

# UOC - La Universidad Virtual

## **Ingeniería Técnica Informática de Sistemas**

### Impresión en un entorno corporativo

Alumno/a: Oscar Lozano Cañadilla

Dirigido por: Jordi Fernández González

CURSO 2004-05

## Resumen

---

Cuando el director de mi trabajo fin de carrera me pregunto si tenía alguna idea concreta, mejor si implicaba experiencia laboral, para realizar este trabajo yo lo tuve claro: La impresión en un entorno corporativo. El objetivo estaba definido: un problema real, en el que pudiera contar con experiencia laboral: la preocupación por contar con un sistema de reporte de uso de los dispositivos de impresión de una empresa de cierto tamaño. Prestando especial interés en la valoración económica del uso de estos dispositivos de impresión. Así pues, en las próximas páginas el lector encontrará los requisitos, el análisis y diseño de un sistema software para el estudio de la impresión en un entorno corporativo. En cuanto al método seguido esta basado en ciclo de vida Rational Unified Process. El resultado será válido para la posible codificación en un lenguaje orientado a objetos del sistema, basándonos en los requisitos previos. Este sistema permitirá conocer el uso de los sistemas de impresión, de forma que se puedan tomar decisiones que faciliten no solo el ahorro de costes, sino también un uso más racional de dichos dispositivos.

# Índice

---

Glosario.....	Pág. 4
Introducción.....	Pág. 5
Alcance.....	Pág. 7
Recogida y documentación de requisitos.....	Pág. 8
Información inicial.....	Pág. 8
Definir requisitos funcionales.....	Pág. 9
Definir requisitos no funcionales.....	Pág. 9
Establecer fuentes de información.....	Pág. 10
Definir contexto del futuro software.....	Pág. 11
Recogida y clasificación de los guiones.....	Pág. 15
Identificar actores.....	Pág. 16
Identificar casos de uso a partir de los guiones.....	Pág. 17
Identificar relaciones entre casos de uso (extensión, inclusión, especialización).....	Pág. 18
Identificar relaciones de especialización entre actores.....	Pág. 18
Documentación textual y formal de los casos de uso.....	Pág. 19
Análisis.....	Pág. 26
Revisión de los casos de uso.....	Pág. 26
Definir paquetes de análisis.....	Pág. 28
Identificación de las clases de entidades.....	Pág. 29
Especificación de atributos.....	Pág. 30
Identificación de las relaciones entre clases.....	Pág. 31
Identificación de las clases frontera, de control y operaciones.....	Pág. 33
Especificación formal de los casos de uso.....	Pág. 44
Análisis de la interfaz de usuario.....	Pág. 56
Diseño.....	Pág. 61
Diseño arquitectónico.....	Pág. 61
Diseñar casos de uso.....	Pág. 63
Obtener diagrama estático de diseño.....	Pág. 68
Diseñar la persistencia.....	Pág. 69
Diseñar interfaz de usuario.....	Pág. 71
Diseño de los subsistemas.....	Pág. 72
Conclusión.....	Pág. 73
Agradecimientos.....	Pág. 74
Bibliografía.....	Pág. 75

## Glosario

---

### *Empleado.*

Persona física perteneciente a la empresa identificada por un número de empleado.

### *Técnico.*

Empleado que une a su condición de empleado la de mantener y gestionar el sistema de gestión de impresión, lo que le permite determinadas prerrogativas sobre el sistema de las cuales carece el empleado como tal.

### *Sistema.*

Nos referiremos a Sistema cuando queremos hacer referencia a determinadas tareas que son ejecutadas de manera automática, bien por su planificación, bien por que se cumplen determinadas condiciones. Expresaremos que dichas tareas son efectuadas por el "sistema".

### *Documento.*

Archivo de cualquier tipo que se imprime, puede ser un documento de tipo Word, Excel, de tipo texto, etc.

### *Servidor de impresión.*

Ordenador que centraliza las peticiones de impresión y donde se encuentran definidos los dispositivos de impresión.

### *Dispositivo de impresión.*

Elemento capaz e imprimir y que se encuentra conectado a la LAN de la empresa. Por ejemplo impresoras, plotter, etc.

### *Infraestructura.*

Conjunto de elementos Hardware que componen la red informática, sea a nivel LAN o WAN de la empresa.

### *Rational Unified Process.*

Ciclo de vida de desarrollo software ideado por la empresa Rational Software para el desarrollo de software orientado a objetos.

### *SQL Server.*

Gestor de Bases de datos de Microsoft

### *Log.*

Archivo, generalmente de texto que recoge información acerca de la ejecución de una determinada tarea.

### *UML.*

Unified Modeling Language, es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos

# Introducción

---

Un sistema software no se compone únicamente de un conjunto de programas ejecutables, también forman parte de ellas las estructuras de datos usadas, la documentación usada en la construcción y la generada para el usuario del sistema.

Un sistema software ha de ser considerado como un producto, con ciertas particularidades como su persistencia al uso y al tiempo pero no al avance tecnológico, pero un producto al fin y al cabo.

Como todo producto, existen un conjunto de métodos, técnicas y herramientas usadas tanto para el desarrollo del producto como para su fabricación. Este conjunto se denomina ingeniería del software.

Este trabajo fin de carrera tiene un contexto muy definido, la gestión de una parte de la infraestructura informática de una empresa. En concreto, nos referimos a los dispositivos de impresión, elementos en su mayoría impresoras de cualquier tipo (láser, inyección de tinta, blanco y negro, color, etc.)

La elección de este problema, se ha guiado en parte por la experiencia adquirida y en parte por un intento de abordar un problema real que puede darse en una empresa de tamaño medio-grande.

Actualmente en cualquier entorno empresarial y quizás con mayor incidencia en grandes empresas de carácter multinacional se ha impuesto un férreo control de costes, este control tiene como consecuencia la demanda de sistemas informáticos de tipo decisional, es decir que sean capaces de ofrecer información que permita a los responsables adecuados tomar decisiones cuyo objetivo sea reducir costes manteniendo los requerimientos y la satisfacción de los usuarios (empleados de la empresa).

Aun cuando el mercado ofrece herramientas dedicadas a la gestión del hardware de una red informática, este mercado está dominado por los productos Tivoli y OpenView de los fabricantes IBM y HP respectivamente, estos productos están especializados en la gestión de PC's, actualizaciones de software, instalaciones, backup, etc. No ofreciendo funcionalidades que puedan ofrecer datos sobre el uso de los dispositivos de impresión de la red (al menos de momento).

Los dispositivos de impresión tienen un coste el cual no solo viene dado por el coste de adquisición y amortización, sino de los consumibles, además en tanto el coste de amortización depende de un determinado periodo de tiempo, el coste por página impresa está fuertemente influenciado por el uso del dispositivo.

A modo de ejemplo se adjunta un gráfico comparativo del coste de impresión por cada página impresa dependiendo del volumen de impresión mensual para un conjunto de dispositivos de impresión. (Figura 1) Evidentemente el responsable de la infraestructura deberá basar la decisión de instalar uno u otro dispositivo en el menor coste posible, para lo cual es necesario disponer de datos que le indiquen el volumen de impresión al que se enfrentará el dispositivo. De esta forma podrá tomar la decisión más beneficiosa.

El objetivo del sistema de gestión de impresión cuyo análisis vamos a efectuar en este trabajo fin de carrera es dar esta información al responsable de la infraestructura para facilitar la toma de decisiones no solo en el momento de

efectuar una instalación, sino en el futuro podrá, evaluando la información que el sistema le reporte, recomendar cambios que conlleven un ahorro para la compañía.

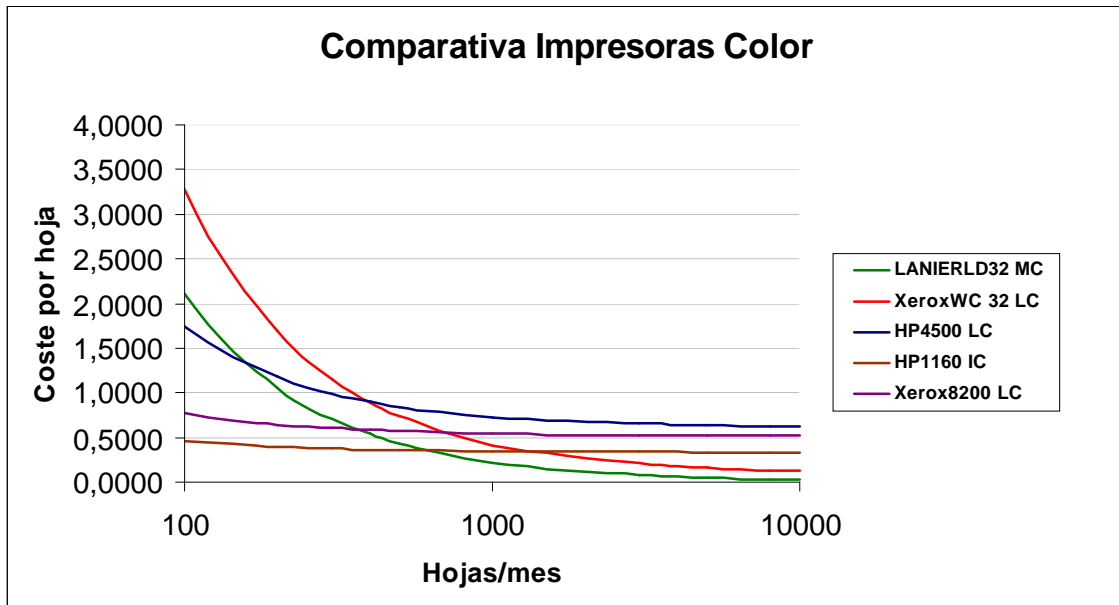


Figura 1. Comparativa de costes de impresión por hoja impresa según volumen mensual de impresión.

## Alcance

---

La producción de software esta constituida por varias etapas, formando todas ellas el denominado ciclo de vida del software. Existen varios modelos de ciclos de vida, y en el caso del desarrollo de software orientado a objetos destaca el llamado Rational Unified Process (de la empresa Rational Software) que es un ciclo de vida iterativo e incremental.

Este ciclo de vida comprende 4 fases bien diferenciadas:

1. Inicio
2. Elaboración
3. Construcción
4. Transición

En cada una de estas fases se realizan los siguientes componentes en distintas proporciones: recogida de requisitos, análisis y diseño, realización y prueba. Cada ejecución de uno o varios componentes es una iteración.

Este TFC aplicara parte de este ciclo de vida para abordar la construcción de un sistema que de seguimiento y análisis del uso de dispositivos de impresión en un entorno corporativo.

En definitiva queda fuera del alcance de este trabajo fin de carrera la codificación/programación, la instalación y mantenimiento del sistema software. Centrándonos en la recogida y documentación de requisitos así como el análisis y diseño del sistema.

# Recogida de documentación y requisitos

## 1.1 Información inicial

Partiremos del resumen de las entrevistas mantenidas con el responsable de la impresión corporativa y los técnicos encargados de la gestión del actual entorno de impresión:

**"...actualmente disponemos de un parque de mas de 200 impresoras repartidas por varias localizaciones, cada localización dispone de un servidor de impresión que se ejecuta bajo Windows 2000 Server. Todas nuestras impresoras están conectadas en red, y están definidas en los servidores de impresión. Todas las impresoras pueden ser usadas por cualquier empleado, inclusive si no se encuentran en la misma localización.**

**Sabemos que el sistema operativo es capaz de generar un fichero con datos de usuario (su numero), la impresora usada (todas están definidas con un nombre único según nuestros estándares), el numero de paginas impresas, el tamaño del documento y la fecha y hora de impresión, además del servidor usado.**

**El sistema ha de ser capaz de recoger estos datos diariamente y alimentar una base de datos SQL Server...**

**En esta Base de Datos se han de poder gestionar las impresoras y servidores, es decir, gestionar altas, bajas, modificaciones por parte de los técnicos encargados de la impresión..."**



## 1.2 Definir requisitos funcionales

Los requisitos funcionales han de describir qué debe realizar el software para el usuario final. En nuestro caso, el sistema de gestión de impresión ha de cumplir los siguientes requisitos funcionales:

- Registrar eventos de impresión con los siguientes datos:
  - o Usuario que imprime
  - o Nombre del documento
  - o Páginas impresas
  - o Tamaño del documento
  - o Impresora usada
- Recoger estos eventos y sus datos de varias fuentes. (servidores de impresión repartidos por la infraestructura corporativa)
- Realizar la carga de estos datos en una Base de Datos.
- Gestionar altas, bajas y modificaciones en la Base de Datos.
- La periodicidad ha de ser diaria.

## 1.3 Definir requisitos no funcionales

Los requisitos funcionales están relacionados con las restricciones que nos impone el entorno tecnológico. En nuestro caso tenemos las siguientes restricciones:

- La infraestructura del cliente esta basada en Windows 2000 (en distintas versiones, Windows 2000, Windows 2000 Server e incluso estaciones Windows XP).
- La base de datos ha de estar integrada con el Standard del cliente, en este caso SQL Server 2000.
- El sistema ha de basarse en la actual infraestructura de impresión.

## 1.4 Fuentes de información

Para la definición de los requisitos anteriormente tratados se ha recurrido a las siguientes fuentes de información:

- Entrevista con el responsable de la impresión corporativa del cliente y los técnicos responsables de la gestión de la actual red de impresión.
- Consulta de los manuales internos relacionados con la gestión de la impresión. (alta de puertos, impresoras, etc.)
- Consultas al fabricante del software usado actualmente en la infraestructura de impresión (Microsoft), estas consultas se han efectuado fundamentalmente en la WEB del fabricante.
- Se ha efectuado una búsqueda de sistemas con una funcionalidad similar ya existentes, es decir de otros fabricantes, como por ejemplo fabricantes de dispositivos de impresión.

## 1.5 Definir contexto del futuro software

Cualquier software esta relacionado con el entorno y la actividad de los usuarios finales del software, desde un punto de vista organizativo e incluso de la terminología a usar, para facilitar el desarrollo es necesario adquirir cierto conocimiento de este contexto, para ello usaremos el modelo de dominio y el modelo de negocio.

- Modelo de dominio

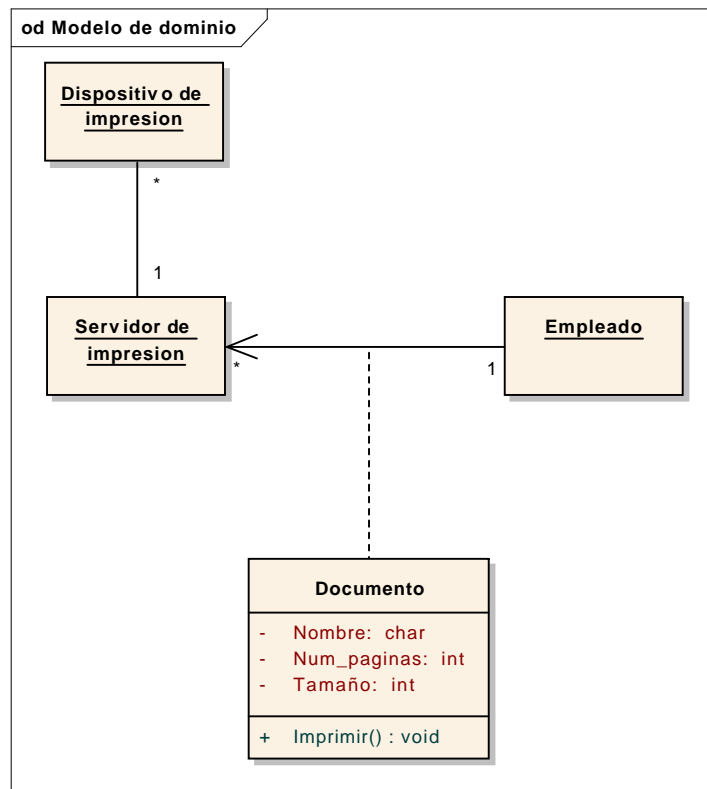


Figura 2. Modelo de dominio.

- Modelo de negocio

El diagrama de casos de uso del modelo de negocio esta representado en la siguiente figura:

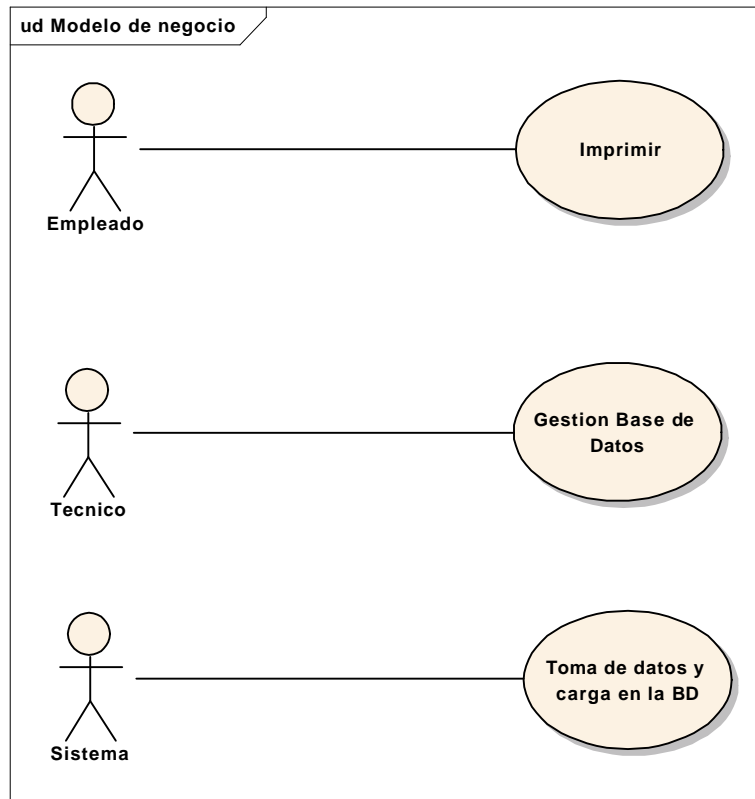


Figura 3. Modelo de negocio.

