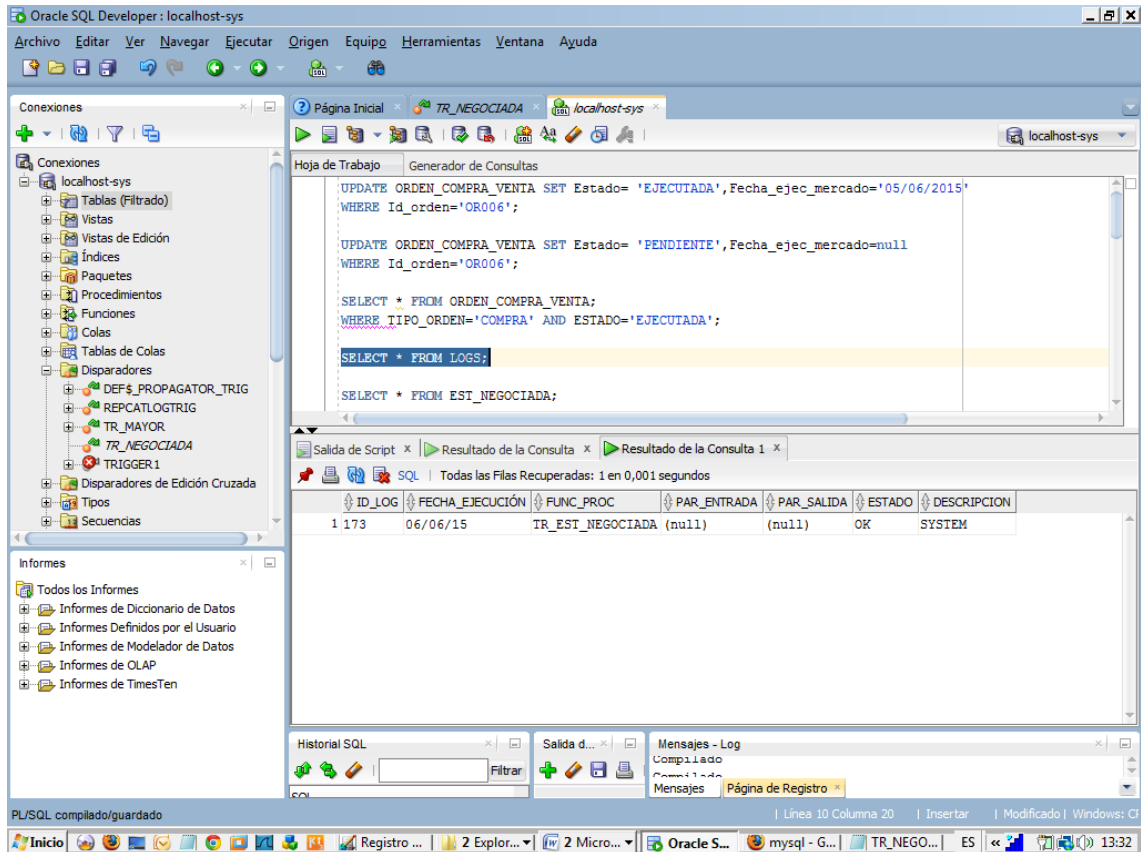
12.3. Pruebas a través de consultas

Se han desarrollado consultas para realizar pruebas en procedimientos y *triggers* implementados. Estas consultas a través de la instrucción *SELECT* muestran los valores que han insertado los procedimientos y *triggers* desarrollados en las tablas de la base de datos. Se han realizado pruebas por separado para comprobar los resultados de los cursores que aparecen en el producto con el objetivo de minimizar errores y comprender los datos obtenidos según los requisitos.

12.4. Análisis de Resultados:

La tabla *logs* almacena los procedimientos y *triggers* que se ejecutan. Existe una columna en la tabla *logs* que indica el estado de la ejecución y que ha permitido, cuando se ha implementado el código, realizar consultas para comprobar el estado de ejecución.

A continuación se visualiza una imagen de *Oracle Developer* con el resultado de una *SELECT* en la tabla *logs*.



