



DOBLE LICENCIATURA EN INGENIERÍA DE  
TELECOMUNICACIÓN Y ADMINISTRACIÓN Y  
DIRECCIÓN DE EMPRESAS

Curso Académico 2016/2017

Proyecto Fin de Carrera

MANTENIMIENTO Y SOPORTE DE UN  
DATAMART DE RIESGOS

Autor : Fernando Caballero Sánchez

Tutor : Dr. Gregorio Robles



# Proyecto Fin de Carrera

Mantenimiento y soporte de un Datamart de riesgos

**Autor :** Fernando Caballero Sánchez

**Tutor :** Dr. Gregorio Robles Martínez

La defensa del presente Proyecto Fin de Carrera se realizó el día            de  
de 2017, siendo calificada por el siguiente tribunal:

**Presidente:**

**Secretario:**

**Vocal:**

y habiendo obtenido la siguiente calificación:

**Calificación:**

Fuenlabrada, a            de            de 2017



*Dedicado a  
mi familia y a mi esposa*



# Agradecimientos





# Resumen

Ha de constar de tres o cuatro párrafos, donde se presente de manera clara y concisa de qué va el proyecto. Han de quedar respondidas las siguientes preguntas:

- ¿De qué va este proyecto? ¿Cuál es su objetivo principal?
- ¿Cómo se ha realizado? ¿Qué tecnologías están involucradas?
- ¿En qué contexto se ha realizado el proyecto? ¿Es un proyecto dentro de un marco general?

Lo mejor es escribir el resumen al final.



# Summary

Here comes a translation of the “Resumen” into English. Please, double check it for correct grammar and spelling. As it is the translation of the “Resumen”, which is supposed to be written at the end, this as well should be filled out just before submitting.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Motivación . . . . .	1
1.3. Estructura de esta memoria . . . . .	1
<b>2. Objetivos</b>	<b>3</b>
2.1. Objetivo general . . . . .	3
2.2. Objetivos específicos . . . . .	3
2.3. Planificación temporal . . . . .	3
<b>3. Estado del arte</b>	<b>5</b>
3.1. Arquitectura y diseño de un Datamart . . . . .	5
3.2. Cliente VPN de Cisco . . . . .	7
3.3. Oracle . . . . .	8
3.4. PuTTY . . . . .	9
3.5. Toad para Oracle . . . . .	10
3.6. FileZilla . . . . .	11
3.7. ODBC . . . . .	11
3.8. Python . . . . .	12
3.9. Shell Scripts . . . . .	13
<b>4. Diseño e implementación</b>	<b>15</b>
4.1. Diseño general . . . . .	15
4.2. Monitorización de la ejecución . . . . .	23
4.3. Ejecución programada de los procesos . . . . .	25

4.4. Ejecución bajo demanda de los procesos . . . . .	26
4.5. Fallos comunes en la ejecución diaria y tratamiento . . . . .	27
4.6. Rendimiento y optimización del Datamart . . . . .	30
<b>5. Resultados</b>	<b>31</b>
<b>6. Conclusiones</b>	<b>33</b>
6.1. Consecución de objetivos . . . . .	33
6.2. Aplicación de lo aprendido . . . . .	33
6.3. Lecciones aprendidas . . . . .	33
6.4. Trabajos futuros . . . . .	34
<b>A. Manual de usuario</b>	<b>35</b>
<b>Bibliografía</b>	<b>37</b>

# Índice de figuras

3.1. Modelo Datamart hub-and-spokes . . . . .	6
3.2. Pantalla de configuración del Cliente VPN de Cisco . . . . .	8
3.3. Pantalla de configuracion de PuTTY . . . . .	10
3.4. Pantalla de configuración de Toad para Oracle . . . . .	11
3.5. Pantalla de configuracion del cliente de Filezilla . . . . .	12
4.1. Entidades Clarify . . . . .	16
4.2. Entidades Spirit . . . . .	18
4.3. Entidades Arbor . . . . .	18
4.4. Entidades CRM . . . . .	19
4.5. Entidades HUR Pandora . . . . .	19
4.6. Entidades BNK . . . . .	19
4.7. Entidades SAP (1) . . . . .	20
4.8. Entidades SAP (2) . . . . .	21
4.9. Entidades Informa . . . . .	21
4.10. Entidades SEPA . . . . .	21
4.11. Entidades Otras . . . . .	22
4.12. Entidades AGEX . . . . .	22
4.13. Entidades BO . . . . .	23
4.14. Entidades Adquisiciones . . . . .	23
4.15. Mapa de sistemas Datamart de riesgos . . . . .	24
4.16. Salida por línea de comandos del script COMPROBAR_EJECUCIONES . . . .	25
4.17. Modificación fichero ctl_lanzador.cfg para reprocesado de los ficheros del pro- ceso CRM . . . . .	29





# Capítulo 1

## Introducción

En este capítulo se introduce el proyecto, dando la información sobre el contexto en el que se ha desarrollado así como información general acerca del mismo.

### 1.1. Contexto

En la actualidad, el área de Inteligencia de Negocio o “Business Intelligence” ha adquirido una gran relevancia para las empresas. Esto es así debido a la elevada capacidad tanto de almacenamiento como de procesamiento de datos, los cuales se descartaban por resultar demasiado costoso su almacenamiento y tratamiento. Empleando un buen tratamiento de toda esta información las organizaciones son capaces de tomar decisiones de negocio de la forma más acertada posible: descifrando tendencias, segmentando mercados y teniendo un mayor conocimiento de la actuación de la organización en un determinado mercado.

ALGÚN GRÁFICO DE LA IMPORTANCIA DEL BI PARA LAS EMPRESAS

### 1.2. Motivación

### 1.3. Estructura de esta memoria



# **Capítulo 2**

## **Objetivos**

**2.1. Objetivo general**

**2.2. Objetivos específicos**

**2.3. Planificación temporal**



# Capítulo 3

## Estado del arte

### 3.1. Arquitectura y diseño de un Datamart

El principal motivo por el que una organización necesita disponer de un Data Warehouse surge debido a que los datos clave de toda la organización se encuentran fragmentados a lo largo de múltiples y dispares aplicativos, ejecutándose en diferentes plataformas y en distintas localizaciones físicas. Esta situación no es favorable para una buena toma de decisiones. Además, cuando existe redundancia de datos en múltiples bases de datos, la calidad de los mismos suele tender a deteriorarse.

Por otro lado, las bases de datos operacionales están diseñadas para operar en los sistemas y recibir los datos de los mismos, por lo que no son adecuadas para llevar a cabo consultas y analíticas on-line.

Los entornos de Business Intelligence se pueden basar en diferentes arquitecturas, dependiendo en las necesidades específicas del negocio. El modelo que se muestra en la siguiente figura es el “hub-and-spokes”, siendo el más popular y el que se usa en muchas organizaciones.

El proceso que siguen los datos en este modelo es el siguiente: los datos se mueven desde las bases de datos de los sistemas operacionales hacia la zona de “*staging*” del Data Warehouse, donde se preparan y modifican los datos para ser almacenados en el Data Warehouse. Finalmente, esos datos preparados se envían hacia los diferentes Data Marts que la organización haya definido.

La tecnología o proceso que se emplea para mover los datos en esta arquitectura se denomina ETL (por sus siglas en inglés “Extract, Transform and Load”). Se emplea para transferir datos

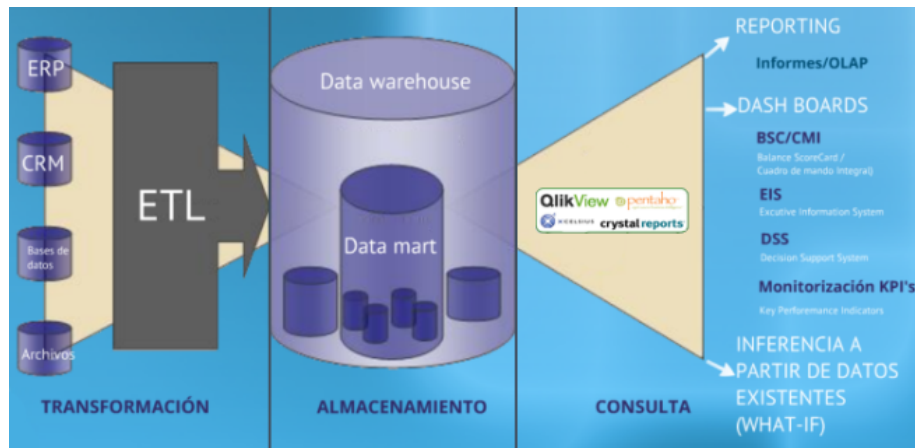


Figura 3.1: Modelo Datamart hub-and-spokes

desde las aplicaciones operacionales a la zona de “*staging*” del Data Warehouse, desde esta al Data Warehouse, y desde este último a cada uno de los Data Marts.