Para identificar los objetos usados en los casos de uso anteriores utilizaremos diagramas de colaboración para cada caso de uso:

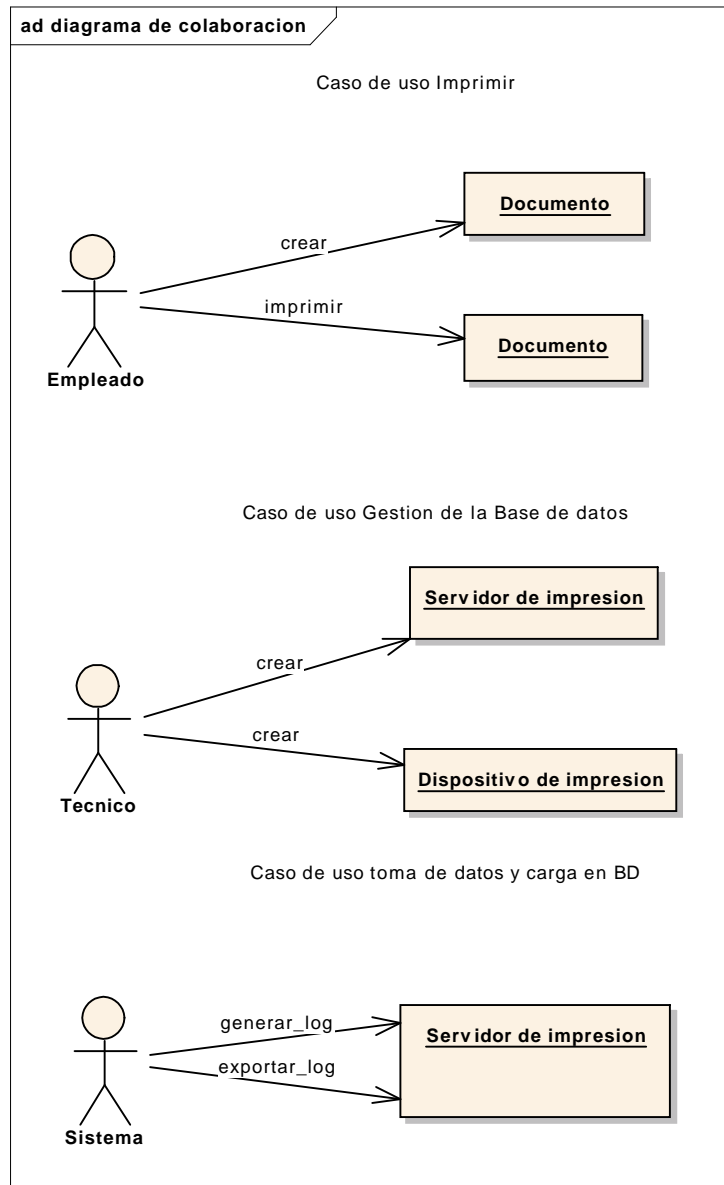


Figura 4. Diagramas de colaboración.

El diagrama de objetos del modelo de negocio es el siguiente:

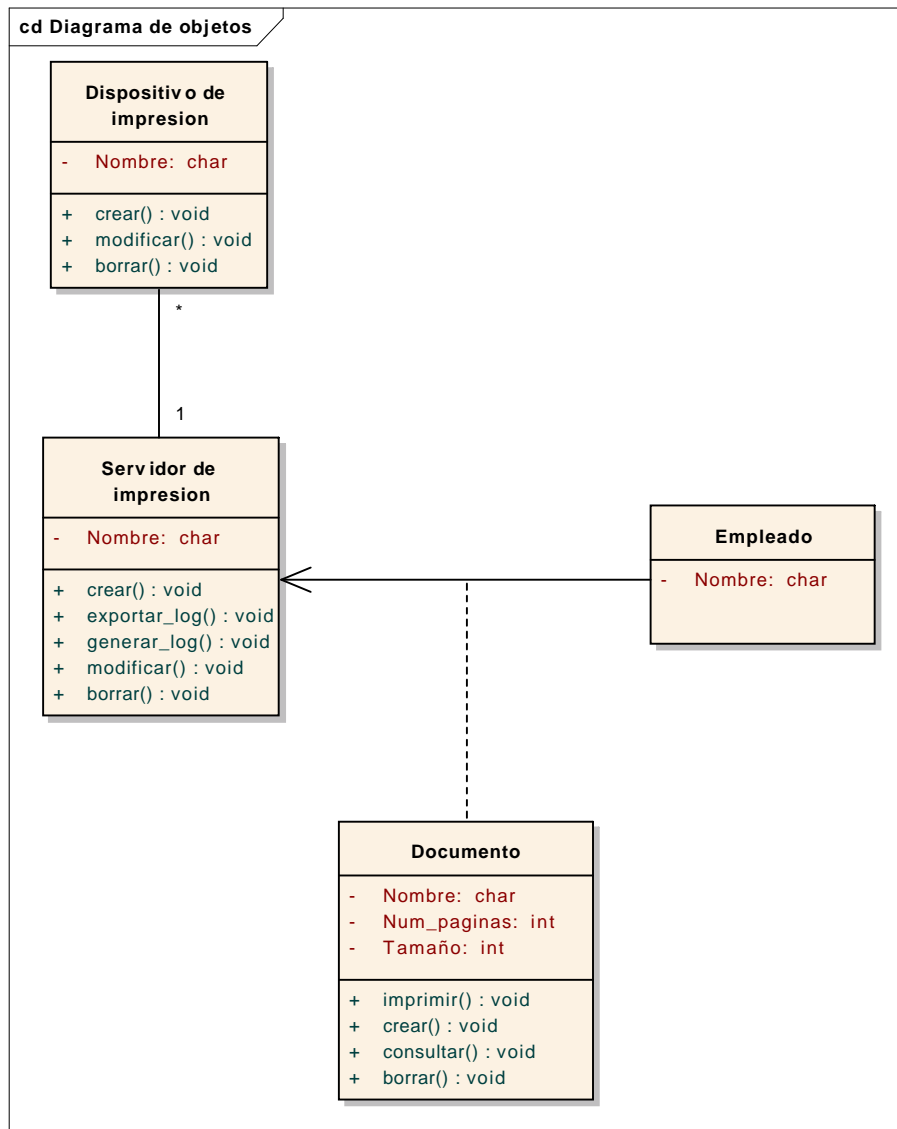


Figura 5. Diagrama de objetos del modelo de negocio.

## 1.6 Recogida y clasificación de los guiones

Después de haber definido el contexto del software, estamos en condiciones de elaborar los guiones necesarios, en nuestro caso tenemos tres guiones:

- Guión del Empleado.

El empleado trabajara con documentos de cualquier tipo, es decir los puede crear, consultar e incluso borrar. Es necesario identificar el documento con un nombre. El interés del empleado es imprimir el documento, para ello una vez sepa el documento o documentos que le interesan, precederá a imprimirlos por el dispositivo de impresión que desee.

- Guión del técnico.

La principal misión del técnico consiste en crear o modificar tanto los servidores de impresión como los distintos dispositivos de impresión existentes.

- Guión del Sistema.

El sistema, de manera automática y atendiendo a la planificación requerida generara un fichero con los datos de impresión de un determinado periodo de tiempo, para posteriormente exportar dicho fichero a la base de datos requerida.

## 1.7 Identificar actores

Se han identificado dos actores, **técnico** y **sistema**. Además, podría pensarse en un tercer actor, **empleado**, sin embargo no tendrá interacción con el "sistema", pues los documentos se generaran e imprimirán con el software habitual por lo que no se considera actor. Sin embargo, tanto el **técnico** como el propio **sistema** si tendrán interacción con el "sistema". Destacar que el actor **sistema** no se corresponde con una persona física, si no con tareas que realizara nuestro "sistema" software.

El actor **técnico** tendrá seis papeles correspondientes a las operaciones de crear, modificar y borrar tanto *servidores de impresión* como *dispositivos de impresión*.

El actor **sistema** tendrá dos papeles correspondientes a las operaciones *generar\_log* y *exportar\_log* de *servidor de impresión*.



## 1.8 Identificar casos de uso a partir de los guiones

Una vez identificados los actores podemos identificar los casos de uso partiendo de los guiones de cada actor implicado en el sistema.

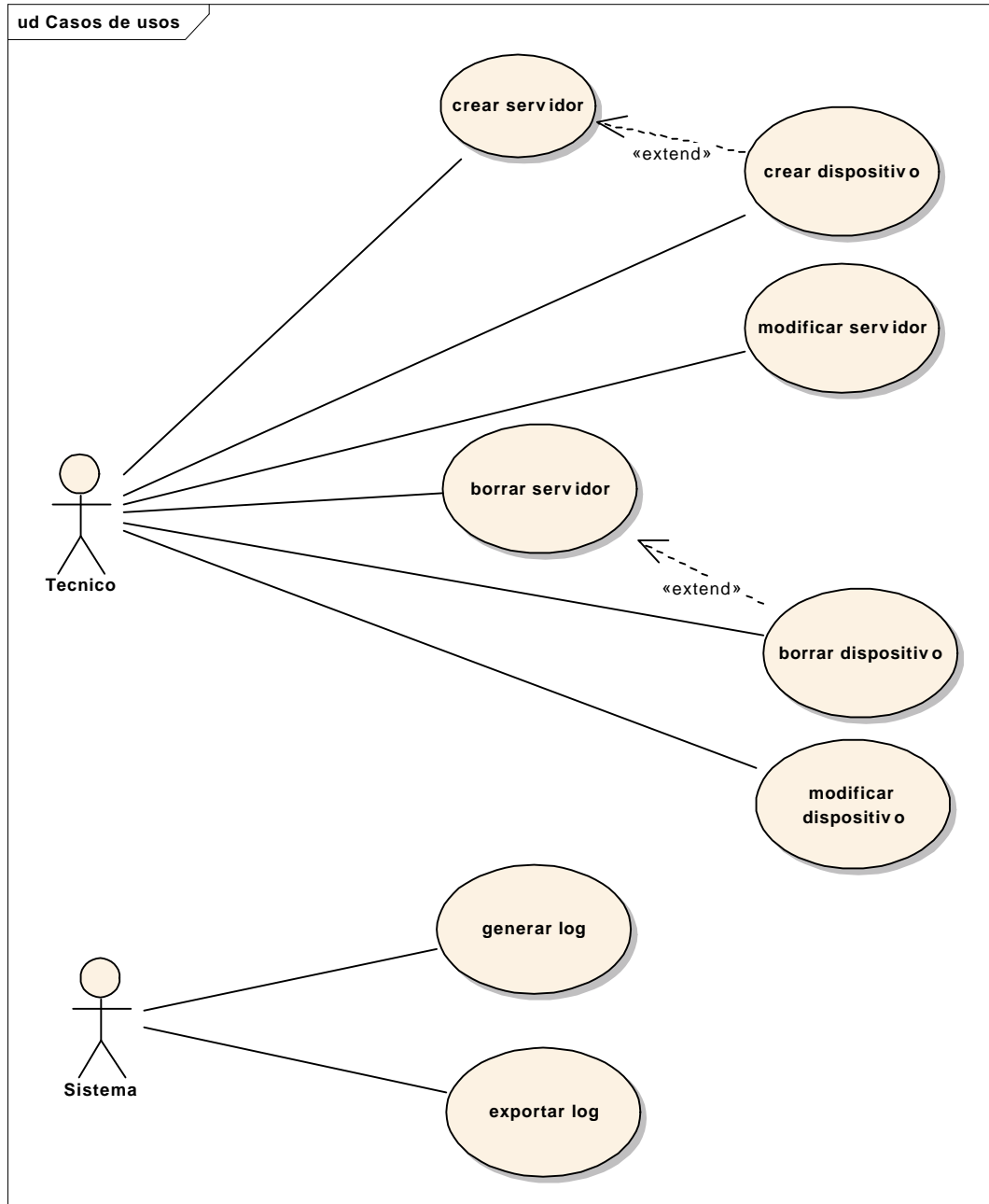


Figura 6. Casos de uso.

## 1.9 Identificar relaciones entre los casos de uso

Podemos comprobar mediante la figura 7 como solo existe una relación entre dos casos de uso, concretamente entre los casos de uso crear servidor y crear dispositivo

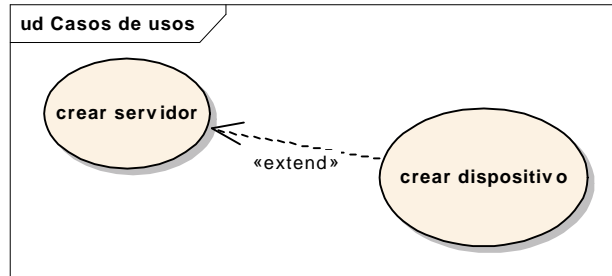


Figura 7. Relación entre casos de uso.

La otra relación entre casos de uso se da entre los casos de uso borrar servidor y borrar dispositivo, tal como muestra la figura 8.

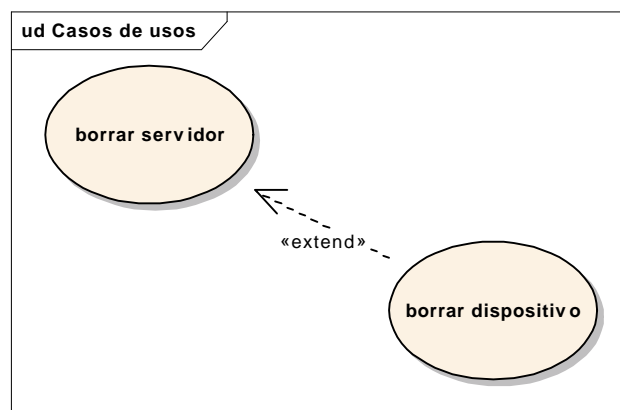


Figura 8. Relación entre casos de uso.

## 1.10 Identificar relaciones de especialización entre actores

Si observamos la figura 6 vemos que no hay relaciones de especialización entre los actores, es decir, no hay ningún caso de uso que puedan hacer ambos actores.

## 1.11 Documentación textual y formal de los casos de USO

Para elaborar la documentación textual de los diferentes casos de uso, utilizaremos una plantilla.

En este TFC utilizaremos una plantilla basada en la utilizada en la asignatura "ingeniería del software" impartida dentro del plan de estudios de Ingeniería Técnica en informática de sistemas impartida por la UOC.

Como se puede comprobar, no todos los casos de uso tienen todos los datos posibles, esto es debido a la etapa en la que nos encontramos (Recogida de documentación).

<b>Nombre del Caso de Uso</b> : crear servidor		<b>Código:</b> 2004TFC001
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso crear servidor, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El servidor de impresión que se esta creando no existe ya.
<b>Poscondiciones</b>	Creado servidor de impresión y dado de alta en el sistema. Opcionalmente se habrán creado un número indeterminado de dispositivos.	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. El Técnico se validara en la aplicación.</li> <li>2. Introducirá los datos necesarios para crear el servidor en el sistema.</li> <li>3. El sistema validara los datos, pedirá confirmación y en caso afirmativo guardara los datos.</li> <li>4. Si no desea dar de alta mas servidores o dispositivos asociados a un servidor abandonara la aplicación.</li> </ol>	
<b>Flujos Alternativos</b>	1. Una vez dado de alta un servidor podrá comenzar a crear dispositivos de impresión asociados a dicho servidor, seguir creando otros servidores o terminar.	
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>	Crear dispositivos	
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : crear dispositivo		<b>Código:</b> 2004TFC002
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso crear dispositivo, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El dispositivo de impresión que se esta creando no existe ya.
<b>Poscondiciones</b>	Creado dispositivo de impresión y dado de alta en el sistema.	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. El Técnico se validara en la aplicación.</li> <li>2. Introducirá los datos necesarios para crear el dispositivo de impresión deseado en el sistema.</li> <li>3. El sistema validara los datos, pedirá confirmación y en caso afirmativo guardara los datos.</li> </ol>	
<b>Flujos Alternativos</b>	<ol style="list-style-type: none"> <li>1. El técnico podrá haber llegado a este caso de uso como extensión del caso de uso crear servidor, no necesitando validación y no siendo necesario indicar a que servidor de impresión estará asociado el dispositivo.</li> </ol>	
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : modificar servidor		<b>Código:</b> 2004TFC003
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso modificar servidor, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El servidor de impresión a modificar ha de estar previamente creado en el sistema.
<b>Poscondiciones</b>	Las modificaciones serán salvadas	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. El Técnico se validara en la aplicación.</li> <li>2. Introducirá los datos del servidor de impresión que desee modificar.</li> <li>3. El sistema validara los datos, pedirá confirmación y en caso afirmativo salvara los cambios.</li> </ol>	
<b>Flujos Alternativos</b>		
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : borrar servidor		<b>Código:</b> 2004TFC004
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso borrar servidor, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El servidor de impresión a borrar ha de estar previamente creado en el sistema. En caso de borrar un servidor de impresión se borrarán todos los dispositivos asociados a el.
<b>Poscondiciones</b>	Borrado del sistema del servidor de impresión y de todos los dispositivos asociados a el.	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. El Técnico se validara en la aplicación.</li> <li>2. Introducirá los datos necesarios del servidor de impresión que se desea borrar.</li> <li>3. El sistema validara los datos, pedirá confirmación y en caso afirmativo borrara los datos. Previamente comprobara si existen dispositivos de impresión asociados al servidor que se desea borrar, en caso afirmativo lo indicara, pidiendo confirmación para proceder a su borrado.</li> </ol>	
<b>Flujos Alternativos</b>		
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>	borrar dispositivo	
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : borrar dispositivo		<b>Código:</b> 2004TFC005
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso crear servidor, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El dispositivo de impresión que se desea borrar esta previamente creado en el sistema.
<b>Poscondiciones</b>	Dispositivo de impresión borrado del sistema.	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. El Técnico se validara en la aplicación.</li> <li>2. Introducirá los datos necesarios para borrar el dispositivo de impresión deseado del sistema.</li> <li>3. El sistema validara los datos, pedirá confirmación y en caso afirmativo borrara el dispositivo.</li> </ol>	
<b>Flujos Alternativos</b>	<ol style="list-style-type: none"> <li>1. El técnico podrá haber llegado a este caso de uso como extensión del caso de uso borrar servidor, no necesitando validación y no siendo necesario indicar a que servidor de impresión estará asociado el dispositivo.</li> </ol>	
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : modificar dispositivo		<b>Código:</b> 2004TFC006
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso modificar dispositivo, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Técnico	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	El dispositivo de impresión a modificar ha de estar previamente creado en el sistema.
<b>Poscondiciones</b>	Las modificaciones serán salvadas	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>4. El Técnico se validara en la aplicación.</li> <li>5. Introducirá los datos del dispositivo de impresión que desee modificar.</li> <li>6. El sistema validara los datos, pedirá confirmación y en caso afirmativo salvara los cambios.</li> </ol>	
<b>Flujos Alternativos</b>		
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		indeterminada
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

<b>Nombre del Caso de Uso</b> : generar log		<b>Código:</b> 2004TFC007
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso generar log, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Sistema	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	Los datos de impresión diarios se recogen del propio log de eventos del sistema.
<b>Poscondiciones</b>	Fichero de texto con los datos diarios de impresión	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. Se solicitara al log de eventos del sistema todos los eventos de impresión.</li> <li>2. Se generara un fichero de texto en una ubicación específica.</li> <li>3. Una vez generado el fichero, se procederá al borrado del log de eventos del sistema.</li> </ol>	
<b>Flujos Alternativos</b>	En caso de existir errores en la copia de ficheros, se generara una entrada en el log de errores de la aplicación	
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		diaria
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		



<b>Casos de Uso relacionados (generalización)</b>	
<b>Prototipo de Pantallas</b>	

<b>Nombre del Caso de Uso</b> : exportar log		<b>Código:</b> 2004TFC008
<b>Autor</b>	Oscar Lozano	
<b>Descripción</b>	Caso de uso exportar log, descripción textual.	
<b>Prioridad (1..10) 1 máx. 10 mín.</b>	1	
<b>Actor/es</b>	Sistema	
<b>Precondiciones:</b>	<b>Casos de Uso</b>	
	<b>Otras</b>	Fichero de log diario previamente generado
<b>Poscondiciones</b>	Fichero exportado.	
<b>Pseudocódigo (flujo normal)</b>	<ol style="list-style-type: none"> <li>1. Se copiara el fichero a una ubicación determinada.</li> <li>2. Comprobada que la copia es correcta, se procederá al borrado del fichero original.</li> </ol>	
<b>Flujos Alternativos</b>	En caso de existir errores en la copia de ficheros, se generara una entrada en el log de errores de la aplicación	
<b>Frecuencia (diaria, mensual, anual, etc.) por usuario</b>		diaria
<b>Casos de uso relacionados (extensión)</b>		
<b>Casos de Uso relacionados (incluidos)</b>		
<b>Casos de Uso relacionados (generalización)</b>		
<b>Prototipo de Pantallas</b>		

## 2. Análisis

---

En la etapa de análisis tendremos tres objetivos claramente identificados, por un lado llevaremos la documentación recogida en la fase de recogida de requisitos a un lenguaje mas formal, que representaremos mediante modelos y diagramas UML, y por otro lado identificaremos las clases fundamentales que servirán para la implementación del software. Finalmente obtendremos la expresión de los casos de uso en términos de las clases fundamentales obtenidas previamente.