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL script with the following content:

```
UPDATE ORDEN_COMPRA_VENTIA SET Estado= 'EJECUTADA', Fecha_ejec_mercado='05/06/2015'
WHERE Id_orden='OR006';

UPDATE ORDEN_COMPRA_VENTIA SET Estado= 'PENDIENTE', Fecha_ejec_mercado=null
WHERE Id_orden='OR006';

SELECT * FROM ORDEN_COMPRA_VENTIA;
WHERE TIPO_ORDEN='COMPRA' AND ESTADO='EJECUTADA';

SELECT * FROM LOGS;

SELECT * FROM EST_NEGOCIADA;
```

The results pane shows the output of the SQL script, which is a table with the following columns: ID_LOG, FECHA_EJECUCIÓN, FUNC_PROG, PAR_ENTRADA, PAR_SALIDA, ESTADO, and DESCRIPCION. The table contains one row of data:

ID_LOG	FECHA_EJECUCIÓN	FUNC_PROG	PAR_ENTRADA	PAR_SALIDA	ESTADO	DESCRIPCION
1 173	06/06/15	TR_EST_NEGOCIADA	(null)	(null)	OK	SYSTEM

Se observa que la columna *par_salida* contiene el error que se pueda producir al ejecutar procedimientos y triggers del producto. Y la columna *par_entrada* contiene los parámetros con los que se ha realizado la ejecución. Para una mayor descripción de la tabla se puede consultar el apartado de creación de la misma.

13. ANEXO

Informáticos e ingenieros, el talento más cotizado de 2015

- Los profesionales cualificados de la construcción, los ingenieros y los informáticos serán los perfiles más demandados durante este año, según el último estudio de Hays
- **Tendencias generales.** Optimización móvil, una escasez de habilidades, innovación continua y gestión del cambio son algunas de las tendencias que impulsarán el crecimiento del empleo a nivel mundial en 2015, según Hays, multinacional británica especializada en la selección de profesionales cualificados. Además, la demanda se concentrará en competencias de alto nivel, sobre todo en áreas técnicas como la informática, la construcción y la ingeniería. Asimismo, seguirá existiendo un aumento en la demanda de profesionales de marketing digital en todos los ámbitos.
- **Puestos de trabajo y Big Data.** El informe apunta que el Big Data, junto a la tecnología correcta de datos, pueden proporcionar «nuevos conocimientos sin precedentes y modelos predictivos en los empleados (no sólo clientes), que conducen a la mejora de las estrategias de reclutamiento y retención». «En 2015 veremos expertos en recursos humanos interpretando los datos relativos a la atracción de talento, compromiso, retención, desempeño y satisfacción, y lo utilizarán para tomar decisiones estratégicas de atracción de talento», señalan.
- **Empleos en crecimiento.** Asimismo el informe apunta hacia un incremento de la demanda de especialistas altamente cualificados en todos los sectores. Además, las previsiones apuntan hacia un nuevo «boom de la construcción» que conllevará una creación de empleos para profesionales cualificados e ingenieros», además de especialistas en inversiones y valoraciones. Finalmente, la búsqueda de nuevos empleos a través del móvil seguirá en aumento, en consonancia con la tendencia arrancada en los últimos años.

<http://www.elmundo.es/economia/2015/02/04/54d11aa6e2704e79298b4570.html>

14. BIBLIOGRAFÍA

- ✚ Materiales didácticos de las asignaturas de Base de datos I, Bases de datos II y Estructura de la Información.
- ✚ Riesgo e incertidumbre en la gestión de proyectos informáticos, Juan Izquierdo Jiménez, Director Técnico de SchlumbergerSema
- ✚ Perfiles profesionales en informática: http://www.eiieo.org/wp-content/uploads/perfiles-informatica_noviembre05.pdf
- ✚ Sistemas automáticos de trading:
<http://www.iit.upcomillas.es/pfc/resumenes/42b9b9b81c0a1.pdf>
- ✚ Domine Oracle 11g. Autor: David Roldán Martínez, Pedro J. Valderas Aranda. Editorial RAMA
- ✚ Fundamentos de base de datos. Autor: Silberschatz, Korth, Sudarshan. Editorial Mc Graw Hill
- ✚ Oracle8 Programación PL/SQL, Scott Urman. Editorial Oracle Press Oficial Osborne.