Mediante el proceso de ETL los datos se extraen, se transforman los valores de los datos inconsistentes, se limpian los datos malformados, se filtran los datos y se cargan en la base de datos objetivo. La programación temporal de los trabajos de ETL es crítica, así como la tolerancia a errores: en caso de fallo de uno de estos trabajos, el resto debería responder de manera apropiada.

El área de “*staging*” del Data Warehouse es donde se copian los datos de los sistemas de origen de manera temporal. Este paso es fundamental en muchas arquitecturas de Data Warehouse para coordinar los tiempos entre todos los sistemas. De manera resumida, todos los datos que se necesitan deben estar disponibles al mismo tiempo antes de ser integrados en el Data Warehouse y, debido a, entre otras cosas, los diferentes ciclos del negocio, los ciclos de procesamiento de datos en los sistemas de origen y las limitaciones de recursos hardware y red, no es factible extraer todos los datos de todas las bases de datos operacionales exactamente al mismo tiempo. Por ejemplo, puede ser razonable extraer la información de ventas diaria, sin embargo, extracciones diarias pueden no ser adecuadas para la información financiera que requiere un proceso de reconciliación a mes cerrado. Por último, es importante tener en cuenta que no todas las organizaciones necesitan un área de “*staging*” en su Data Warehouse, ya que para muchos negocios es factible usar ETL para copiar directamente la información de las bases de datos operacionales al Data Warehouse.

El objetivo del Data Warehouse en toda la arquitectura es integrar la información corpora-

tiva. En este sentido, será el Data Warehouse el que contenga la “única versión de la verdad” para la organización, que habrá sido cuidadosamente construida a partir de la información almacenada en diferentes bases de datos operacionales. La cantidad de información que se llega a almacenar en un Data Warehouse es masiva, debido a que los datos se almacenan en un nivel de detalle muy granular. Por ejemplo, cada venta que ha tenido lugar en una organización se almacena y relaciona, permitiendo llevar a cabo agregados, agrupaciones, filtrados y demás operaciones en una cantidad casi ilimitada de formas.

Un aspecto a destacar es que, a diferencia de lo que se suele pensar de manera generalizada, el Data Warehouse no contiene toda la información que existe en una organización. El objetivo de un Data Warehouse es proporcionar métricas clave para el negocio que sean necesarias y útiles para la toma de decisión táctica y estratégica de la organización.

Debido al gran volumen de información que almacena un Data Warehouse, las personas que toman decisiones y necesitan las métricas no acceden a este Data Warehouse directamente. Esto lo hacen a través de varias herramientas “front-end” que leen datos de los Data Marts específicos de cada área de negocio o temática.

Los trabajos de ETL extraen la información del Data Warehouse y pueblan el/los Data Marts de la organización para su uso por parte de los grupos que toman las decisiones dentro de las organizaciones. Los Data Marts pueden ser dimensionales (esquemas en estrella) o relacionales, en función de como se necesite consumir la información y de qué herramienta de “front-end” se emplee para presentar la información.

Cada Data Mart puede contener diferentes combinaciones de tablas, columnas y filas del Data Warehouse de la organización. Por ejemplo, una unidad de negocio o grupo de usuarios que no necesita datos históricos puede requerir únicamente las transacciones del año en curso, o el área de recursos humanos puede necesitar ver todos los detalles de todos los empleados, mientras que información como el salario o dirección de casa puede no ser apropiada para un Data Mart orientado al área de ventas.

## **3.2. Cliente VPN de Cisco**

El cliente VPN de Cisco es una herramienta del proveedor Cisco que permite configurar una red VPN. Actualmente se encuentra obsoleta y sin soporte por parte del fabricante. A través

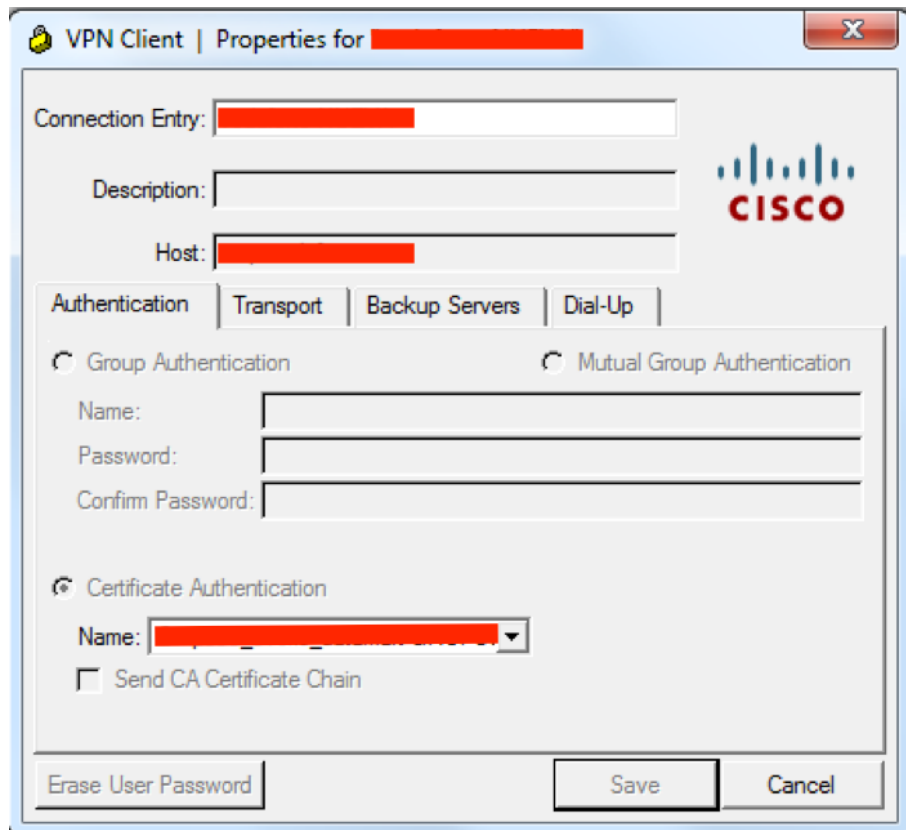


Figura 3.2: Pantalla de configuración del Cliente VPN de Cisco

de Cisco VPN Cliente se crea una conexión segura sobre Internet entre los ordenadores y las máquinas donde residen tanto los datos del Datamart como la ejecución de los diferentes procesos propios del Datamart de Riesgos. Por lo tanto, esta herramienta permite acceder a la red privada de las máquinas del Datamart como si nos encontrásemos físicamente en la red.

Una vez instalada la herramienta es necesario configurarla, agregando la conexión con los parámetros facilitados por el cliente. Para garantizar una conexión segura, el cliente proporciona los certificados necesarios, los cuales habrá que seleccionar también en el cliente VPN.

A continuación, se muestra una captura de pantalla de la herramienta Cisco VPN Client donde se lleva a cabo la configuración de la conexión.

### 3.3. Oracle

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado



por Oracle Corporation. Durante el desarrollo de este proyecto se usa la versión 11.2.0 de Oracle Database.