En la etapa de análisis describiremos el software de forma orientada a objetos, concretamente nos centraremos en el funcionamiento del software, sin abordar cuestiones relacionadas con la implementación. De hecho, el análisis será totalmente independiente del lenguaje de programación elegido.

### 2.1 Revisión de los casos de uso

En la fase de recogida de documentación y requisitos hemos elaborado unos casos de uso sencillos que usaremos como base para definir los casos de uso en esta fase de análisis.

Por norma general es fácil que en la definición de la documentación se obvien requisitos generalmente relacionados con el propio manejo de la información, es un error muy común pensar que la información es eterna, por lo que hay que prever mecanismos de borrado e incluso procedimientos para pasar información que no será usada con frecuencia a medios de almacenamiento como unidades de cinta o similares que no permanecerán on-line. Si observamos los casos de uso de la figura 6 comprobaremos que aun que se han previsto casos de uso para crear, modificar y borrar tanto dispositivos como servidores, no se contemplan casos de uso para consultar dicha información ni tampoco la generación de informes. Este es el momento de revisar los casos de uso para completarlos con estas carencias.

Así mismo se ha supuesto que el actor sistema será el encargado de generar los log de manera automática, así como de exportarlos. En primer lugar puede ser necesario usar estos casos de uso sin que estemos regidos por la planificación del sistema (actor), y en segundo lugar el caso de uso exportar log peca de sencillo, pues como ya hemos puesto de manifiesto no se puede generar información como si fuese un pozo sin fondo y aun cuando se almacene en una base de datos se ha de prever procedimientos para realizar una limpieza de los datos según una caducidad previamente establecida.

Al consultar tanto un servidor como un dispositivo se puede proceder a modificar e incluso borrar el servidor o dispositivo. Borrar un servidor conlleva el borrado de todos los dispositivos que de el dependen.

En la figura 9 podemos apreciar los casos de uso convenientemente revisados:

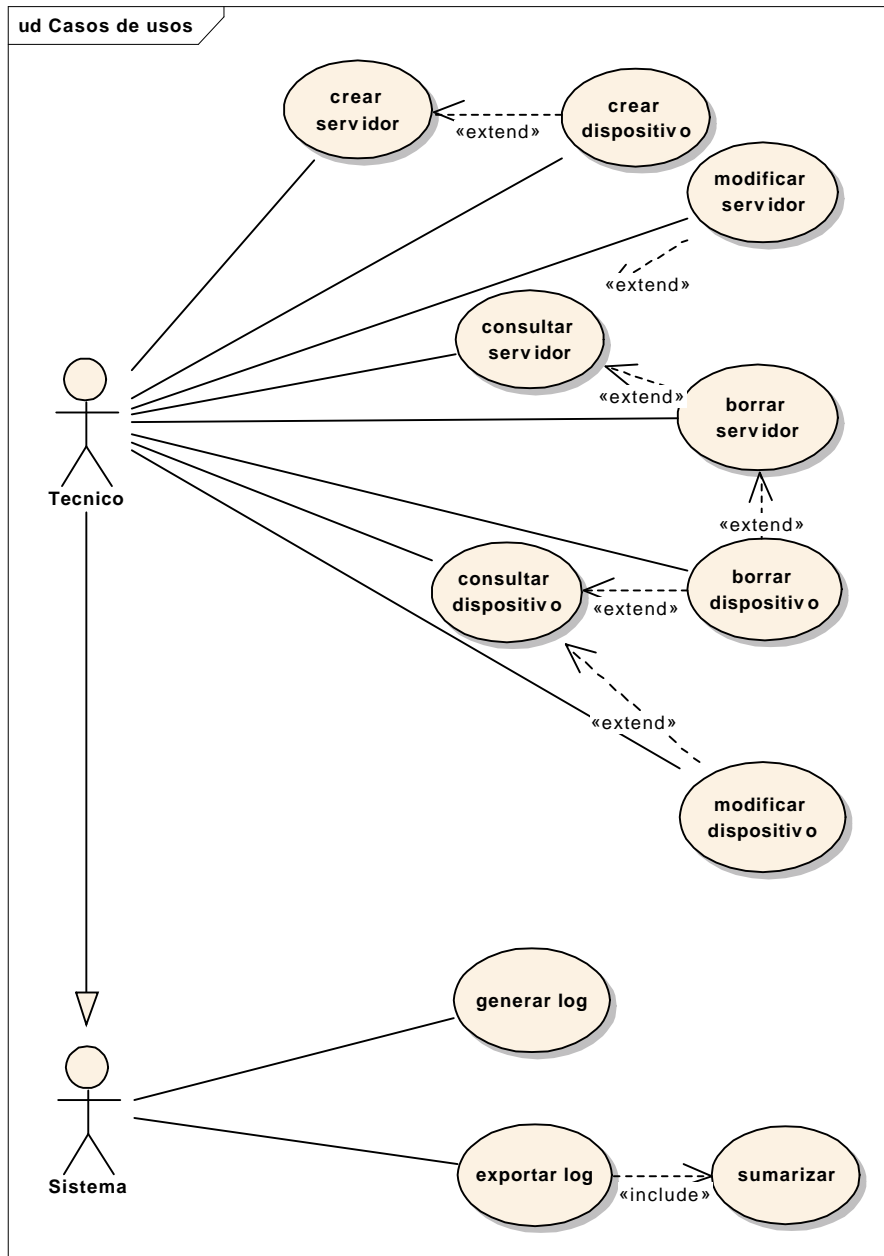


Figura 9. Casos de uso revisados.

## 2.2 Definir paquetes de análisis

Atendiendo a los datos recopilados tanto en la fase de recogida de documentación y requisitos como a la revisión de los casos de uso efectuada anteriormente observamos una diferenciación clara que nos servirá para definir dos paquetes de análisis: *infraestructura* y *reporte*.

Dentro del paquete de *infraestructura* podemos definir dos paquetes de servicio *servidores* y *dispositivos*, tal como se aprecia en la figura 10.

El paquete *infraestructura* contendrá todos los casos de uso relativos a *servidores* y *dispositivos* mientras el paquete *reporte* contendrá todos los casos de uso referidos a la generación y exportación de los logs en los que se basa el reporte. Como se puede comprobar, los elementos de cada paquete están fuertemente relacionados, lo que nos asegura la coherencia de cada paquete de análisis, además entre los elementos de los paquetes *infraestructura* y *reporte* no se dan muchas relaciones (de hecho solo tenemos la relación de dependencia entre el paquete *reporte* e *infraestructura* pues los elementos de *reporte* dependen de los elementos de *infraestructura*) lo que nos garantiza que los elementos de los paquetes sean poco dependientes entre ellos.

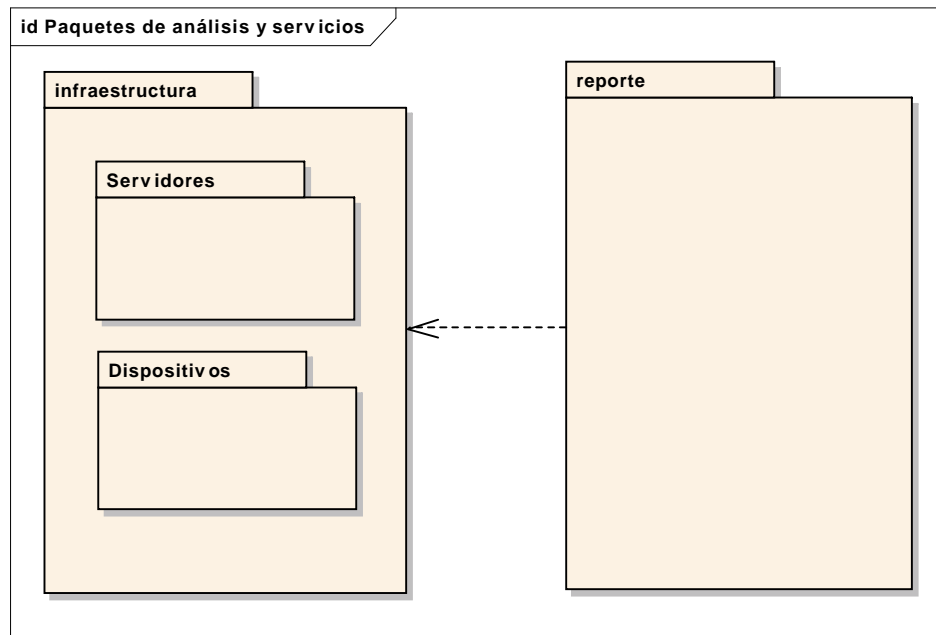


Figura 10. Paquetes de análisis y servicios.

## 2.3 Identificación de las clases de entidades

Empezaremos identificando las clases de entidades para lo que tomamos como referencia los casos de uso revisados (Figura 9). Por cada caso de uso se enumeran las clases que se encuentran. Las clases de entidades modelan entidades o acontecimientos del mundo real de los que el software debe utilizar información (atributos) la gran mayoría de estos objetos serán persistentes, es decir su información perdurara a los procesos de creación guardándola en una base de datos. (También podría usarse ficheros, pero en nuestro caso como se vera mas adelante usaremos una base de datos para cumplir con los requisitos)

<b>Caso de uso</b>	<b>clases</b>
crear servidor	servidor de impresión
crear dispositivo	dispositivo de impresión
modificar servidor	servidor de impresión
consultar servidor	servidor de impresión
borrar servidor	servidor de impresión
borrar dispositivo	dispositivo de impresión
consultar dispositivo	dispositivo de impresión
modificar dispositivo	dispositivo de impresión
generar log	log
exportar log	log
sumarizar	log

Hay clases que aunque no se mencionan en la descripción de los casos de uso, sin embargo su uso si se desprende en la fase de recogida y documentación de requisitos. Como por ejemplo *usuario*. También destacar la creación de clases con relación de herencia de la clase *dispositivo*, estas clases se verán con mas detenimiento al abordar las relaciones entre clases en el punto 2.5

En resumen, una primera lista de clases de entidades seria la siguiente: *servidor de impresión*, *dispositivo de impresión*, *log* y *usuario*.

## 2.4 Especificación de atributos

A continuación se especifican los atributos de cada una de las clases ya sean identificadas en el punto anterior o derivadas de los requerimientos de la fase de recogida y documentación de requisitos, así como el tipo de datos teniendo en cuenta que el modelo de análisis es independiente del lenguaje que se use para la implementación, estos atributos serán de tipo abstracto sin referirse a ningún lenguaje en concreto. Por este mismo motivo los nombres de atributos no están sometidos a las restricciones de ningún lenguaje de programación concreto, pero para evitar cambiar los nombres en las fases posteriores adoptaremos nombres de atributo compatibles con la programación. (Nombres formados por cifras y/o letras sin utilizar caracteres especiales como acentos y limitando el uso de underscores de forma que estos no aparezcan ni al comienzo ni al final del nombre de atributo)

<b><u>Clase</u></b>	<b><u>Atributos</u></b>	<b><u>Tipo</u></b>
servidor de impresión	Nombre	string
	Ubicación	string
dispositivo de impresión	Nombre	string
	Ubicación	string
	Coste_color	sort
	Coste_bn	sort
	Coste_copia	sort
	Coste_adquisicion	long
	fecha_adquisicion	string
numero_serie	string	
log	Type	string
	DateTime	string
	Source	string
	Category	string
	Event ID	string
	User	string
	Computer	string
	Description	string
	Data	string
LogName	string	
empleado	Numero	string
	Nombre	string
	Apellidos	string
	Centro_trabajo	string
	Responsable	string

## 2.5 Identificación de las relaciones entre clases

La figura 11 expresa las relaciones de herencia:

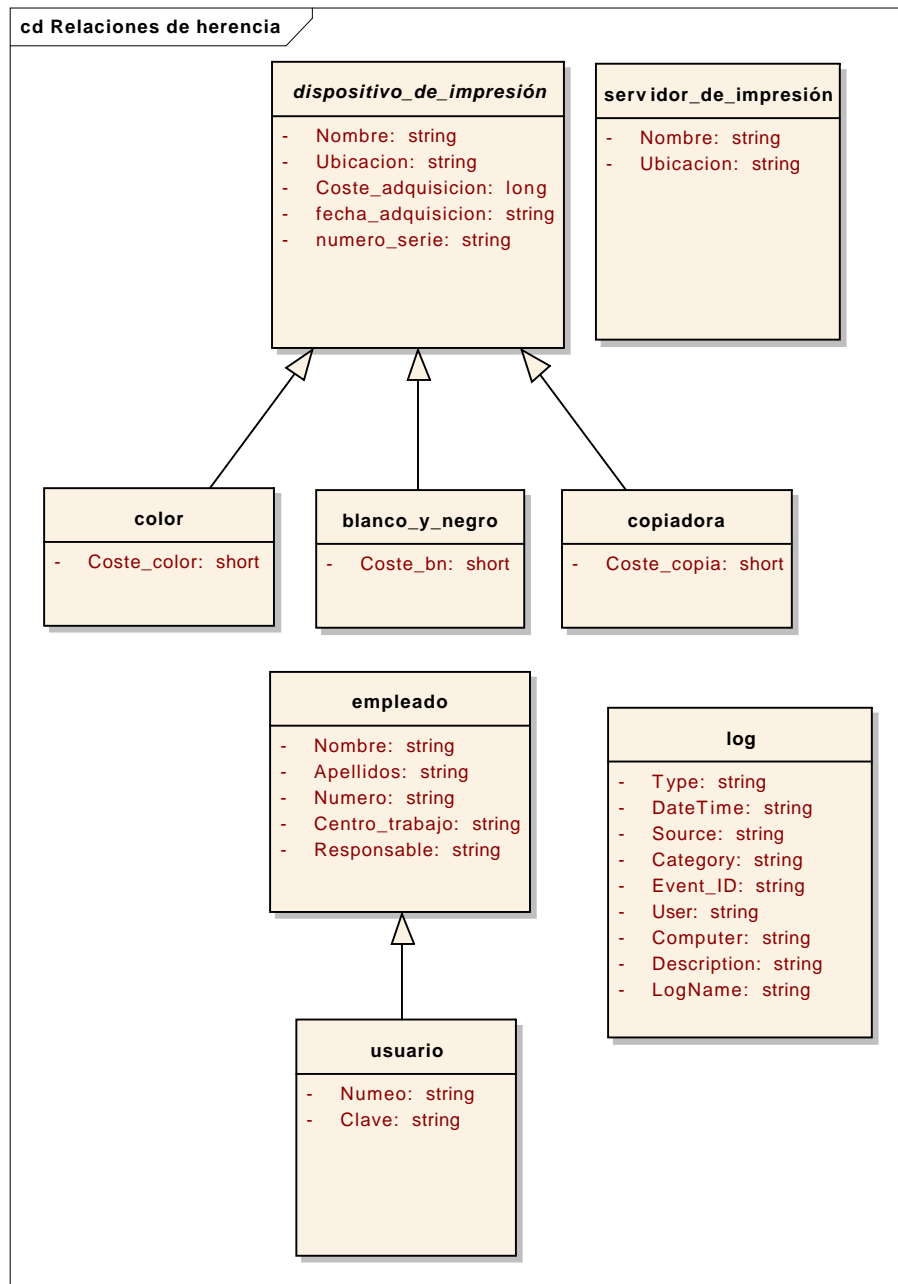


Figura 11. Relaciones de herencia.

Como se puede apreciar, las clases *color*, *blanco\_y\_negro* y *copiadora* tienen atributos propios (*Coste\_color*, *Coste\_bn* y *Coste\_copia* respectivamente) luego son definidas como subclases de *dispositivo\_de\_impresion*.

Así mismo se añade la clase *usuario* como usuario de la aplicación final pues es requerida su identificación para poder acceder a dicha aplicación, de ahí la inclusión del atributo *Clave*.

A continuación podemos apreciar las asociaciones entre las clases mediante la figura 12:

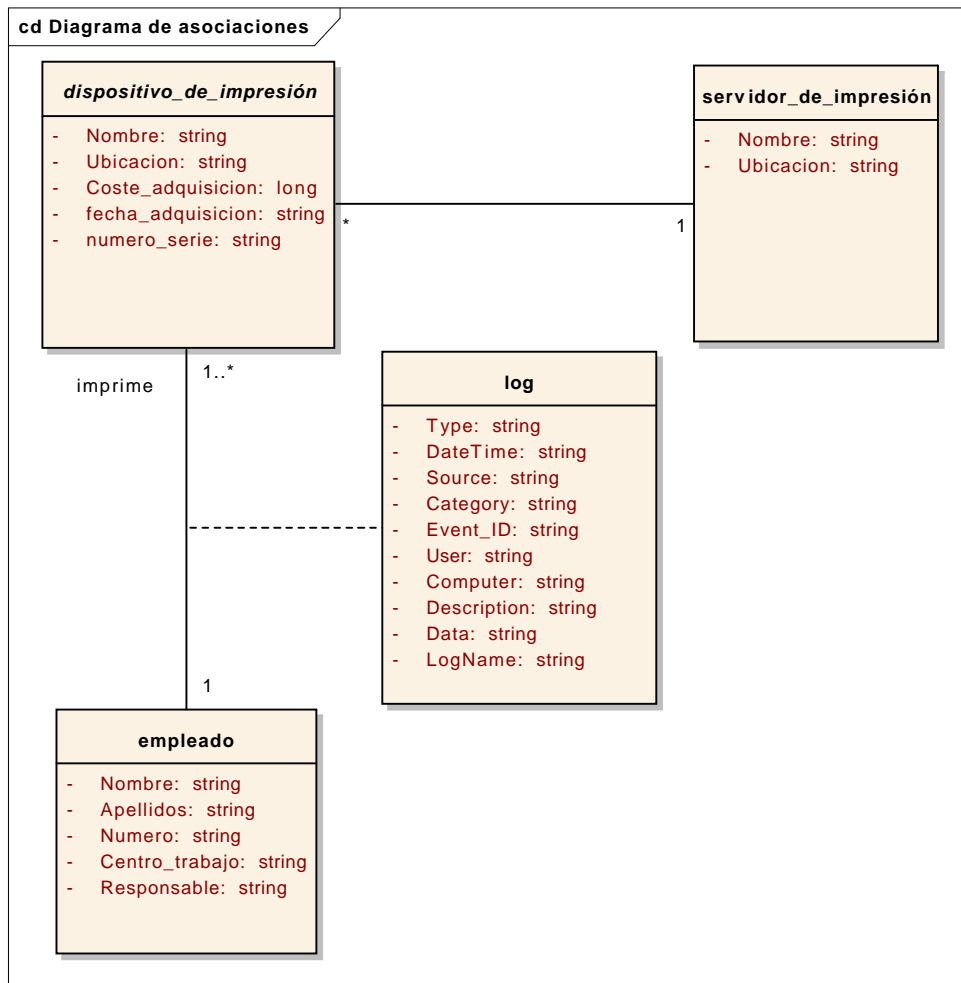


Figura 12. Diagrama de asociaciones.

En primer lugar destacar que la asociación log se establece con la superclase Dispositivo y no con sus subclases, esto es debido a que es común a todas ellas. De igual modo podemos observar la relación con empleado y con la subclase usuario, pues usuario solo se usara para validar la aplicación, el hecho de imprimir es común a cualquier empleado.

Un dispositivo solo esta relacionado con un solo servidor de impresión y en este puede estar definido multitud de dispositivos, de ahí la relación 1..\*.



## 2.6 Identificación de las clases frontera, de control y operaciones.

Para identificar las clases frontera, de control y las operaciones asociadas nos centraremos en cada caso de uso. Para cada caso de uso construiremos un diagrama de colaboración simplificado.

Como podremos comprobar, en todos los casos de uso (exceptuando aquellos realizados por el actor Sistema) el actor seleccionara una opción a la clase frontera Menu, la cual se corresponde con el menú de la aplicación, esta clase pasara la opción a la clase de control GestorMenu la cual es responsable de llamar a la clase de control principal de cada caso de uso.

Los nombres de los mensajes en todo caso serán operaciones de las clases destinatarias, exceptuando los que van de las clases frontera al actor, en este caso no le piden ninguna operación al destino que es el actor.

### Caso de uso 1, "crear servidor"

Este caso de uso "crear servidor" extiende al caso de uso numero "crear dispositivo" cuando una vez creado un servidor se decide añadir uno o varios dispositivos al nuevo servidor. La figura 13 muestra el diagrama de colaboración simplificado.

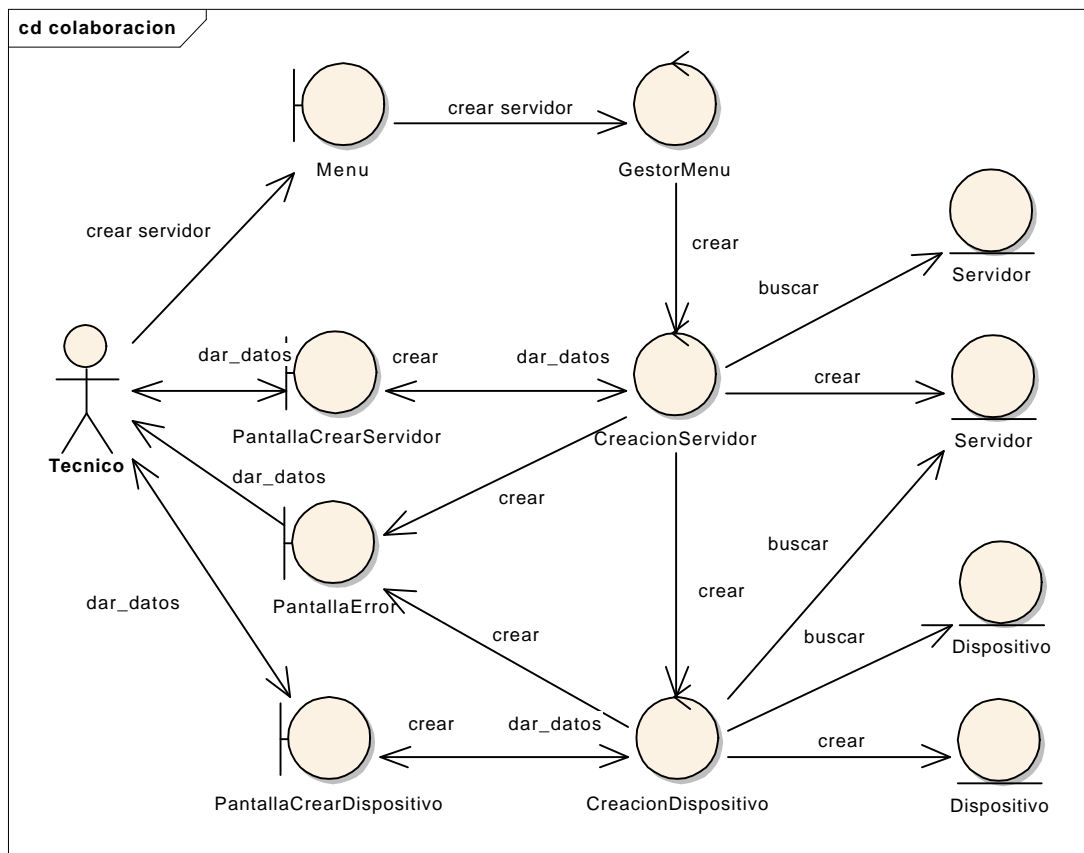


Figura 13. Diagrama de colaboración del caso de uso "crear servidor".

## Caso de uso 2, "crear dispositivo"

La figura 14 nos muestra el diagrama de colaboración del caso de uso "crear dispositivo".

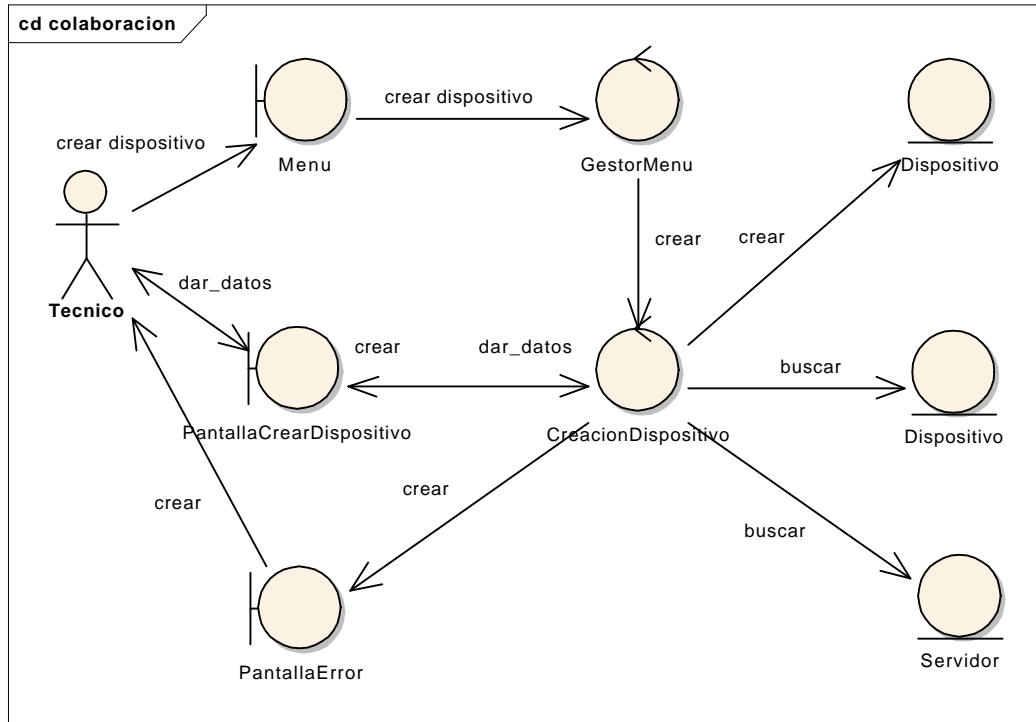


Figura 14. Diagrama de colaboración del caso de uso "crear dispositivo".

### Caso de uso 3, "modificar servidor"

En este caso de uso y tal y como se muestra en la figura 15, se busca el servidor que el actor Técnico desea modificar, y en caso de encontrarse el servidor se generara un mensaje de error que llegara al actor, en caso contrario, se modificara el servidor con los datos introducidos por el actor.

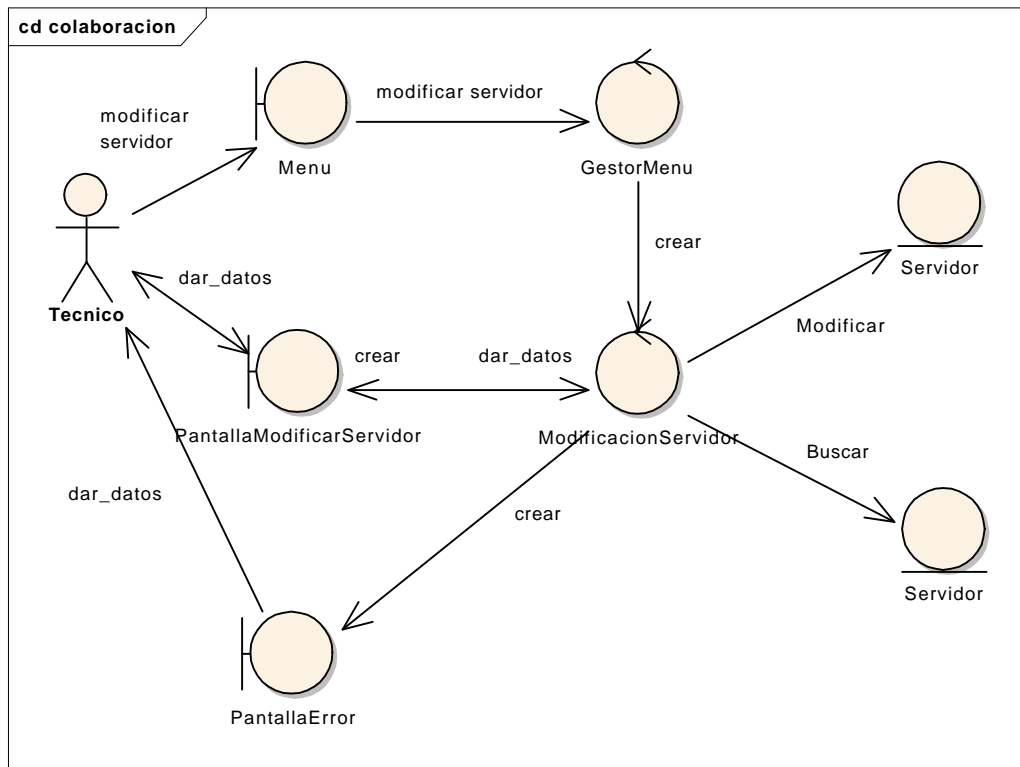


Figura 15. Diagrama de colaboración del caso de uso "modificar servidor".

#### Caso de uso 4, "consultar servidor"

Si no se encuentra el servidor, se envía un mensaje de error al actor. En caso contrario se muestran los datos por pantalla (PantallaConsultarServidor) y en adición se pueden imprimir mediante la clase de control impresión.

Existe la opción de una vez recuperados los datos del servidor proceder a su modificación, de ahí que se extienda el caso de uso "modificar servidor".

Como se puede apreciar en la figura 16 la clase de control principal, no solo extiende el caso de uso "modificar servidor" tal como se ha mencionado, si no también el caso de uso "borrar servidor". En dicha figura solo se ha representado la clase de control principal de este caso (EliminacionServidor), pudiendo verse el diagrama de colaboración completo en la descripción del propio caso de uso (Figura 17).

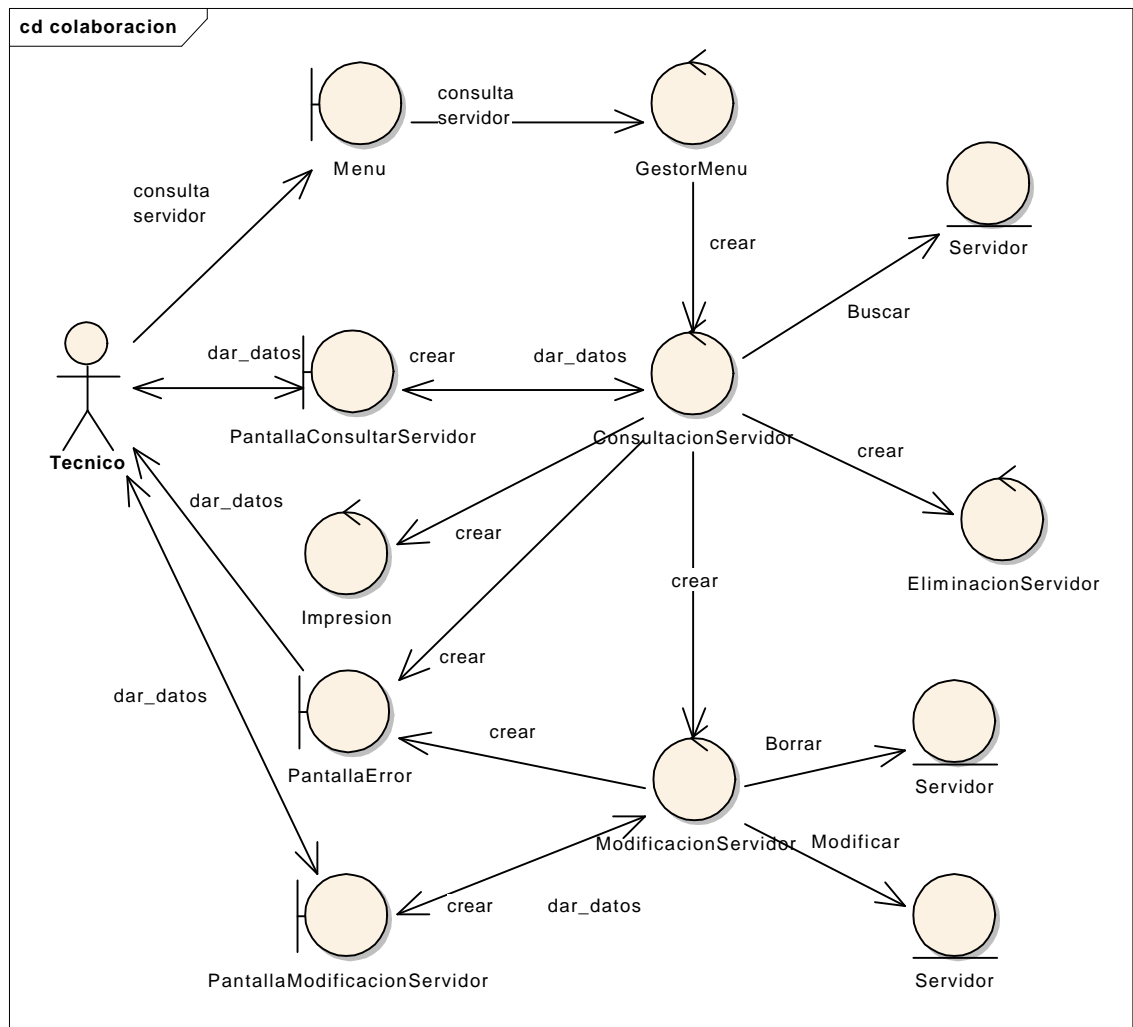


Figura 16. Diagrama de colaboración del caso de uso "consultar servidor".

Caso de uso 5, "borrar servidor".

El caso de uso borrar servidor busca los datos del servidor que se desea buscar, y en caso de no encontrar dicho servidor presentara un mensaje de error al actor. Si encontrase el servidor que se desea borrar comprobara si tiene asociados dispositivos, mediante el caso de uso que extiende (borrar dispositivo) en cuyo caso se procederá al borrado de los dispositivos asociados al servidor que se esta borrando.