Cabe señalar que el lenguaje usado para la gestión de la base de datos de Oracle es el Structured Query Language o SQL, así como la extensión procedimental PL/SQL para Oracle:

- SQL is a set-based declarative language that provides an interface to an RDBMS such as Oracle Database. In contrast to procedural languages such as C, which describe how things should be done, SQL is nonprocedural and describes what should be done. Users specify the result that they want (for example, the names of current employees), not how to derive it. SQL is the ANSI standard language for relational databases. All operations on the data in an Oracle database are performed using SQL statements. For example, you use SQL to create tables and query and modify data in tables. A SQL statement can be thought of as a very simple, but powerful, computer program or instruction.
- PL/SQL is a procedural extension to Oracle SQL. PL/SQL is integrated with Oracle Database, enabling you to use all of the Oracle Database SQL statements, functions, and data types. You can use PL/SQL to control the flow of a SQL program, use variables, and write error-handling procedures. A primary benefit of PL/SQL is the ability to store application logic in the database itself. A procedure or function is a schema object that consists of a set of SQL statements and other PL/SQL constructs, grouped together, stored in the database, and run as a unit to solve a specific problem or to perform a set of related tasks. The principal benefit of server-side programming is that built-in functionality can be deployed anywhere.

## 3.4. **PuTTY**

PuTTY es una herramienta de software libre cuya misión es crear un terminal UNIX en entorno Windows. Esta herramienta la desarrolló originalmente Simon Tatham para plataformas Windows, aunque a día de hoy es soportado por voluntarios y se está trabajando en clientes para sistemas operativos Mac OS.

En este Proyecto, PuTTY se usa para abrir en los ordenadores desde donde se realiza el soporte y mantenimiento del Datamart un terminal UNIX conectado a través de SSH a las

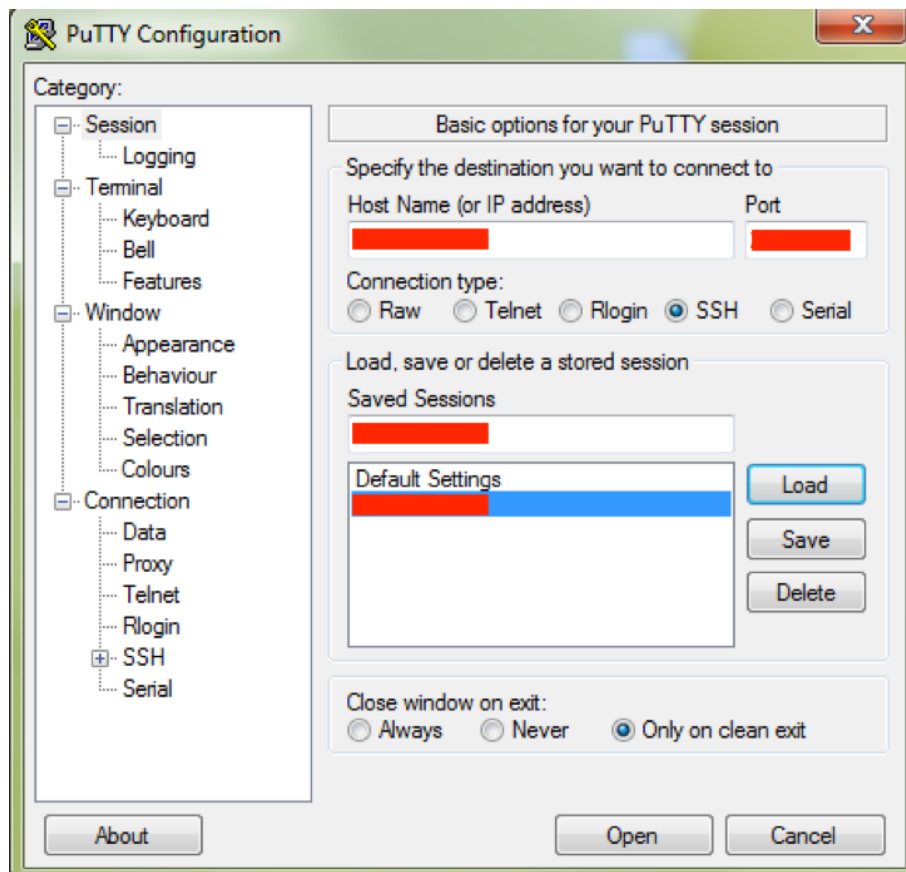


Figura 3.3: Pantalla de configuracion de PuTTY

máquinas que alojan físicamente este Datamart de Riesgos.

En la figura 3.3 se muestra la pantalla de configuración de PuTTY, donde será necesario introducir los parámetros correspondientes para conectar con las máquinas del Datamart, una vez conectados a la red privada a través del cliente VPN.

### 3.5. Toad para Oracle

Toad para Oracle es una herramienta que permite asegurar la máxima productividad en el desarrollo y administración de bases de datos Oracle, permitiendo reducir el tiempo y esfuerzo necesarios para desarrollar, administrar y mantener bases de datos de Oracle.

En la figura 3.4 se muestra una captura de pantalla de Toad for Oracle donde se lleva a cabo la configuración de los diferentes entornos (Desarrollo, Preproducción y Producción).

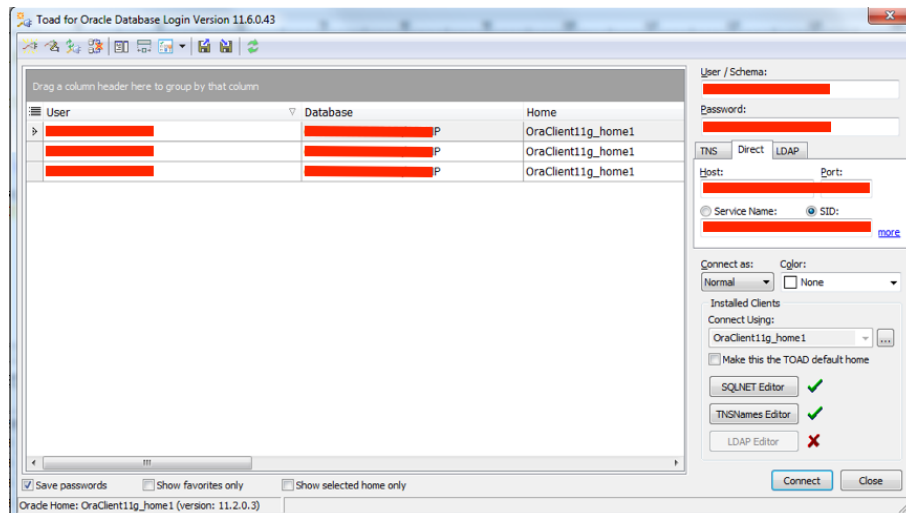


Figura 3.4: Pantalla de configuración de Toad para Oracle

## 3.6. FileZilla

FileZilla es una solución FTP de código abierto y software libre distribuido de manera gratuita bajo la Licencia Pública General GNU. Este software dispone tanto de cliente como de servidor, sin embargo, para el objeto de este Proyecto se usa únicamente el cliente FTP.

El cliente FTP permite la transferencia entre dos máquinas, en el caso de este Proyecto se conecta el cliente FTP a las máquinas donde reside el Datamart con el objetivo de administrar el sistema de ficheros desde entorno Windows desde la interfaz gráfica que proporciona el Cliente FTP de FileZilla. Para su correcto funcionamiento, es necesario configurar los parámetros correspondientes a la IP de la máquina remota, así como el usuario y contraseña.

En la figura 3.5 se presenta una captura de pantalla de la página de configuración del Cliente FTP de FileZilla.

## 3.7. ODBC

Por último, en la realización de este proyecto también es necesario disponer de ODBC configurado con el fin de llevar a cabo análisis de datos y actualización de informes ejecutados a través de Microsoft Excel.