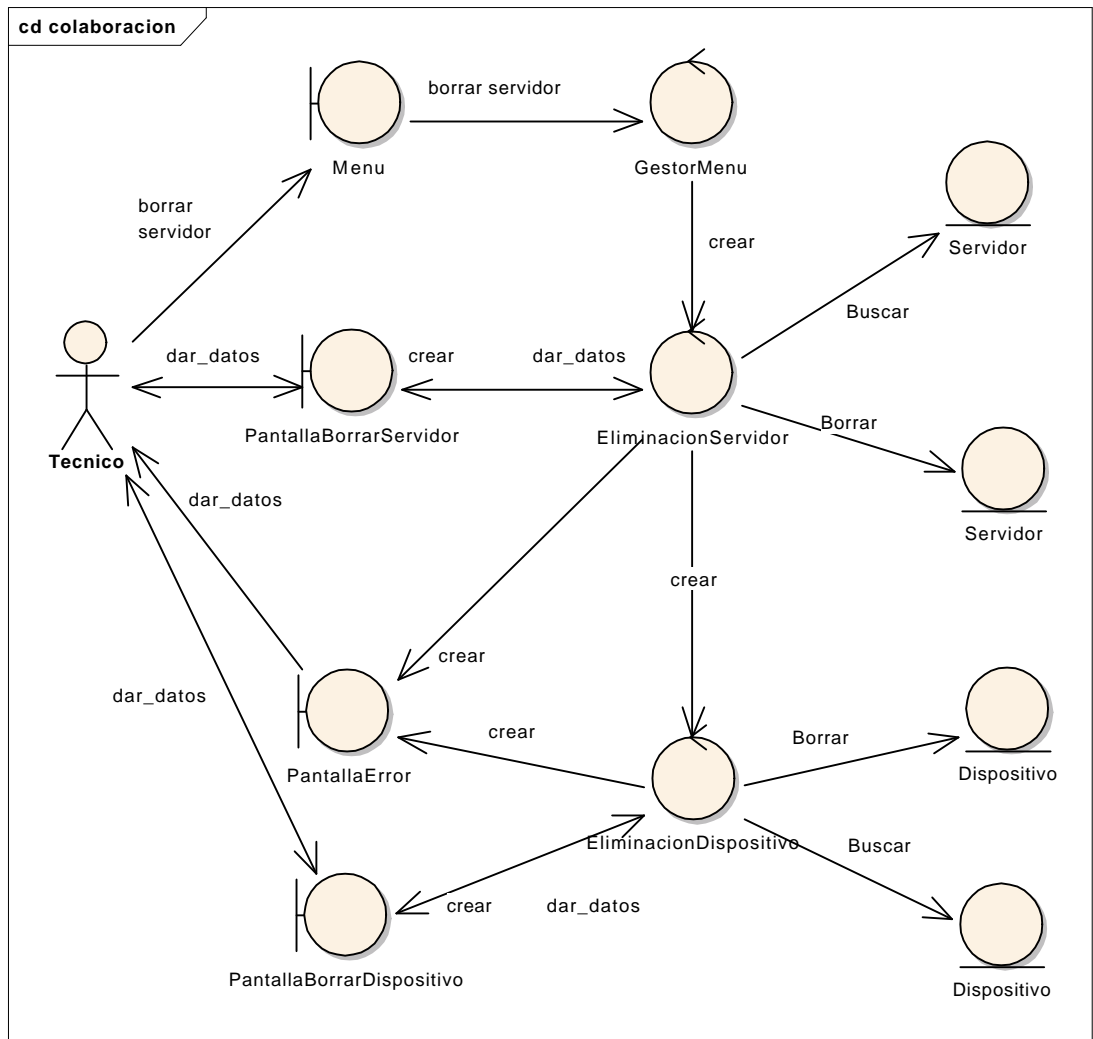


Figura 17. Diagrama de colaboración del caso de uso "borrar servidor".

## Caso de uso 6, "borrar dispositivo"

Previamente a la eliminación de un dispositivo, se ha de buscar, en caso de no existir, se mandara un mensaje de error al actor, y si existe se borraría tal y como se puede apreciar en la figura 18..

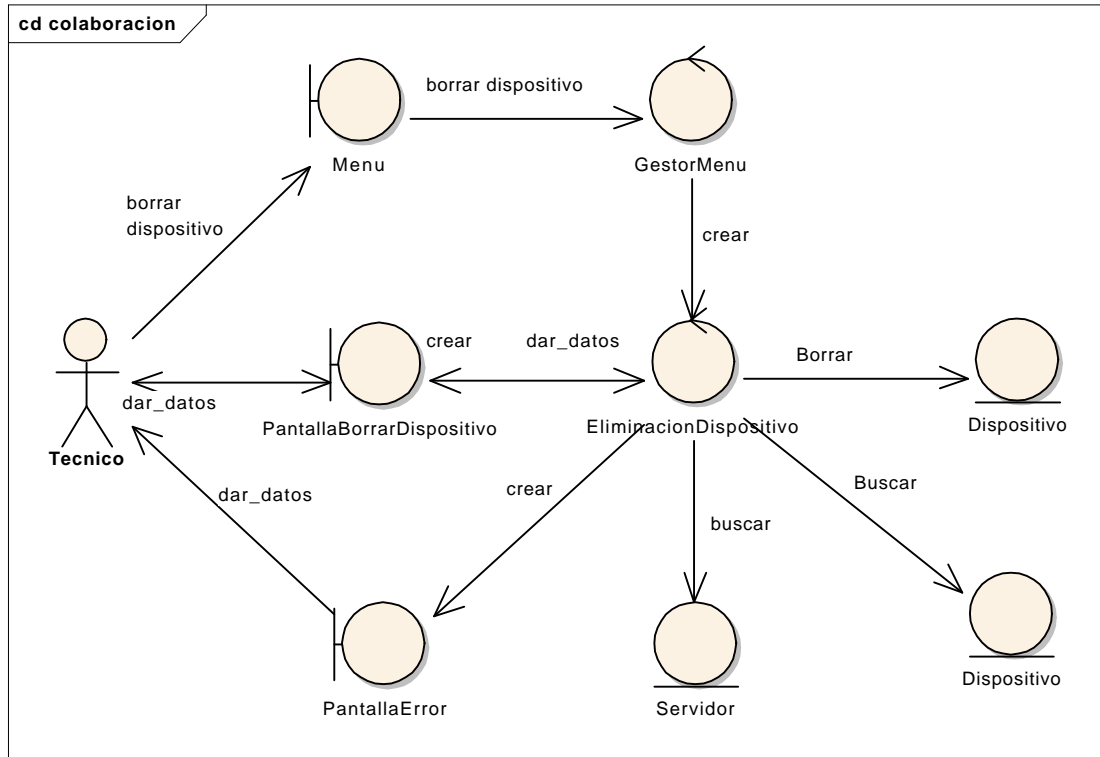


Figura 18. Diagrama de colaboración del caso de uso "borrar dispositivo".

Caso de uso 7, "consultar dispositivo"

Este caso de uso extiende a su vez a los casos de uso borrar dispositivo y modificar dispositivo, de ahí que su clase de control principal pueda llamar a la clase de control principal de los casos de uso extendidos. Esto es así pues una vez recuperados los datos del dispositivo a consultar, si no se encontrase se mandaría el pertinente mensaje de error al actor, se ofrecen las opciones de eliminar el dispositivo o bien modificarlo.

En caso de solo querer consultar el dispositivo, los datos se presentaran en pantalla, pudiendo imprimirse mediante la clase de control impresión.

Tal y como ocurría en el diagrama del caso de uso consultar servidor, no se representa en su totalidad el caso de uso borrar dispositivo, el cual puede ser consultado en la figura 18.

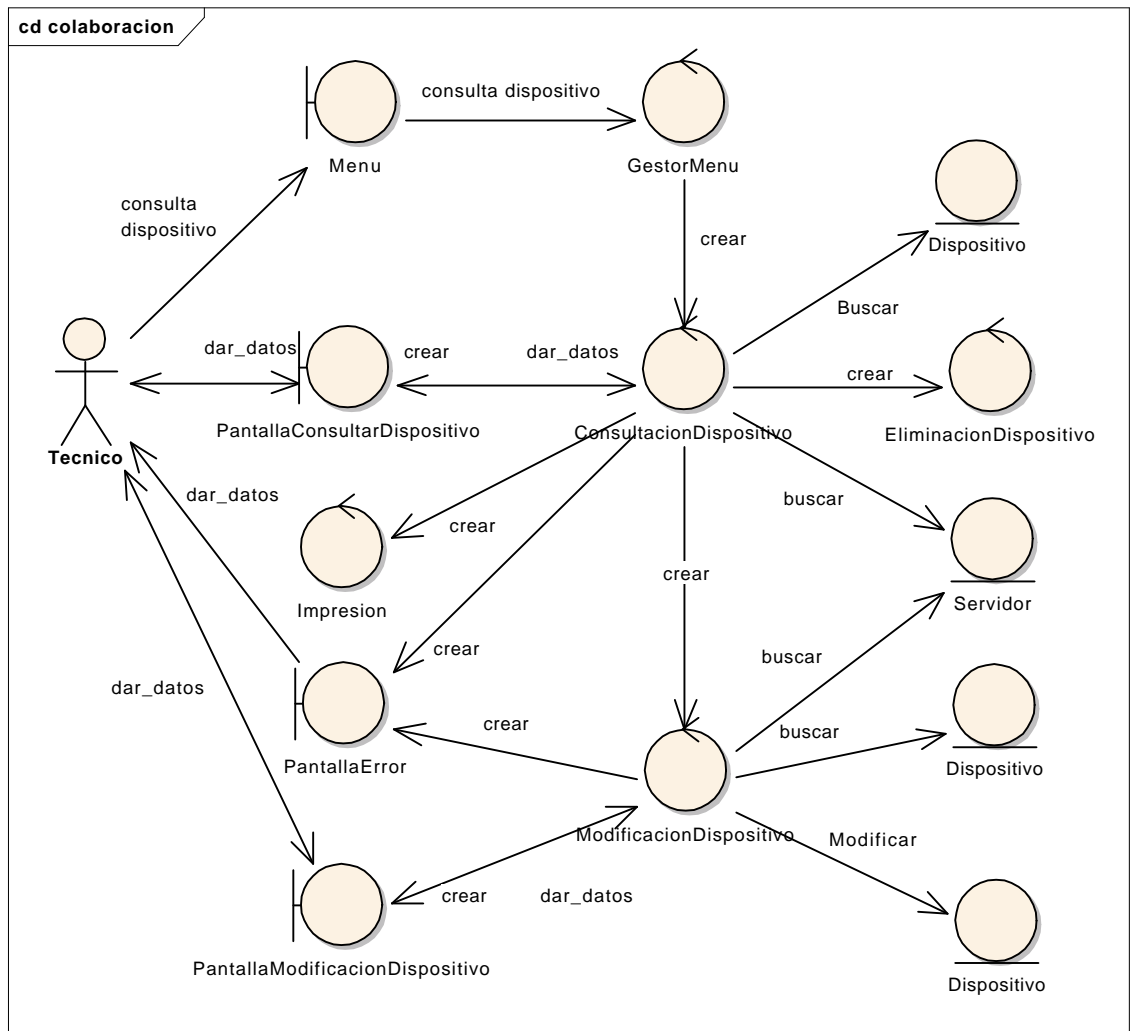


Figura 19. Diagrama de colaboración del caso de uso "consultar dispositivo".

### Caso de uso 8 "modificar dispositivo".

A la hora de modificar un dispositivo, se buscara dicho dispositivo segundoss datos aportados por el actor en la clase frontera PantallaModificarDispositivo, si no se encontrase el dispositivo, se mandaría un mensaje de error al actor.

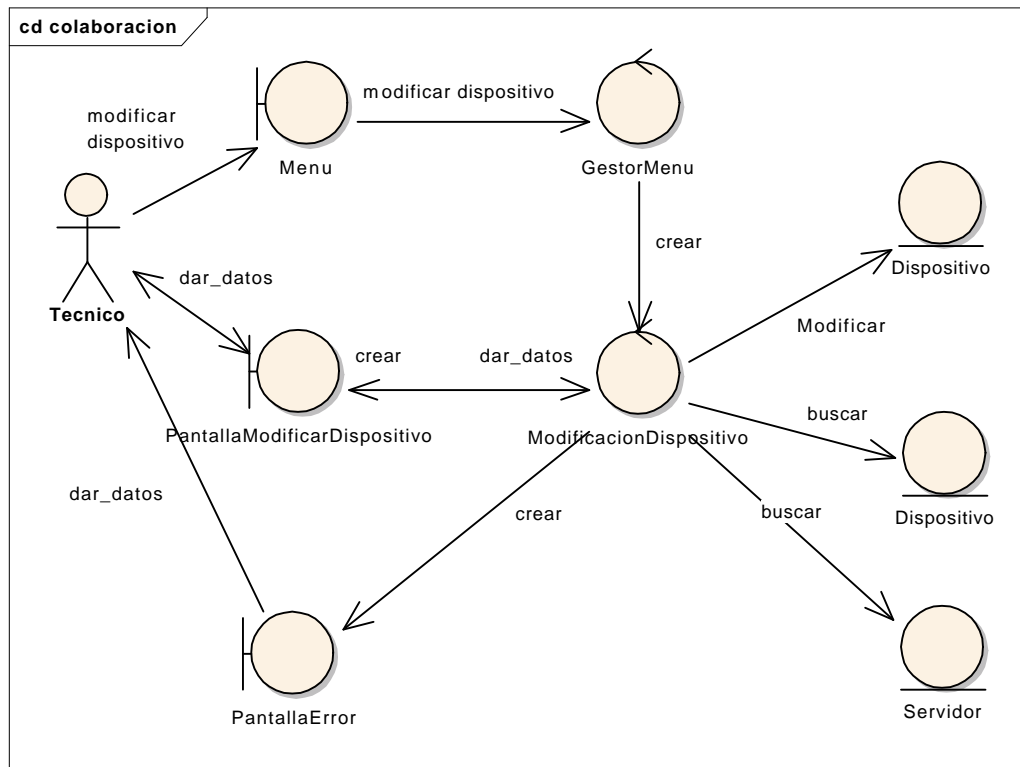


Figura 20. Diagrama de colaboración del caso de uso "modificar dispositivo".



Caso de uso 9, "generar log".

En este caso de uso, el actor principal es el propio sistema, por lo que no tenemos clases frontera. La clase de control principal, GeneracionLog consultara el log de eventos de impresión, generara el fichero de texto y borrara el log para evitar acumulaciones tal y como puede apreciarse en la figura 21.

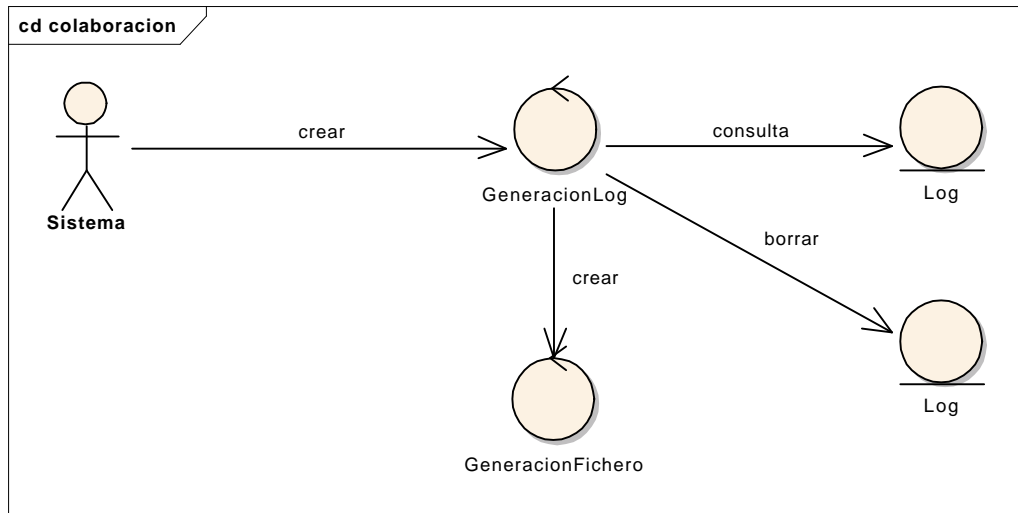


Figura 21. Diagrama de colaboración del caso de uso "generar log".

Caso de uso 10, "exportar log".

Al igual que ocurría con el caso de uso anterior, no se necesitan clases frontera. El actor sistema creara la clase de control principal ExportacionLog la cual creara clases de control para exportar el fichero de log previamente creado, extenderá al caso de uso sumarizar y borrara el fichero origen.

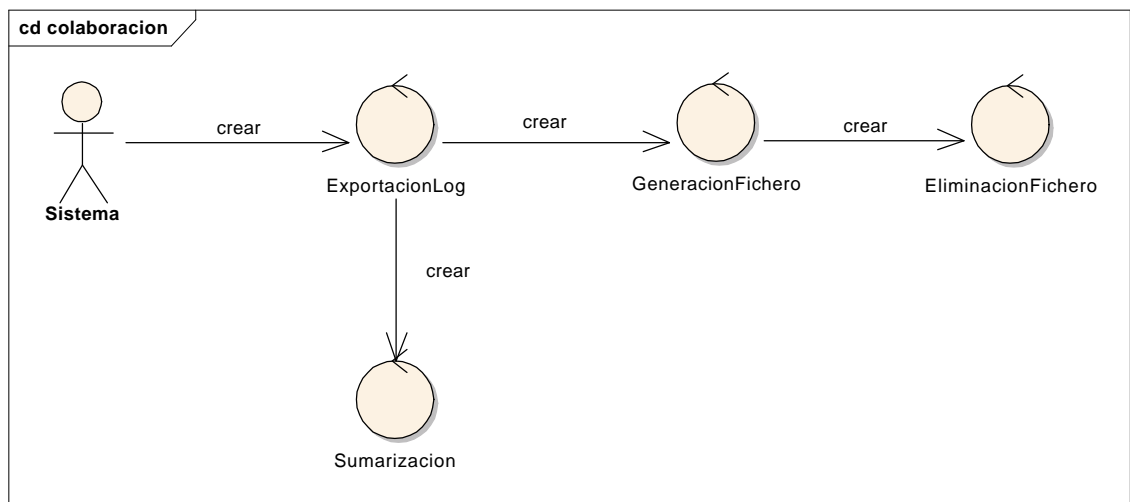


Figura 22. Diagrama de colaboración del caso de uso "exportar log".

Caso de uso 11, "sumarizar".

Este último caso de uso es muy sencillo, pues solo tiene una clase de control que se encargara de efectuar una sumarización de datos de impresión en la BD.

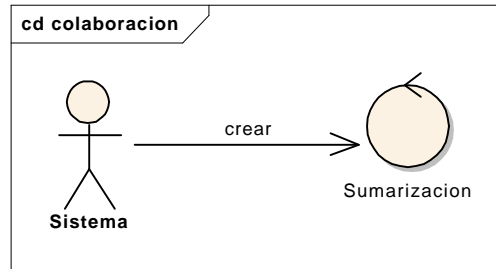


Figura 23. Diagrama de colaboración del caso de uso "sumarizar".

Adicionalmente el actor Tecnico extiende al actor sistema, esto supone que el actor Tecnico también puede ejecutar los casos de uso del actor sistema, (es decir, podrá mediante el menú efectuar las operaciones del sistema) pero en ese caso se añadirán a los casos de uso del actor sistema la clase frontera Menu y la clase de control GestorMenu, la cual llamara a la clase de control principal, tal y como esta indicada en los diagramas de colaboración números 20, 21 y 22. Así mismo se añadirá la clase frontera pantalla para presentar los datos al actor Tecnico, este extremo no era necesario para el actor Sistema, de ahí que se prescindiera de estas clases.

Para especificar este escenario tenemos los siguientes diagramas de colaboración: (Figuras 24 y 25)

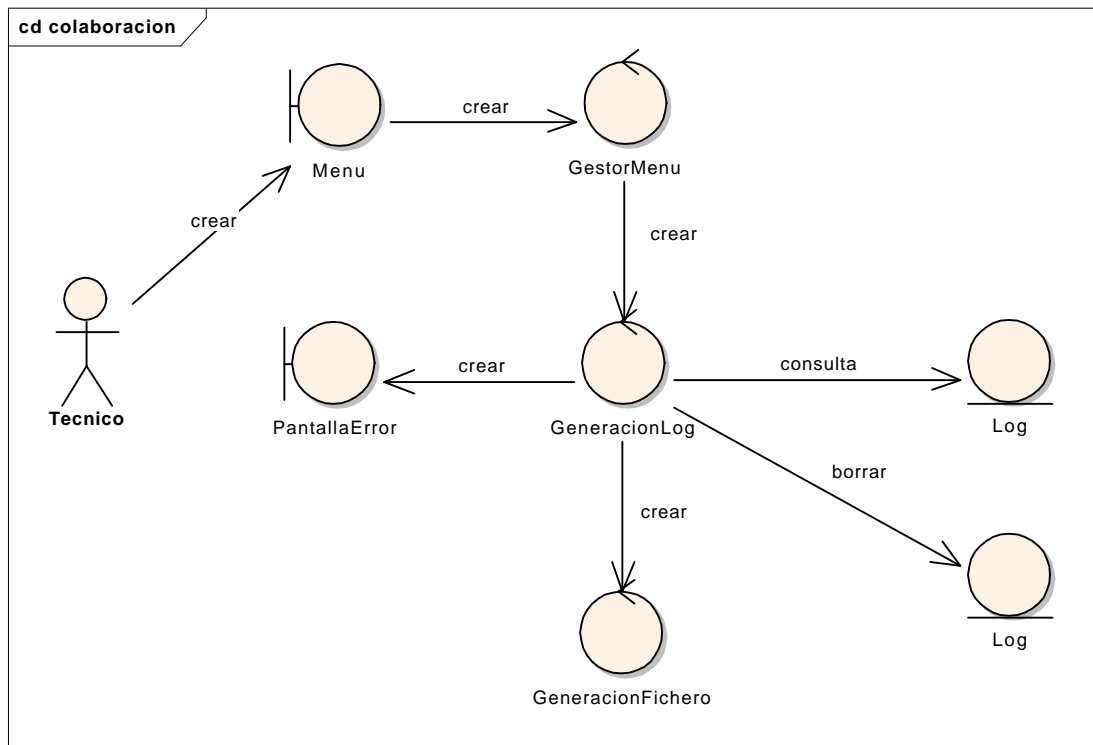


Figura 24. Diagrama de colaboración del caso de uso "generar log" para el actor Tecnico.

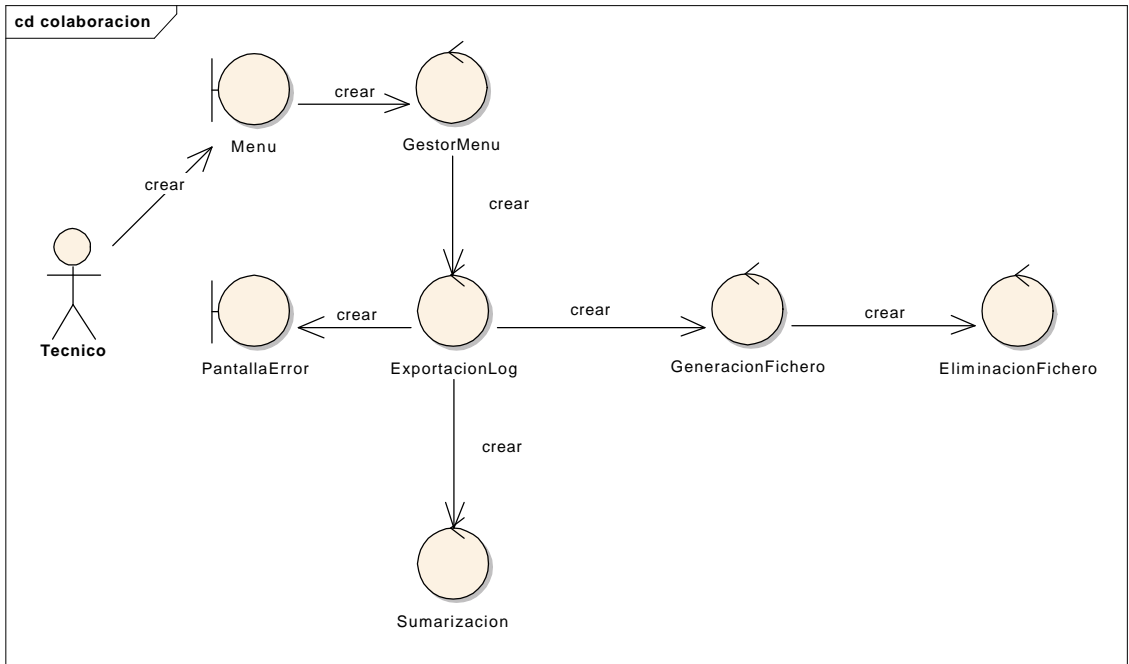


Figura 25. Diagrama de colaboración del caso de uso "exportar log" para el actor Tecnico.

## 2.7 Especificación formal de los casos de uso

Aunque pueda parecer que con los diagramas de colaboración simplificados que hemos presentado en el punto 2.6 tenemos una especificación completa de los casos de uso, esto no es correcto. Si nos fijamos en los diagramas de colaboración podremos apreciar que hay mensajes que se envían cuando se cumple determinada condición (como por ejemplo al realizar una búsqueda y no obtener datos, enviamos un mensaje de error) y también se puede apreciar que de una misma clase salen distintos mensajes sin que quede claro ni la evaluación de condiciones ni el orden de entrada y salida de mensajes en una misma clase. Es decir, en estos casos de uso no se han tenido en cuenta ni los flujos principales ni las posibles alternativas, ni las condiciones para que se realice uno u otro.

Para paliar esta situación, necesitamos realizar una especificación más detallada de los casos de uso, utilizaremos diagramas de secuencia puesto que nos ofrecen una visión que si bien es complementaria a la aportada por los diagramas de colaboración nos resolverá las indeterminaciones antes comentadas.

En definitiva nos estamos centrando en el "QUE" de los casos de uso y no en el "COMO".

Diagrama de secuencia del caso de uso "crear servidor" (Figura 26)

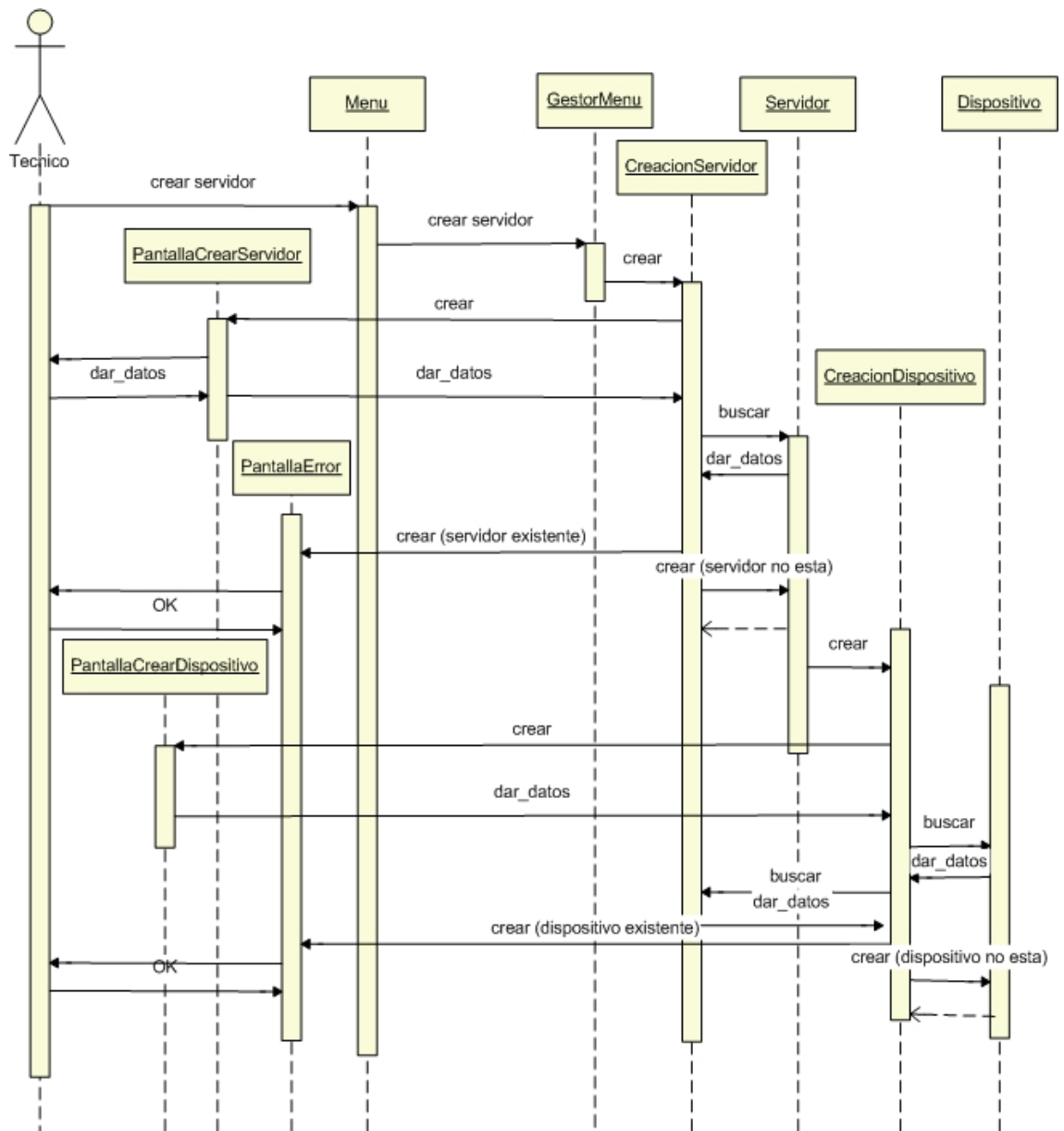


Figura 26. Diagrama de secuencias del caso de uso "crear servidor".

Diagrama de secuencia del caso de uso "crear dispositivo" (Figura 27)

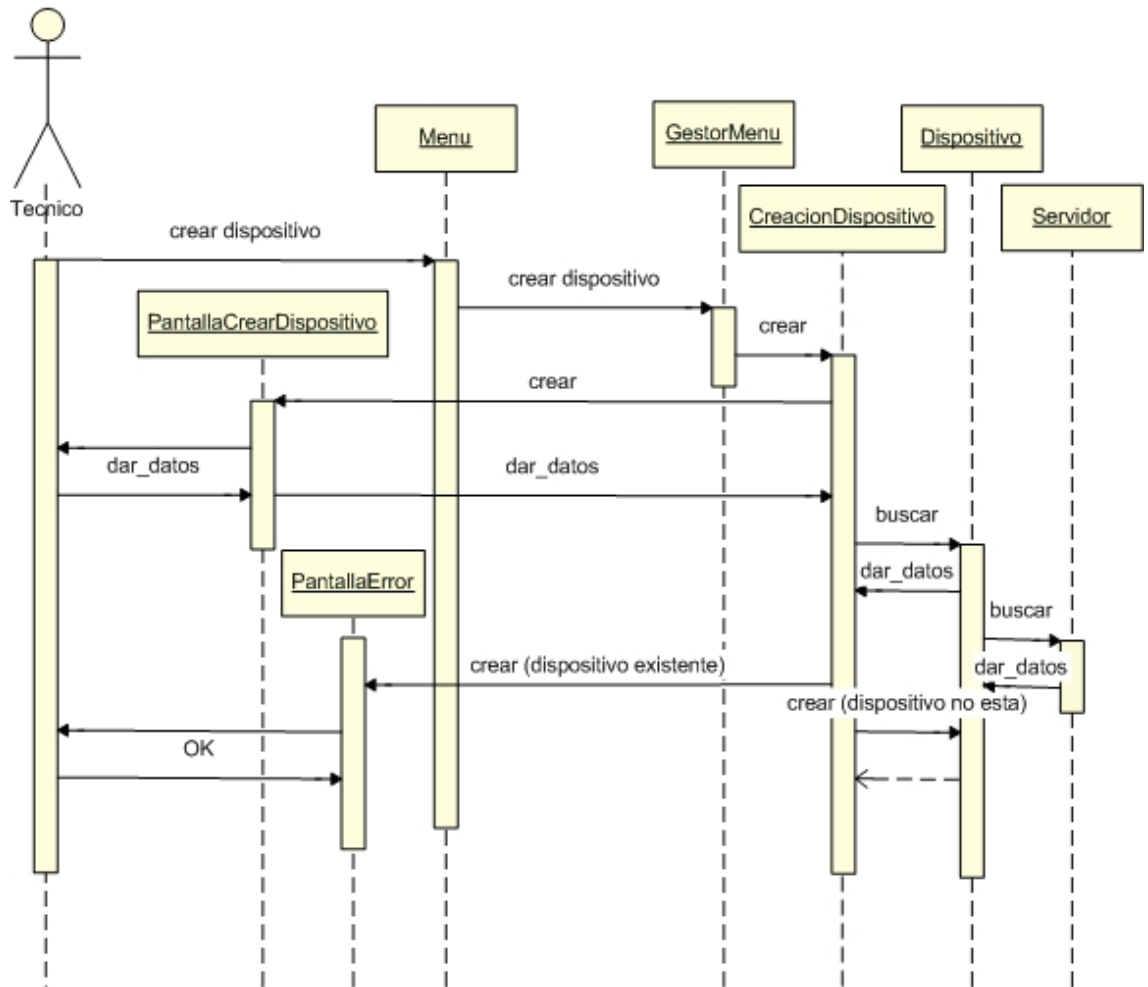


Figura 27. Diagrama de secuencias del caso de uso "crear dispositivo".

Diagrama de secuencia del caso de uso "modificar servidor" (Figura 28)

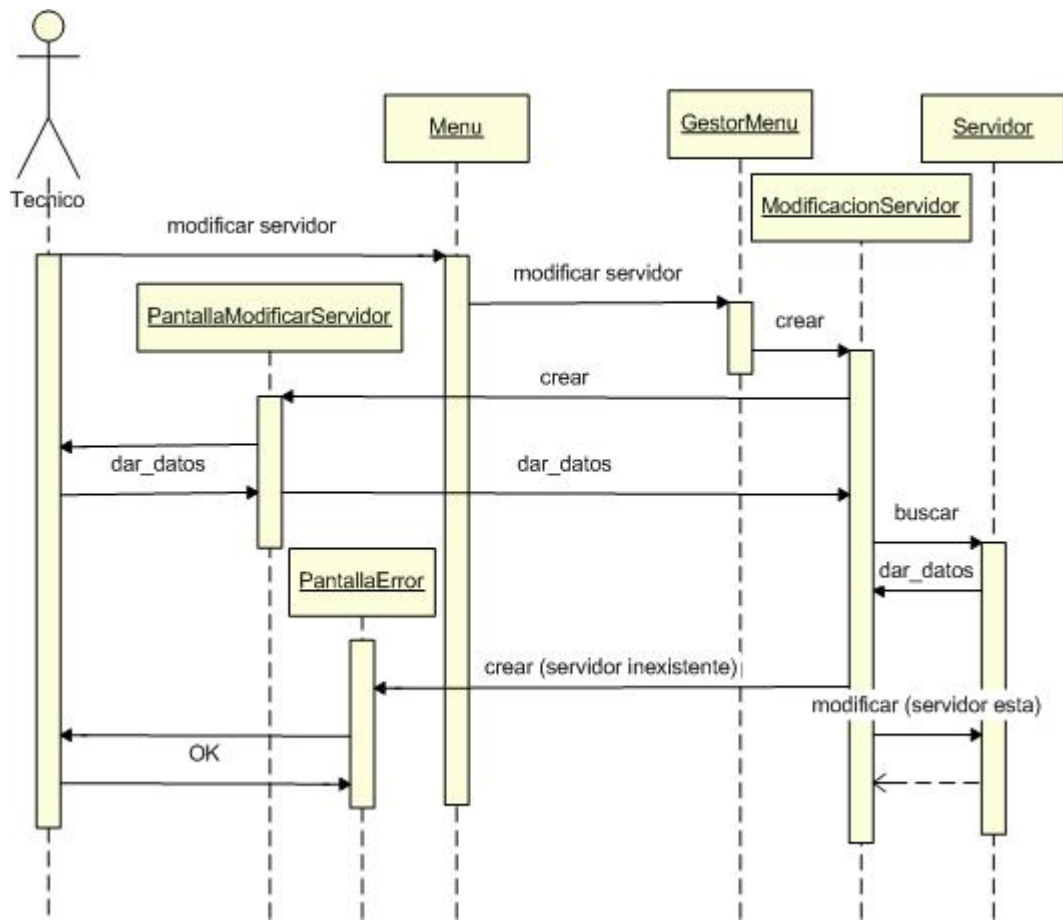


Figura 28. Diagrama de secuencias del caso de uso "modificar servidor".

Diagrama de secuencia del caso de uso "consultar servidor" (Figura 29)

Como se puede observar en la figura 29, en el diagrama de secuencia solo se expresa el caso de uso sin incluir aquellos casos de uso que extiende al considerar que no es necesario al disponer de sus propios diagramas. En este caso extiende a los casos de uso modificar servidor y borrar servidor, cuyos diagramas de secuencia se pueden consultar en las figuras 28 y 30 respectivamente.