Open DataBase Connectivity (ODBC, por sus siglas) es un estándar de acceso a las bases de datos desarrollado por SQL Access Group. El objetivo de ODBC es hacer posible el acceso a

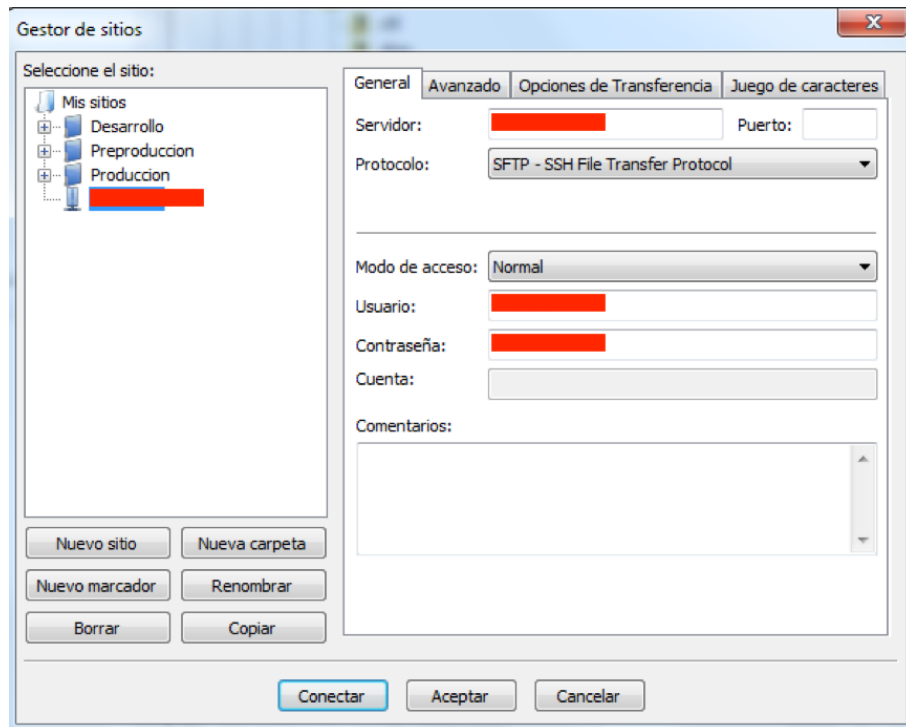


Figura 3.5: Pantalla de configuracion del cliente de Filezilla

cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos almacene los datos.

Para configurar ODBC iremos al Administrador de orígenes de datos y en la pestaña “DNS de sistema” agregaremos tantos orígenes de datos como entornos o esquemas tengamos.

## 3.8. Python

Python es un lenguaje de programación de código abierto o “open-source” que cuenta con la licencia Python Software Foundation License, compatible con la Licencia pública general de GNU. La filosofía de Python hace hincapié en una sintaxis que favorece que el código sea legible, principal característica de este lenguaje de programación. Otras características que hacen de Python un lenguaje de programación muy versátil son las siguientes:

- Es un lenguaje de programación multiparadigma. Esto quiere decir que, más que forzar a los encargados del desarrollo del software a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.

- Por otro lado, se trata de un lenguaje interpretado, es decir, no necesita compilación, con lo que la máquina es capaz de ejecutar la sucesión de instrucciones sin traducción previa. Esto hace que cualquier código sea independiente de las características concretas de la máquina o sistema operativo y, por tanto, es portable.
- Por último, Python usa tipado dinámico: una misma variable puede tomar valores de distinto tipo en diferentes momentos de la ejecución del código.

### **3.9. Shell Scripts**

Los Shell Scripts contienen una serie de comandos de la Shell de Unix que, en la mayor parte de los que se usan en este proyecto, llevan a cabo una serie de operaciones sobre los ficheros y recopilan los parámetros necesarios para la ejecución del SQL correspondiente. Por ello, juegan un papel fundamental en este proyecto, ya que todos los procesos que se ejecutan para llevar a cabo las tareas propias del Datamart de Riesgos se lanzan a través de scripts. Un Shell script es un programa diseñado para ser ejecutado por la Shell de Unix, el intérprete de línea de comandos.



# Capítulo 4

## Diseño e implementación

Este capítulo contiene la parte principal del proyecto, ya que describe el diseño y funcionamiento del Datamart, las tareas principales llevadas a cabo en el soporte y mantenimiento del mismo, así como errores más comunes en su funcionamiento y las metodologías para solventarlos.

### 4.1. Diseño general

El Datamart de riesgos se divide en dos partes: operacional y analítico. En la parte del operacional se lleva a cabo el tratamiento de información y generación de informes relativos a los datos diarios. Por otro lado, en el analítico se lleva a cabo la manipulación de la información relativa a los ciclos de facturación o mensual.

El operacional del Datamart se procesa de manera diaria en un esquema propio de la base de datos. Hay que tener en cuenta que existe cierta información que se recibe y procesa varias veces al día, como es el caso de la información de consumos recibida desde Arbor. Las fuentes de entrada del sistema operacional son, en esencia, las siguientes:

- Clarify: información relativa a la cartera de clientes. Desde Clarify se reciben una vez al día diferentes ficheros, entre ellos: clientes, cuentas, servicios, descuentos, límites, restricciones, promociones, ... En la tabla de la figura 4.1 se muestran los ficheros que se reciben, así como la periodicidad, el volumen de registros y tamaño estimados:
- Spirit: este sistema es, básicamente, el mismo que Clarify donde se está migrando toda la

ENTIDADES CLARIFY		
CLIENTES	Nombre del fichero	Clientes_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,3 Mb (aprox.)
	Número de registros	5.000 (aprox.)
CUENTAS	Nombre del fichero	Cuentas_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,5 Mb (aprox.)
	Número de registros	9.000 (aprox.)
SERVICIOS	Nombre del fichero	Servicios_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	8,5 Mb (aprox.)
	Número de registros	45.000 (aprox.)
DESCUENTOS A NIVEL DE CUENTA	Nombre del fichero	DescCuentas_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,1 Mb (aprox.)
	Número de registros	11.000 (aprox.)
DESCUENTOS A NIVEL DE SERVICIO	Nombre del fichero	DescServicios_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,5 Mb (aprox.)
	Número de registros	14.000 (aprox.)
LÍMITES	Nombre del fichero	Limites_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,2 Mb (aprox.)
	Número de registros	2.500 (aprox.)
RESTRICCIONES SOBRE SERVICIOS	Nombre del fichero	Restricciones_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	2 Mb (aprox.)
	Número de registros	25.000 (aprox.)
SERVICIOS SUPLEMENTARIOS DE VOZ	Nombre del fichero	SSGSM_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	3,5 Mb (aprox.)
	Número de registros	45.000 (aprox.)
SERVICIOS SUPLEMENTARIOS DE DATOS	Nombre del fichero	SSGPRS_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,35 Mb (aprox.)
	Número de registros	4.700 (aprox.)
SERVICIOS PREPAGO	Nombre del fichero	Prepago_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,02
	Número de registros	300
PROMOCIONES	Nombre del fichero	Promociones_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	2,1 Mb (aprox.)
	Número de registros	26.000 (aprox.)
SERVICIOS KO	Nombre del fichero	KO_Servicios_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,003 Mb (aprox.)

Figura 4.1: Entidades Clarify



información. Por lo tanto, para tener toda la información de la cartera de clientes completa es necesario tratar datos que provienen de Spirit. Los ficheros recibidos son equivalentes a los de Clarify y también se reciben una vez al día. En la tabla de la figura 4.2 se muestran los detalles de los ficheros recibidos desde Spirit:

- Arbor: desde el sistema Arbor se recibe la información de consumo de los clientes. Esta información se recibe y procesa cada dos horas. En la tabla de la figura 4.3 se muestran los detalles de los ficheros recibidos desde Arbor:
- CRM: desde CRM se recibe la información de cualquier contacto que tienen los calls centers con los clientes. Se reciben los ficheros una vez al día. En la tabla de la figura 4.4 se muestran los detalles de los ficheros recibidos:
- DWH: desde el Data Warehouse se reciben varios ficheros, todos ellos se reciben y procesan una vez al día.
- HUR: desde este sistema se recibe el detalle concreto del consumo realizado por los clientes. En la tabla de la figura 4.5 se muestran los detalles de los ficheros recibidos:
- Bancos (BNK): desde los bancos se recibe la información relativa a las devoluciones de los recibos. Se reciben ficheros de diferentes bancos una vez al día y se procesan con la misma frecuencia. En la tabla de la figura 4.6 se muestran los detalles de los ficheros recibidos:
- SAP: de SAP se recibe la información relativa a las devoluciones procesadas por este mismo sistema, así como los movimientos contables y la deuda. En las tablas de las figuras 4.7 y 4.8 se muestran los detalles de los ficheros recibidos:
- Informa: se trata de una plataforma que ofrece información acerca de empresas; para este Datamart desde Informa llega la información de las empresas en situación concursal de forma semanal. En la tabla de la figura 4.9 se muestran los detalles del fichero recibido:
- SEPA: desde SEPA se cargan las devoluciones procesadas por este sistema. En la tabla de la figura 4.10 se muestran los detalles del fichero recibido:
- Provisión Masiva. En la tabla de la figura 4.11 se muestran los detalles del fichero recibido:

ENTIDADES SPIRIT		
CLIENTES	Nombre del fichero	Cientes_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	230 Mb (aprox.)
	Número de registros	4.500.000 (aprox.)
CUENTAS	Nombre del fichero	Cuentas_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	600 Mb (aprox.)
	Número de registros	4.700.000 (aprox.)
SERVICIOS	Nombre del fichero	Servicios_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1.500 Mb (aprox.)
	Número de registros	7.000.000 (aprox.)
LÍMITES	Nombre del fichero	Limites_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	27 Mb (aprox.)
	Número de registros	285.000 (aprox.)
RESTRICCIONES SOBRE SERVICIOS	Nombre del fichero	Restricciones_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	57 Mb (aprox.)
	Número de registros	760.000 (aprox.)
SERVICIOS SUPLEMENTARIOS DE VOZ	Nombre del fichero	SSGSM_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	400 Mb (aprox.)
	Número de registros	5.750.000 (aprox.)
SERVICIOS SUPLEMENTARIOS DE DATOS	Nombre del fichero	SSGPRS_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0 Mb (aprox.)
	Número de registros	0 (aprox.)
SERVICIOS PREPAGO	Nombre del fichero	Prepago_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0 Mb (aprox.)
	Número de registros	0 (aprox.)
PROMOCIONES	Nombre del fichero	Promociones_????????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	100 Mb (aprox.)
	Número de registros	4.500.000 (aprox.)

Figura 4.2: Entidades Spirit

ENTIDADES ARBOR		
CONSUMO	Nombre del fichero	clientes_alarma_????????_?????.dat
	Periodicidad	Diaria (cada 2 horas).
	Tamaño del fichero	370 Mb (aprox.)
	Número de registros	3.000.000 (aprox.)

Figura 4.3: Entidades Arbor

ENTIDADES CRM		
CASOS	Nombre del fichero	Casos_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	7,2 Mb (aprox.)
	Número de registros	47.000 (aprox.)
SUBCASOS	Nombre del fichero	Subcasos_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,2 Mb (aprox.)
	Número de registros	9.500 (aprox.)
INTERACCIONES	Nombre del fichero	Interacciones_???????_?????.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,6 Mb (aprox.)
	Número de registros	11.000 (aprox.)

Figura 4.4: Entidades CRM

ENTIDADES HUR PANDORA		
CONSUMO ROAMING	Nombre del fichero	HUR_Roaming_?????????????.dat
	Periodicidad	Diaria (cada 2 horas).
	Tamaño del fichero	0,016 (aprox.)
	Número de registros	130 (aprox.)

Figura 4.5: Entidades HUR Pandora

ENTIDADES BNK		
DEVOLUCIONES RETENIDAS POR EL BANCO LA	Nombre del fichero	Refact*_*_cxa_EUR.dat
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,6 Mb (aprox.)
	Número de registros	5.000 (aprox.)
DEVOLUCIONES RETENIDAS POR EL BANCO BBVA	Nombre del fichero	*.NS*.out
	Periodicidad	Ciclada a partir de la fecha de vencimiento.
	Tamaño del fichero	0,5 Mb (aprox.)
	Número de registros	5.000 (aprox.)
DEVOLUCIONES RETENIDAS POR EL BANCO	Nombre del fichero	Refact*san_*.dat
	Periodicidad	Ciclada a partir de la fecha de vencimiento.
	Tamaño del fichero	0,5 Mb (aprox.)
	Número de registros	5.000 (aprox.)

Figura 4.6: Entidades BNK

ENTIDADES SAP (1)		
DEVOLUCIONES PRE- PROCESADAS POR SAP	Nombre del fichero	exdev_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	53 Mb (aprox.)
	Número de registros	500.000 (aprox.)
DEVOLUCIONES PRE- RECHAZADAS	Nombre del fichero	VFES_SAP*_PP_???_SDD????_????????????????????????????????_????????_?????.pr
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	1,1 Mb (aprox.)
	Número de registros	3.000 (aprox.)
MOVIMIENTOS CONTABLES (APUNTES ESTADÍSTICOS)	Nombre del fichero	estad_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,001 Mb (aprox.)
	Número de registros	30 (aprox.)
	Nombre del fichero	estad_diario_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,07 Mb (aprox.)
	Número de registros	2.000 (aprox.)
MOVIMIENTOS CONTABLES (ABONOS)	Nombre del fichero	exabo_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,37 Mb (aprox.)
	Número de registros	7.500 (aprox.)
	Nombre del fichero	exabo_diario_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,3 Mb (aprox.)
	Número de registros	6.000 (aprox.)
MOVIMIENTOS CONTABLES (FACTURAS ORDINARIAS)	Nombre del fichero	fctor_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	3 Mb (aprox.)
	Número de registros	65.000 (aprox.)
	Nombre del fichero	fctor_diario_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	3 Mb (aprox.)
	Número de registros	65.000 (aprox.)
MOVIMIENTOS CONTABLES (TARJETAS)	Nombre del fichero	pgtar_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,02 Mb (aprox.)
	Número de registros	600 (aprox.)
	Nombre del fichero	pgtar_diario_???????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,07 Mb (aprox.)
	Número de registros	2.000 (aprox.)

Figura 4.7: Entidades SAP (1)

ENTIDADES SAP (2)		
MOVIMIENTOS CONTABLES (PROMESAS DE PAGO)	Nombre del fichero	proms_????????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,05 Mb (aprox.)
	Número de registros	1.000 (aprox.)
	Nombre del fichero	proms_diario_????????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,08 Mb (aprox.)
	Número de registros	2.000 (aprox.)
MOVIMIENTOS CONTABLES (TRANSFERENCIAS)	Nombre del fichero	trans_????????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,03 Mb (aprox.)
	Número de registros	900 (aprox.)
	Nombre del fichero	trans_diario_????????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	0,04 Mb (aprox.)
	Número de registros	1.200 (aprox.)
DEUDA	Nombre del fichero	recat_????????_?????.txt
	Periodicidad	Diaria (una vez al día).
	Tamaño del fichero	115 Mb (aprox.)
	Número de registros	1.300.000 (aprox.)
CALENDARIO	Nombre del fichero	calendario_sap_????????_?????.txt
	Periodicidad	Semanal (Martes).
	Tamaño del fichero	0,1 Mb (aprox.)
	Número de registros	1.300 (aprox.)
REMESAS	Nombre del fichero	OUT_VFES_SAP*_PP_???_SDD????_????????????????????????????_????????_?????.dat
	Periodicidad	Cuando mandan el fichero.
	Tamaño del fichero	10 Mb (aprox.)
	Número de registros	40.000 (aprox.)

Figura 4.8: Entidades SAP (2)

ENTIDADES INFORMA		
SITUACIONES CONCURSALES	Nombre del fichero	I_Situaciones_Concursales_????????_?????.txt
	Periodicidad	Semanal
	Tamaño del fichero	0,5 Mb (aprox.)
	Número de registros	1.000 (aprox.)

Figura 4.9: Entidades Informa

ENTIDADES SEPA		
DEVOLUCIONES SEPA	Nombre del fichero	????????????VFES_RP_PP_*_????????_?????.csv
	Periodicidad	Cuando mandan el fichero.
	Tamaño del fichero	20 Mb (aprox.)
	Número de registros	340.000 (aprox.)

Figura 4.10: Entidades SEPA

OTRAS ENTIDADES		
PROVISIÓN MASIVA LÍMITES	Nombre del fichero	Sal_??_Prov_Mas_LimCred_Integrado_????????_?????.dat
	Periodicidad	Diaria
	Tamaño del fichero	0,2 Mb (aprox.)
	Número de registros	10 (aprox.)

Figura 4.11: Entidades Otras

ENTIDADES AGEX		
CARTERA AGEX	Nombre del fichero	TLF_TDX*_?????????.txt
	Periodicidad	Diaria.
	Tamaño del fichero	0,2Mb (aprox.)
	Número de registros	25.000 (aprox.)