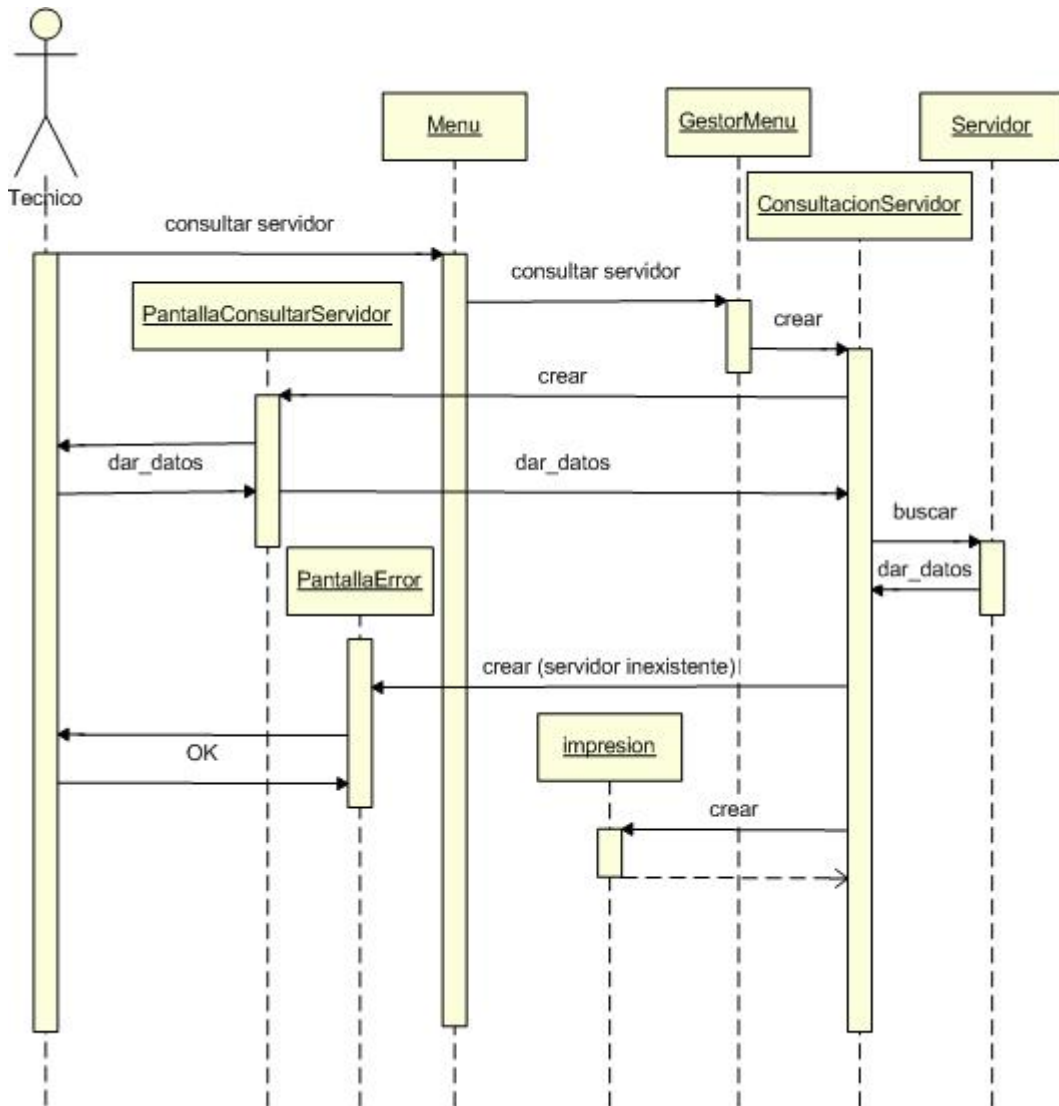


Figura 29. Diagrama de secuencias del caso de uso "consultar servidor".



Diagrama de secuencia del caso de uso "borrar servidor" (Figura 30)

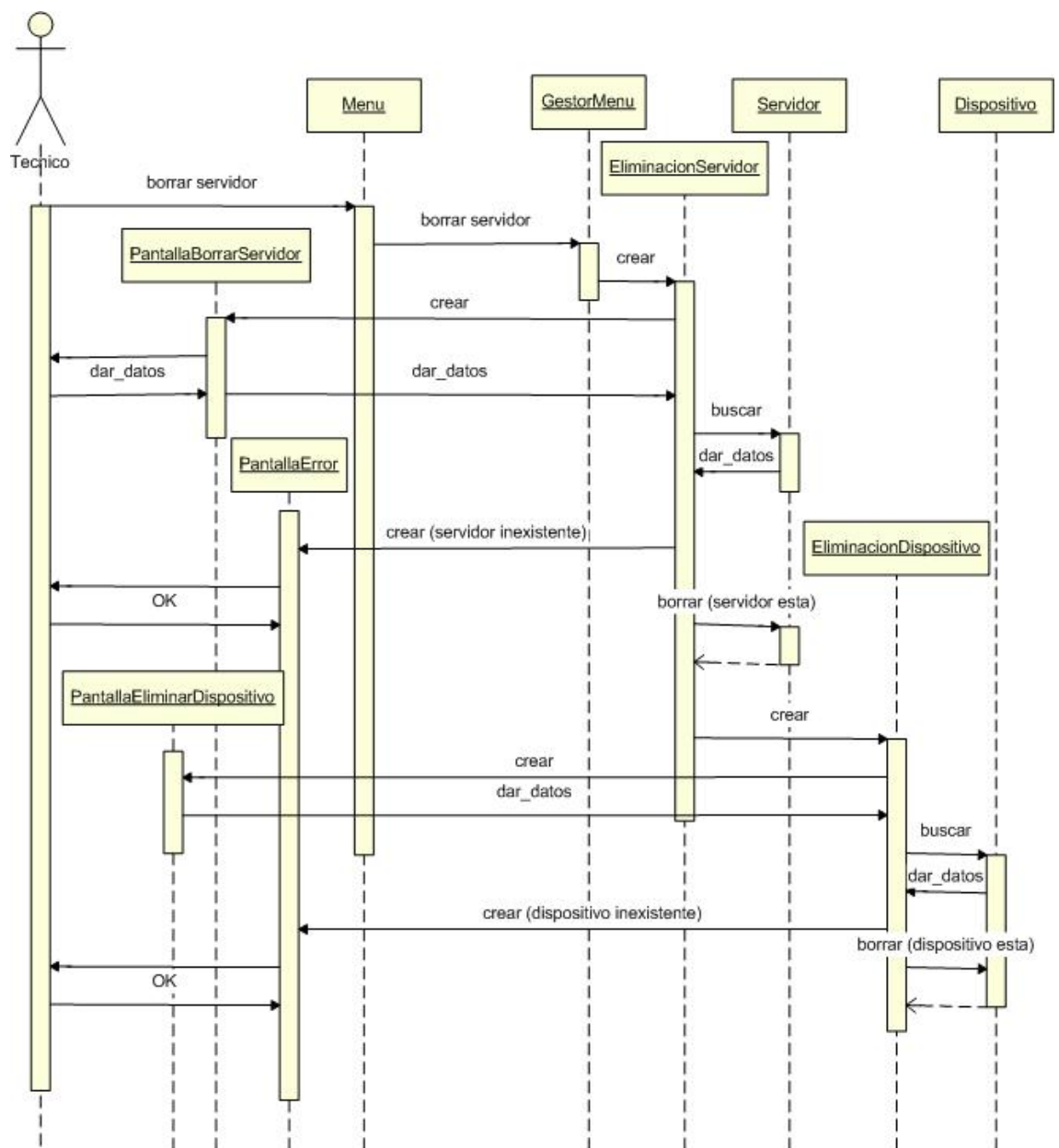


Figura 30. Diagrama de secuencias del caso de uso "borrar servidor".

Diagrama de secuencia del caso de uso "borrar dispositivo" (Figura 31)

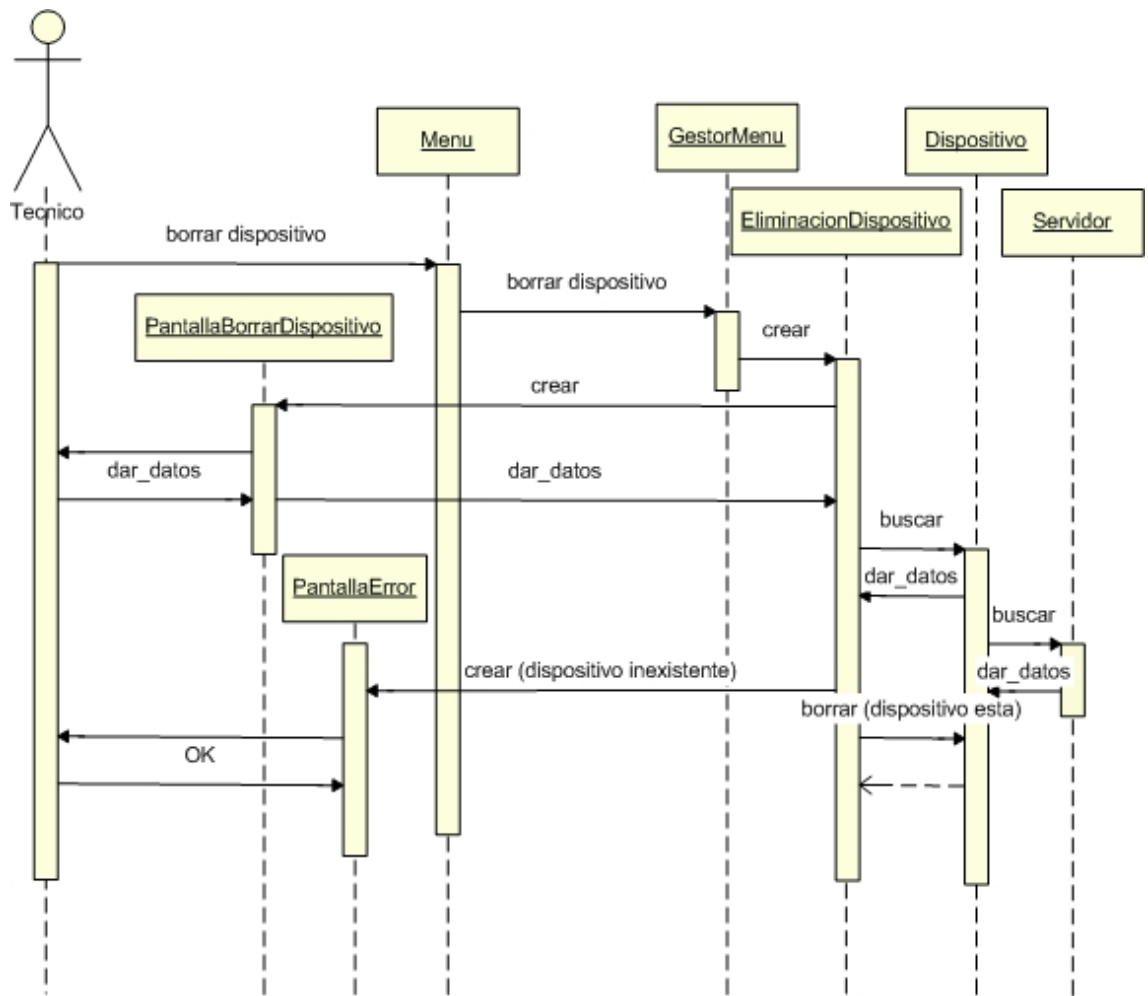


Figura 31. Diagrama de secuencias del caso de uso "borrar dispositivo".

Diagrama de secuencia del caso de uso "consultar dispositivo" (Figura 32)

Como se puede observar en la figura 32, en el diagrama de secuencia solo se expresa el caso de uso sin incluir aquellos casos de uso que extiende al considerar que no es necesario al disponer de sus propios diagramas. En este caso extiende a los casos de uso modificar dispositivo y borrar dispositivo, cuyos diagramas de secuencia se pueden consultar en las figuras 33 y 31 respectivamente.

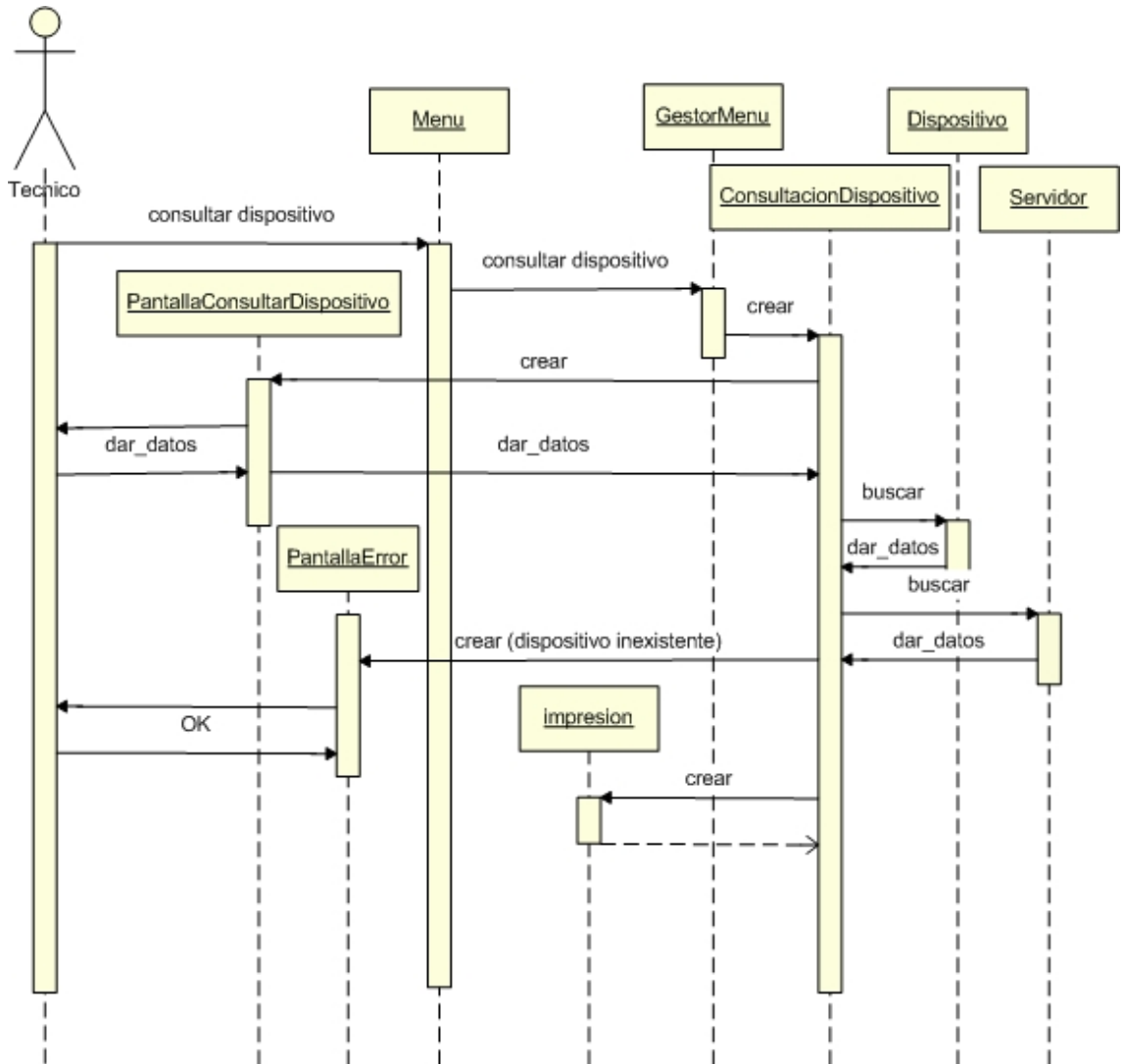


Figura 32. Diagrama de secuencias del caso de uso "consultar dispositivo".

Diagrama de secuencia del caso de uso "modificar dispositivo" (Figura 33)

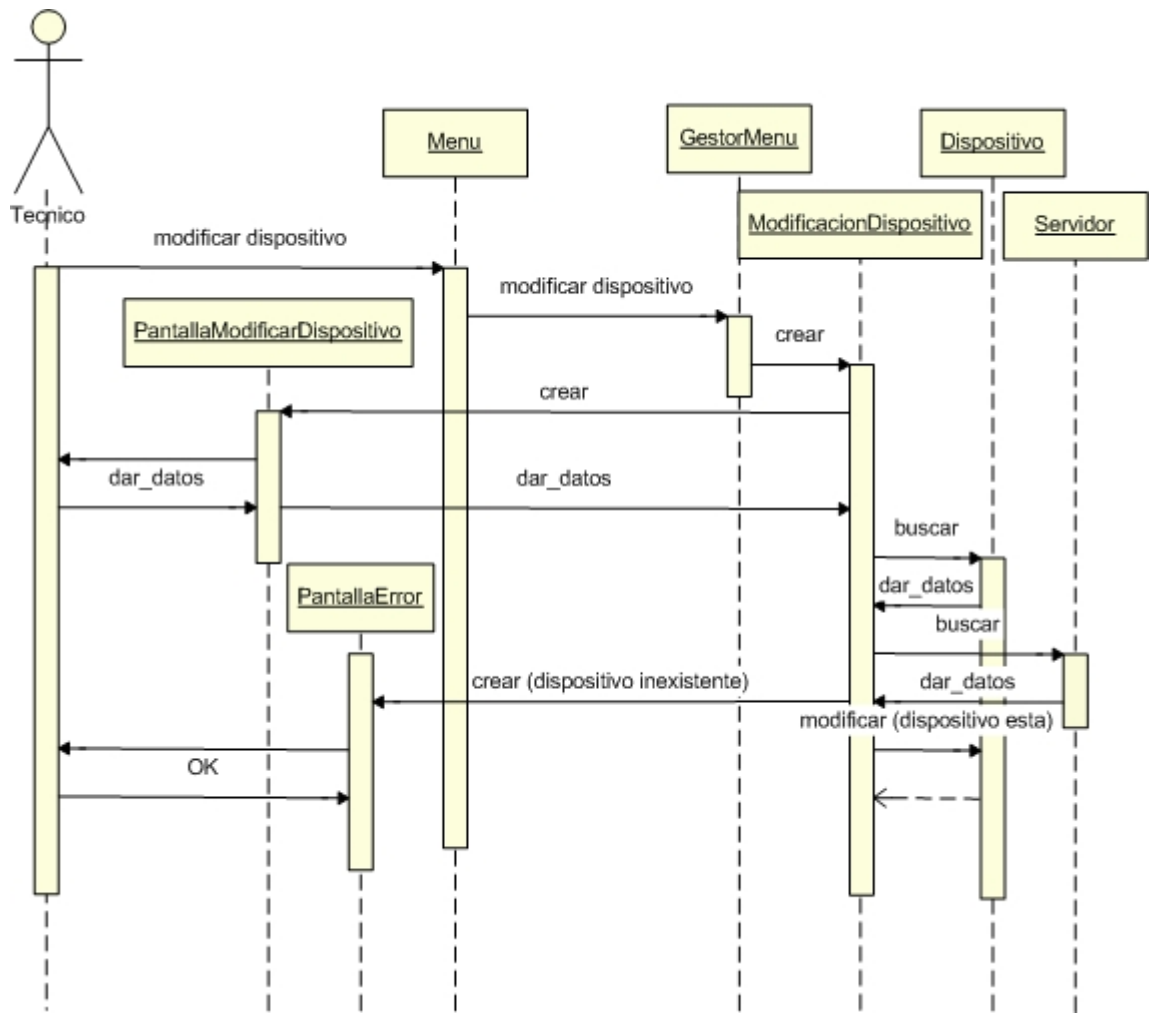


Figura 33. Diagrama de secuencias del caso de uso "modificar dispositivo".