Figura 4.12: Entidades AGEX

- Asignaciones.
- Agex. Las agencias externas envían su cartera para su inclusión en el Datamart. En la tabla de la figura 4.12 se muestran los detalles del fichero recibido:
- Diccionarios. La información que se carga de Diccionarios es muy estática, es decir, no suele haber cambios de una ejecución a otra (se trata de tablas descriptivas). A pesar de ello, se reciben de manera diaria.

El sistema analítico del Datamart se procesa de manera ciclada, es decir, después de cada ciclo de facturación. La información del sistema analítico está en un esquema de la base de datos diferente al operacional. Las fuentes de entrada del sistema analítico son:

- DWH: después de cada ciclo de facturación, desde el Datawarehouse se recibe en el Datamart de Riesgos la información de cartera, facturación, cambio de direcciones, promociones y llamadas pasarela.
- BO: desde BO se recibe la información correspondiente a la cartera y a los cobros. En la tabla de la figura 4.13 se muestran los detalles de los ficheros recibidos:
- Adquisiciones: desde Adquisiciones se recibe la información relativa a las solicitudes de alta, portabilidades, migraciones y cambios de titular, junto con una serie de indicadores. En la tabla de la figura 4.14 se muestran los detalles del fichero recibido:

ENTIDADES BO		
CARTERA USAC	Nombre del fichero	CC_Usac_????????.dat
	Periodicidad	Ciclada (después de cada ciclo de facturación).
	Tamaño del fichero	40 Mb comprimido (aprox.)
	Número de registros	4.200.000 (aprox.)
COBROS	Nombre del fichero	CC_Cobros_????????.dat
	Periodicidad	Ciclada (después de cada ciclo de facturación).
	Tamaño del fichero	8 Mb comprimido (aprox.)
	Número de registros	300.000 (aprox.)

Figura 4.13: Entidades BO

ENTIDADES ADQUISICIONES		
ADQUISICION ES	Nombre del fichero	Pendiente de definir el nombre definitivo.
	Periodicidad	Ciclada (después de cada ciclo de facturación).
	Tamaño del fichero	115 Mb (aprox.)
	Número de registros	500.000 (aprox.)

Figura 4.14: Entidades Adquisiciones

En la figura 4.15 se muestra un mapa de todos los sistemas de los que se extrae información para su procesamiento y tratamiento en el Datamart de Riesgos. Así mismo, se muestran los informes que el Datamart produce como salida y que se detallarán más adelante en este Proyecto.

## 4.2. Monitorización de la ejecución

Para la monitorización diaria de las ejecuciones se dispone de un script que muestra la salida desde la propia línea de comandos. Este script lo que hace es comprobar los logs que generan los diferentes procesos: si alguno de los procesos no finaliza correctamente, en el correspondiente log aparecerá el error y este script mostrará por pantalla una alerta; del mismo modo, si los procesos periódicos no se han ejecutado cuando correspondía (por ejemplo, porque no han llegado los ficheros necesarios) también mostrará una alerta. Este script también refleja cuando un proceso se encuentra en ejecución. Para ejecutar el script, nos conectaremos a la VPN mediante el Cliente VPN de Cisco, ejecutaremos PuTTY e iniciaremos sesión en la máquina UNIX. Para lanzar la ejecución del script, nos situaremos en la carpeta donde se encuentra mediante el comando “cd” seguido de la carpeta y posteriormente ejecutaremos el script con “ksh COMPROBAR\_EJECUCIONES”. En la figura 4.16 se muestra la salida por línea de comandos que proporciona este script.

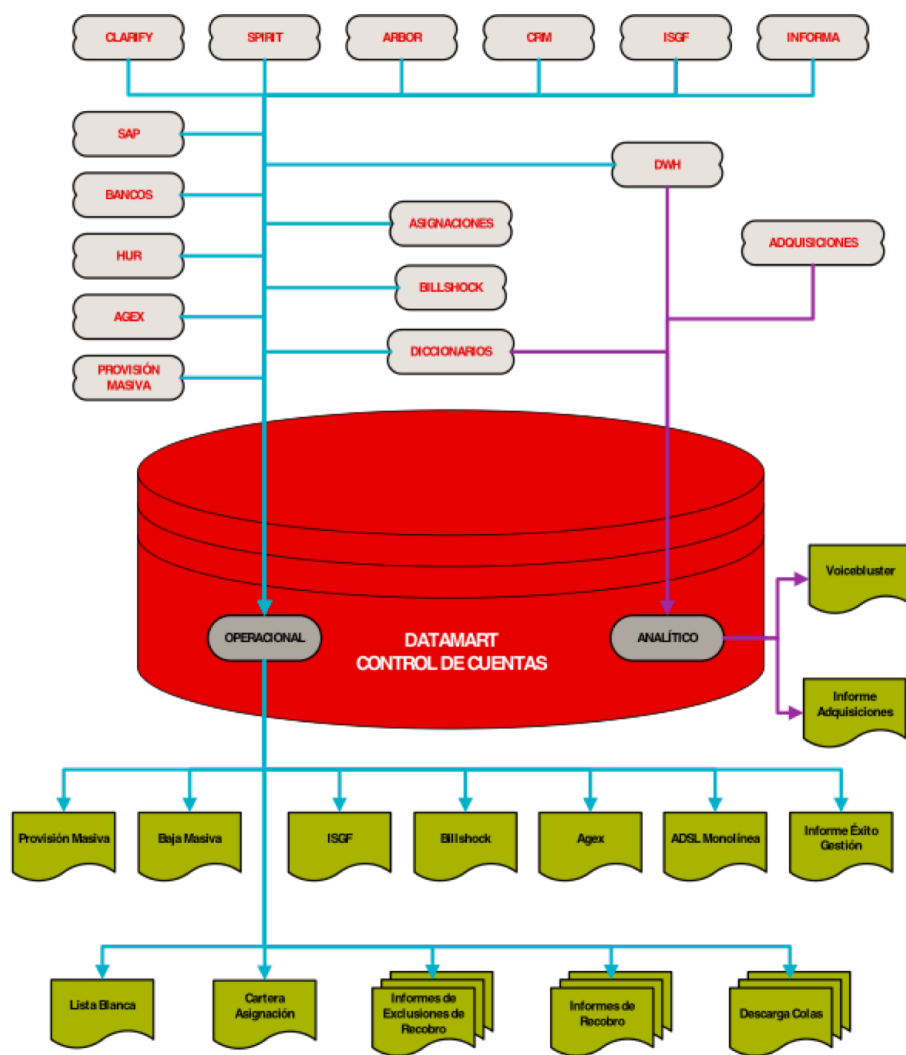


Figura 4.15: Mapa de sistemas Datamart de riesgos



```

kx@kxserver:~$ ksh COMPROBAR_EJECUCIONES
-----
EJECUCIONES ENCONTRADAS
-----
LANZA_ARBOR      FIN      log/diario/rel/rel_consumo_20140404_101631.log      0404_102919 -> FIN
LANZA_BO         FIN      log/diario/rel/rel_consumo_offnet_20140404_082341.log 0404_082916 -> FIN
LANZA_CARTERA    FIN      log/diario/rel/rel_duplicados_cartera_20140404_105556.log 0404_105752 -> FIN
LANZA_CONTROL_CLIE FIN      log/diario/inf/inf_control_clientes_20140404_091454.log 0404_092008 -> FIN
LANZA_CRM        FIN      log/diario/rel/rel_interac_estado_actual_20140404_094729.log 0404_095718 -> FIN
LANZA_DEUDA_12   FIN      log/diario/rel/rel_deuda_20140404_081723.log      0404_084333 -> FIN
LANZA_DEUDA_7    FIN      log/diario/inf/inf_informe_recobro_detalle_20140404_085612.log 0404_090927 -> FIN
LANZA_DEV_BNK    FIN      log/informes                                     0404_092953 -> FIN
LANZA_DEV_BNK_RET FIN      log/diario/rel/rel_devolucion_banco_ret_20140404_092416.log 0404_092953 -> FIN
LANZA_DEV_RECIBO  FIN      log/informes                                     0404_174035 -> FIN
LANZA_DEV_SEPA   FIN      log/diario/rel/rel_devolucion_procesada_20140403_171921.log 0404_100432 -> FIN
LANZA_DEV_SEPA   FIN      log/diario/rel/rel_devolucion_sepa_20140404_100131.log 0404_021238 -> FIN
LANZA_DIARIO     FIN      log/informes/inf/inf_lista_blanca_20140404_002315.log 0404_105440 -> INI -> FICHERO dimCola.sql
LANZA_DIM        EJECE   log/diario/dim/dimCola_20140404_105336.log          0404_091619 -> FIN
LANZA_DWH        FIN      log/diario/rel/rel_traspaso_saldo_20140404_091407.log 0404_102054 -> INI -> FICHERO dimHistoricoExclusion.sql
LANZA_EXCLUSIONES EJECE   log/diario/dim/dimHistoricoExclusion_20140404_102054.log 0404_111738 -> FIN
LANZA_HUR        FIN      log/diario/rel/rel_consumo_20140404_111716.log      0403_130207 -> FIN
LANZA_INF_ADSEL  FIN      log/informes/inf/inf_adsl_pincho_20140403_122331.log 0403_130207 -> FIN
LANZA_INF_AGEEX  FIN      log/informes                                     0404_111125 -> FIN
LANZA_INF_BILL   FIN      log/diario/rel/rel_interaccion_billshock_20140404_111105.log 0403_183320 -> FIN
LANZA_INTERA_BIL FIN      log/informes/inf/inf_isgf_20140403_182657.log       0404_064215 -> FIN
LANZA_ISGF       FIN      log/diario/rel/rel_limite_estado_actual_20140404_064151.log 0404_105752 -> FIN
LANZA_LIMITE     FIN      log/diario/rel/rel_duplicados_cartera_20140404_105556.log 0404_094733 -> FIN
LANZA_LIMPIAR    FIN      log/diario/inf/inf_limite_baja_masiva_result_20140404_094733.log 0404_181318 -> FIN
LANZA_MASIVO_14  FIN      log/diario/rel/rel_mvto_contable_20140403_180258.log 0404_021310 -> FIN
LANZA_MASIVO_22  FIN      log/diario/rel/rel_mvto_contable_diario_20140404_020226.log 0404_031219 -> FIN
LANZA_MVTO       FIN      log/diario/inf/inf_mvto_contable_diario_20140404_015006.log 0404_031219 -> FIN
LANZA_NO_COLAS   FIN      log/informes
LANZA_REPESA     FIN      log/informes
kx@kxserver:~$

```

Figura 4.16: Salida por línea de comandos del script COMPROBAR\_EJECUCIONES

Los procesos los podemos diferenciar en aquellos cuya ejecución está programada y aquellos cuya ejecución se lleva a cabo bajo demanda. En los siguientes apartados se detallan los diferentes procesos, así como las cuestiones a tener en cuenta para el soporte y mantenimiento de los mismos.

### 4.3. Ejecución programada de los procesos

Para controlar la ejecución programada hay que fijarse en la periodicidad de los diferentes procesos, ya que puede ser que se ejecuten una sola vez al día o varias. Para ello, existen dos ficheros de configuración en los cuales se lleva a cabo la planificación de los procesos: `./cfg/ctl_lanzador.cfg` y `./cfg/ctl_planificador.cfg`.

El lanzador, `ctl_lanzador.cfg`, contiene los scripts que componen los lanzadores de tareas que se tienen que ejecutar, mientras que, en el planificador, `ctl_planificador.cfg`, se configura la periodicidad y horas en las que estas tareas se ejecutan. Por ejemplo, en el fichero del planificador nos podemos encontrar una línea como la siguiente:

```
04|00|10|08|00|/home/kxenuser/bin/LANZADOR-LANZA_CARTERA
```

Esto nos indica que el proceso `LANZA_CARTERA` se empieza a ejecutar mediante el lanzador a las 04:00AM, comprobando si llega información nueva a sus respectivas carpetas cada

10 minutos. Si han llegado los ficheros, se ejecuta LANZA\_CARTERA y se procesan la información, en caso contrario, se sigue esperando hasta las 08:00AM.

En caso de que alguno de los procesos falle, se debe revisar el log de ejecución que se encuentra en ./log/FICHERO\_ERRORES\_LANZADOR. Aquí se puede identificar el fallo que se ha producido y, para analizar en detalle el error, se debe buscar el log del script concreto que falló en la ruta correspondiente.

Una vez localizado y resuelto el error y antes de relanzar el proceso, hay que eliminar el error del log FICHERO\_ERRORES\_LANZADOR, ya que el lanzador no permite ejecutar procesos que presentan error en dicho log. Una vez se ha limpiado el log de errores, se lanza por línea de comandos el proceso correspondiente:

*nohup ksh LANZADOR "LANZADOR QUE FALLÓ" &*

De cara a optimizar los recursos del Datamart, se deben comentar en el fichero ctl\_lanzador.cfg los scripts del proceso, dejando sin comentar únicamente el que falló y sus sucesores, volviendo a descomentarlos una vez haya finalizado para que no interfiera en sucesivas ejecuciones.

Por último, es importante tener en cuenta que algunos procesos dependen de otros, con lo que se debe controlar cuidadosamente la ejecución de aquellos procesos predecesores para que realicen sus tareas en plazo y no retrasen la ejecución del proceso dependiente.

## 4.4. Ejecución bajo demanda de los procesos

En el momento de realización de este proyecto se disponía de tres procesos que se ejecutaban bajo demanda. La monitorización de estos procesos también se lleva a cabo mediante el script COMPROBAR\_EJECUCIONES.

Para los procesos que se ejecutan bajo demanda es necesario que el solicitante nos remita el fichero a procesar. Al ser un proceso más manual que en el caso de ejecuciones programadas, es altamente recomendable, antes de procesar el fichero en cuestión, comprobar que nos lo envían con la información bien formada y con los campos correctos. Para ello, con el siguiente comando podemos detectar de una manera rápida si los campos son los adecuados:

*head TLF\_TDX\_?????????.txt*

## 4.5. Fallos comunes en la ejecución diaria y tratamiento

### 1. Error en una tabla

En el caso de producirse algún error que afecte a alguna de las tablas que usan varios procesos, habrá que desplanificar la ejecución de dichos procesos para evitar un bucle de errores que afecte a más tablas del Datamart.

Cuando tengamos que desplanificar algún proceso y luego queramos que se lance, lo volveremos a planificar con la hora actual (por ejemplo, si son las 14:00, en el planificador lo pondremos para que se lance a las 14:05). Habrá que tener especial cuidado al realizar esto, ya que el planificador a las 00:00 detiene todos los procesos y lee la planificación que haya configurada en ese momento. El problema radica en que, si tenemos algún proceso que tenga que estar ejecutándose continuamente, al replanificarlo a otra hora no se volverá a lanzar al día siguiente hasta esa hora. Para evitar esto, una vez se haya lanzado el proceso, en el planificador lo volvemos a cambiar para ponerlo a su hora original.

Ejemplo de uso: hay un problema en la tabla SERVICIOS\_V. Desplanificamos todas las tareas que usan este relacional hasta que solventemos el problema. Una vez solucionado, las volvemos a poner en el planificador con la hora actual + 5 minutos, y una vez se hayan lanzado, las volvemos a cambiar con la planificación original.

### 2. Procesos con dependencias

Como se ha indicado anteriormente, existen procesos que presentan dependencias con otros procesos. Esto queda reflejado en el fichero de configuración `ctl_dependencias.cfg`, que contiene configuración como la siguiente:

```
LANZA_EXCLUSIONES;pre_exclusiones_laborable.sh;LANZA_DEUDA_12;rel_deuda.sh;0;36000
```

```
LANZA_EXCLUSIONES;pre_exclusiones_laborable.sh;LANZA_CRM;rel_interac_estado_actual.sh;0;36000
```

```
LANZA_EXCLUSIONES;pre_exclusiones_laborable.sh;LANZA_CRM_BO;rel_interaccion_cuenta_cont.s
```

```
LANZA_EXCLUSIONES;pre_exclusiones_laborable.sh;LANZA_MVTO_DIA;rel_mvto_contable_diario.s
```

En este ejemplo, vemos que `LANZA_EXCLUSIONES` depende de las tareas `LANZA_DEUDA_12`, `LANZA_CRM`, `LANZA_CRM_BO` y `LANZA_MVTO_DIA`. En concreto la dependencia consiste en que los scripts que ahí se indican se deben haber ejecutado correctamente en el

día en el que se está intentando ejecutar LANZA\_EXCLUSIONES. Si fallase la ejecución de, por ejemplo, LANZA\_CRM\_BO, habría que seguir los siguientes pasos:

- Encontrar el fallo por el que la cadena CRM\_BO falló, solucionarlo y relanzar esta cadena.
- *ksh ultima\_ejecucion\_correcta.sh LANZA\_EXCLUSIONES inf\_informe\_exclusion.sh.*  
El script *ultima\_ejecucion\_correcta.sh* lo que hace es forzar a que se detecte como correcta la ejecución de los scripts.

### 3. CRM

CRM es uno de los procesos que se ejecutan diariamente de forma programada. Este proceso lo que hace es cargar en el Datamart las interacciones que han tenido los clientes con el servicio de atención al cliente, categorizando esas interacciones en CASOS, SUBCASOS e INTERACCIÓN. Es habitual que la ejecución de este proceso falle debido a que los ficheros necesarios no llegan uno o varios días seguidos. A modo ilustrativo, a continuación se detallan los pasos que habría que seguir para solventar esta situación.

- En primer lugar, una vez reclamados y obtenidos los ficheros, se dejan en la ruta de recepción correspondiente. Para el movimiento de ficheros podemos utilizar la línea de comandos a través de PuTTY o bien podemos utilizar la interfaz gráfica de FileZilla.
- A continuación, en el fichero de configuración del lanzador, *ctl\_lanzador.cfg*, se modificará la parte de LANZA\_CRM tal y como se muestra en la figura 4.17, comentando todos los scripts pertenecientes a este proceso, excepto los scripts encargados de cargar los ficheros en tablas de *staging*. En estas tablas se cargan los datos brutos de los ficheros.
- Posteriormente, mediante línea de comandos ejecutamos el proceso con la instrucción *nohup ksh LANZADOR LANZA\_CRM &*.
- Una vez completado el proceso de carga de datos, comprobamos a través de Oracle que la información se ha cargado correctamente en las diferentes tablas de *staging* (STG\_CRM\_CASOS, STG\_CRM\_SUBCASOS, STG\_CRM\_INTERACCIONES).

```
#####
#### RECLAMACIONES (Nuevo CRM) #####
#####
#LANZA_CRM;NO;NO;LANZA_CRM;/home/kxenuser/bin/diario/pre/pre_ficheros_crm.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/stg/stg_crm_casos.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/stg/stg_crm_interacciones.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/stg/stg_crm_subcasos.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dic_tipo_reclamacion.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dic_subtipo_reclamacion.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dic_razon_reclamacion.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dic_resolucion_reclamacion.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dic_grupo_trabajo.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diccionario/rel/dicCola_reclamacion.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/ctl/ctlCola_carga.sh-RECLAMA-E
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_caso_cliente.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_caso_cuenta.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_caso_servicio.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_caso_cliente.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_caso_cuenta.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_caso_servicio.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_caso_estado_actual.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/ctl/ctlCola_carga.sh-RECLAMA-S
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_subcaso.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_subcaso.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_subcaso_estado_actual.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_interaccion_cliente.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_interaccion_cuenta.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/bad_interaccion_servicio.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_interaccion_cliente.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_interaccion_cuenta.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_interaccion_servicio.sh
#LANZA_CRM;SI;NO;LANZA_CRM;/home/kxenuser/bin/diario/rel/rel_interac_estado_actual.sh
```

Figura 4.17: Modificación fichero ctl\_lanzador.cfg para reprocesado de los ficheros del proceso CRM

- Ahora habrá que borrar de los CASOS y los SUBCASOS la información un día posterior a la fecha del fichero no recibido, por ejemplo, si los ficheros del día 18 de mayo (con fecha del 17) no llegaron, habría que ejecutar lo siguiente en Oracle:

```
delete from STG_CRM_SUBCASOS
where FECHA_CIERRE is null;

delete from STG_CRM_SUBCASOS
where FECHA_CREACION <= TO_DATE('20140518', 'yyyymmdd');

delete from STG_CRM_CASOS
where FECHA_CIERRE is null;

delete from STG_CRM_CASOS
```

- Con este último paso eliminamos los casos posteriores a la fecha del fichero, que no interesa actualizar y los casos que no tuvieron cierre el día de no recepción del fichero. Nos quedamos con los casos cerrados el día del fichero, se hayan creado ese día o no. Del *staging* de interacciones no se elimina nada porque todo son altas nuevas, así que se hace un volcado completo de la información del fichero al relacional.
- Volvemos al fichero ctl\_lanzador.cfg y comentamos de manera inversa, es decir, comentamos los scripts que cargan en *staging* y descomentamos el resto de scripts. Por

último, desde línea de comandos ejecutamos el lanzador igual que la primera vez.

El proceso realizado aquí para corregir este error producido por la ausencia de ficheros es una tipología, pudiendo encontrarnos con otras, como por ejemplo que los ficheros de entrada del proceso contengan información malformada o con duplicados, etc. No existe una forma general de solucionar los problemas que se puedan presentar en el Datamart y, por ello, no se abordan en este proyecto, ya que dependerá de la manera en la que el proceso esté diseñado, en los sistemas de información de entrada (cada sistema ofrece la información de una manera, por ejemplo, unos sistemas pueden enviar información con la vista de un solo día, o con la información histórica) y también depende de las necesidades y consideraciones que el área de negocio tenga.

## **4.6. Rendimiento y optimización del Datamart**

Uno de los aspectos importantes y, en ocasiones olvidado debido al trabajo operativo del día a día, es el de mantener un buen rendimiento en el Datamart, evitando sobrecargas que puedan provocar la caída del sistema.

Para ello, resulta de vital importancia revisar y mantener una adecuada planificación de los procesos, evaluando la carga que suponen cada uno de ellos y programando su ejecución de manera que no se solapen varios procesos que sobrecarguen el Datamart.

Del mismo modo, es importante mantener unos procesos eficientes. Este es un aspecto controvertido, ya que en las organizaciones se trabaja en equipos multidisciplinares y con gran número de personas, y no suele existir una única manera correcta de hacer las cosas. La manera de mantener unos procesos eficientes es revisarlos manualmente en busca de comandos innecesarios en los scripts o queries de SQL muy complejas e ineficientes.

Para evaluar el coste de las instrucciones SQL, Oracle cuenta con una utilidad la cual, sin necesidad de ejecutar la query sobre el Datamart, nos evalúa el coste. Con esto nos podemos hacer una idea de la carga que supondrá para el Datamart y, o bien diseñar una query menos costosa para llevar a cabo la misma tarea o, si esto no fuese posible, tenerlo en cuenta a la hora de ejecutar el proceso en el que se encuentre el script que contenga la query en cuestión.

## **Capítulo 5**

### **Resultados**





# Capítulo 6

## Conclusiones

### 6.1. Consecución de objetivos

Esta sección es la sección espejo de las dos primeras del capítulo de objetivos, donde se planteaba el objetivo general y se elaboraban los específicos.

Es aquí donde hay que debatir qué se ha conseguido y qué no. Cuando algo no se ha conseguido, se ha de justificar, en términos de qué problemas se han encontrado y qué medidas se han tomado para mitigar esos problemas.

### 6.2. Aplicación de lo aprendido

Aquí viene lo que has aprendido durante el Grado/Máster y que has aplicado en el TFG/TFM. Una buena idea es poner las asignaturas más relacionadas y comentar en un párrafo los conocimientos y habilidades puestos en práctica.

1. a

2. b

### 6.3. Lecciones aprendidas

Aquí viene lo que has aprendido en el Trabajo Fin de Grado/Máster.

1. a

2. ba

## **6.4. Trabajos futuros**

Ningún software se termina, así que aquí vienen ideas y funcionalidades que estaría bien tener implementadas en el futuro.

Es un apartado que sirve para dar ideas de cara a futuros TFGs/TFM.

# **Apéndice A**

## **Manual de usuario**



# **Bibliografía**