Diagrama de secuencia del caso de uso "generar log" (Figura 34)

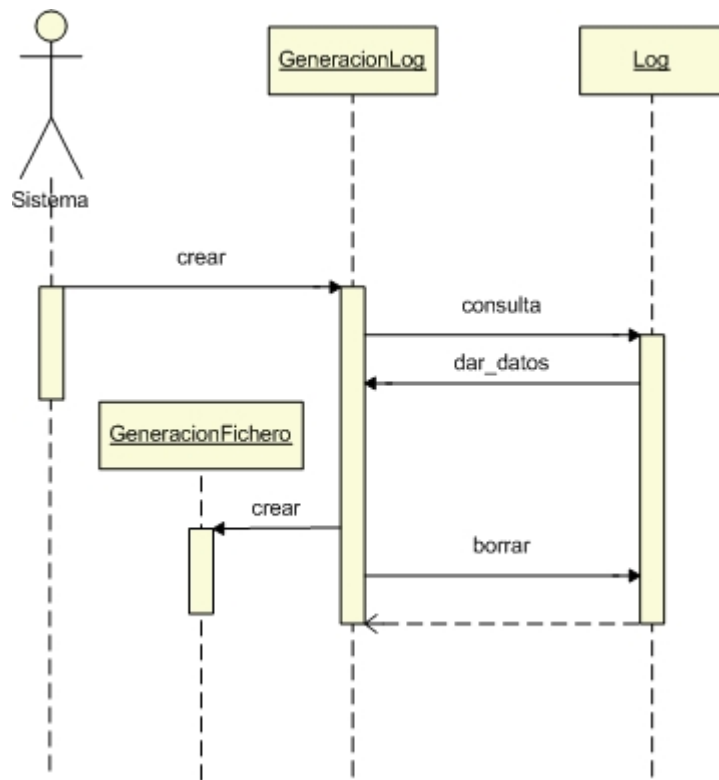


Figura 34. Diagrama de secuencias del caso de uso "generar log".

Diagrama de secuencia del caso de uso "exportar log" (Figura 34)

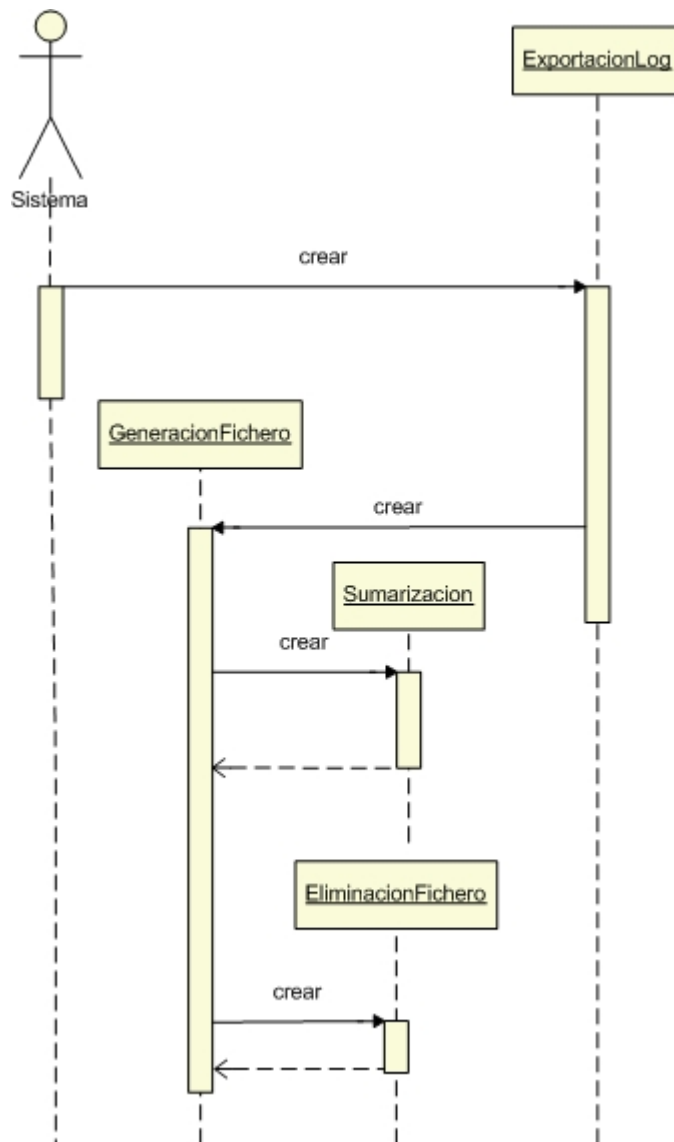


Figura 34. Diagrama de secuencias del caso de uso "exportar log".

Diagrama de secuencia del caso de uso "sumarizar" (Figura 35)

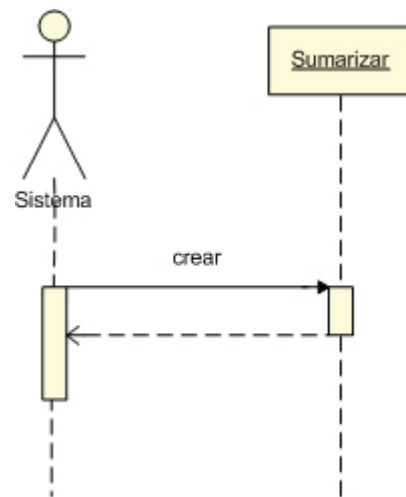


Figura 35. Diagrama de secuencias del caso de uso "sumarizar".

## 2.8 Análisis de la interfaz de usuario

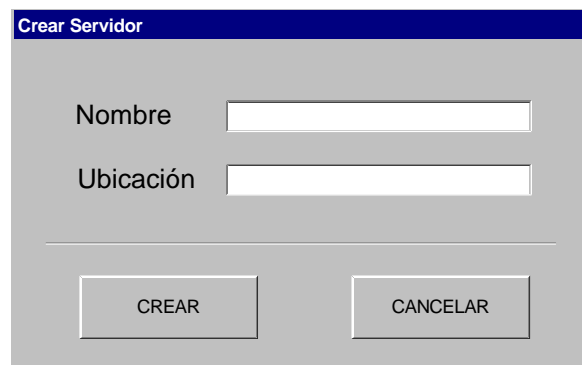
Llegado el momento de analizar la interfaz de usuario, es esencial prestar atención en este paso pues se trata de la presentación que tendrá el sistema a ojos del usuario.

Nuestro objetivo es realizar un esquema de cada ventana asociada a una clase frontera. Queda fuera de alcance aquellas ventanas denominadas secundarias cuyo fin es presentar mensajes genéricos, solicitar confirmaciones al usuario o bien la introducción de un valor o parámetro.

Para poder llevar esta fase de análisis a buen puerto es fundamental contar con la colaboración de los usuarios finales, de forma que el contenido de las ventanas, así como su disposición sean de su agrado, evitando de esta forma el rechazo del sistema.

A continuación se muestran los esquemas correspondientes a cada clase frontera:

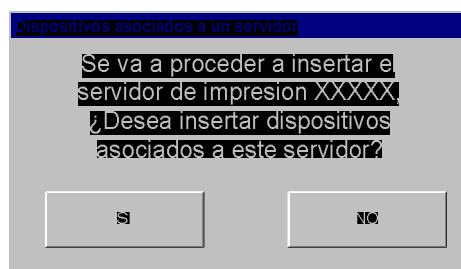
- Clase PantallaCrearServidor



El diagrama muestra una ventana de diálogo con el título "Crear Servidor". Dentro de la ventana, hay dos campos de texto etiquetados "Nombre" y "Ubicación". Debajo de los campos, hay dos botones: "CREAR" y "CANCELAR".

Figura 36. Esquema clase frontera "PantallaCrearServidor".

Al crear un servidor se puede solicitar al actor si desea crear dispositivos asociados a dicho servidor (extensión de crear dispositivo), para lo cual se le presentara la siguiente pantalla:



El diagrama muestra una ventana de diálogo con el título "Dispositivos asociados a un servidor". El texto principal dice: "Se va a proceder a insertar e servidor de impresion XXXXX ¿Desea insertar dispositivos asociados a este servidor?". Hay dos botones: "S" (Sí) y "N" (No).



- Clase PantallaCrearDispositivo

Crear Dispositivo

Nombre

Ubicación

Nombre servidor

---

Costes

Pag. color  Pag. ByN

Pag. copiada  Adquisicion

---

Fecha adquisicion

Numero de serie

---

CREAR CANCELAR

Figura 37. Esquema clase frontera "PantallaCrearDispositivo".

Se introducirán los atributos del dispositivo, además de un servidor al que se le asociara, y se pulsara el botón CREAR para crear el dispositivo.

- Clase PantallaModificarServidor

Modificar Servidor

Nombre  Buscar

Ubicación

---

MODIFICAR CANCELAR

Figura 38. Esquema clase frontera "PantallaModificarServidor".

Como se puede apreciar en la figura 38, se realizara la búsqueda mediante el atributo nombre, una vez seleccionado un servidor en concreto (datos presentados en la pantalla) se podrá modificar y salvar pulsando el botón "MODIFICAR".

- Clase PantallaConsultarServidor

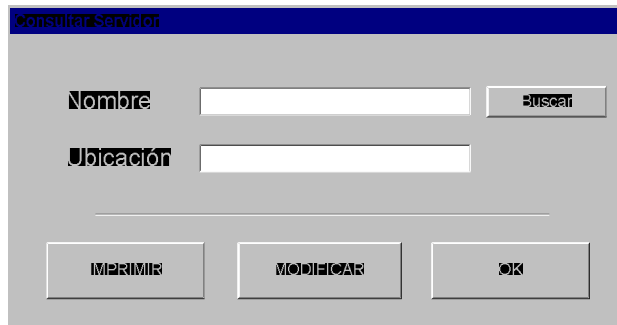


Figura 39. Esquema clase frontera "PantallaConsultarServidor".

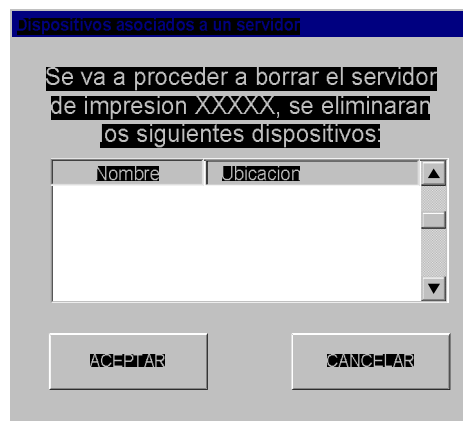
Como se puede apreciar en la figura, se puede buscar un servidor por nombre, y una vez presentados los datos, imprimirlos o bien extender el caso de uso modificar servidor mediante el botón MODIFICAR.

- Clase PantallaBorrarServidor



Figura 40. Esquema clase frontera "PantallaBorrarServidor".

En este caso, se va a proceder a borrar un servidor después de buscar dicho servidor por su nombre. Evidentemente no tiene sentido conservar dispositivos asociados a un servidor que se va a borrar, por lo tanto al pulsar el botón BORRAR, se activara una ventana para borrar los dispositivos asociados al servidor que se esta borrando. (Se extiende el caso de uso "borrar dispositivo")



- Clase PantallaBorrarDispositivo

Este diagrama muestra la interfaz de usuario para la clase 'Borrar Dispositivo'. El formulario tiene un encabezado azul con el título 'Borrar Dispositivo'. Contiene tres campos de texto para 'Nombre', 'Ubicación' y 'Nombre servidor', seguidos de un botón 'Buscar'. Una sección titulada 'Costes' incluye cuatro campos de texto: 'Pag. color', 'Pag. ByN', 'Pag. copiada' y 'Adquisicion'. Debajo de esta sección hay dos campos más: 'Fecha adquisicion' y 'Numero de serie'. En la parte inferior del formulario se encuentran dos botones: 'BORRAR' y 'CANCELAR'.

Figura 41. Esquema clase frontera "PantallaBorrarDispositivo".

- Clase PantallaConsultaDispositivo

Este diagrama muestra la interfaz de usuario para la clase 'Consultar Dispositivo'. El formulario tiene un encabezado azul con el título 'Consultar Dispositivo'. Contiene tres campos de texto para 'Nombre', 'Ubicación' y 'Nombre servidor', seguidos de un botón 'Buscar'. Una sección titulada 'Costes' incluye cuatro campos de texto: 'Pag. color', 'Pag. ByN', 'Pag. copiada' y 'Adquisicion'. Debajo de esta sección hay dos campos más: 'Fecha adquisicion' y 'Numero de serie'. En la parte inferior del formulario se encuentran tres botones: 'IMPRIMIR', 'MODIFICAR' y 'OK'.

Figura 42. Esquema clase frontera "PantallaConsultaDispositivo".

Como se puede apreciar en la figura 42, se puede buscar un dispositivo por nombre del dispositivo y nombre del servidor al que esta asociado, y una vez presentados los datos, imprimirlos o bien extender el caso de uso modificar dispositivo mediante el botón MODIFICAR.

- Clase PantallaModificarDispositivo

El formulario 'Modificar Dispositivo' tiene un título en azul oscuro. Contiene los siguientes campos de texto: 'Nombre', 'Ubicación' y 'Nombre servidor'. Debajo de estos campos hay un botón 'Buscar'. Una línea horizontal separa esta sección de la siguiente, titulada 'Costes'. En esta sección hay cuatro campos de texto: 'Pag. color', 'Pag. ByN', 'Pag. copiada' y 'Adquisicion'. Otra línea horizontal separa esto de los campos 'Fecha adquisicion' y 'Numero de serie'. Al final del formulario hay dos botones: 'MODIFICAR' y 'CANCELAR'.

Figura 43. Esquema clase frontera "PantallaModificarDispositivo".

Para modificar un dispositivo, previamente se ha de buscar este dispositivo y su servidor asociado mediante el atributo nombre (de ambos), para posteriormente presentarnos todos sus datos, en caso de encontrarse, dándonos la posibilidad de modificarlos. Pulsando el botón MODIFICAR se almacenaran los cambios, y con el botón CANCELAR se descartaran.

### 3. Diseño

En esta etapa nos centraremos en la elaboración de la solución al problema que da origen a nuestro sistema software sin entrar en la realización, la cual no es objeto del presente trabajo.

En la anterior etapa de análisis hemos descrito un modelo con las funciones que tiene que desempeñar el software así como también hemos especificado los requisitos no funcionales que ha de cumplir el software. Esta etapa, de diseño, servirá de puente entre la etapa de análisis y la de realización. Nuestro objetivo esta relacionado con la elaboración, por primera vez tendremos en cuenta la tecnología que se deberá utilizar en el proyecto.

#### 3.1 Diseño arquitectónico

El diseño arquitectónico engloba la configuración de red, la utilización o no de un marco que pudiera estar disponible, el establecimiento de los subsistemas, sus interfaces y las dependencias entre estos.

- Configuración de red.

La configuración de red viene determinada por los nodos que tendremos, en este caso, y tal y como se indica en los requerimientos tendremos la gestión centralizada pero la adquisición de datos se realizara en cada servidor de impresión, los cuales están en distintas localizaciones. El sistema se aprovechara de la actual infraestructura de red existente. En la figura 44 se indican los actuales nodos, las características de las conexiones así como su ancho de banda.

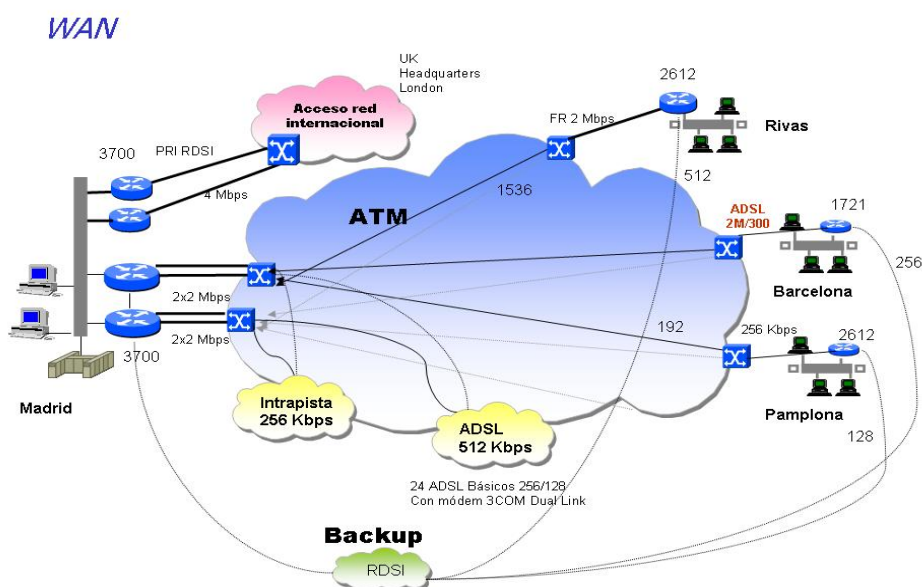


Figura 44. Configuración de red.

En cuanto a la fiabilidad de las conexiones, todas ellas disponen de back-up mediante líneas RDSI con un ancho de banda acorde con el tráfico de red de cada nodo.

- Subsistemas

Los subsistemas a usar pueden ser propios del proyecto o bien software de mercado (comercial) e incluso subsistemas de otros proyectos (reusabilidad).

En nuestro caso partiremos de la descomposición del software en paquetes que ya realizamos en la etapa de análisis (figura 10, paquetes de análisis y servicios) en el cual indicaremos las modificaciones que consisten en la inclusión del subsistema para exportar los datos de impresión del propio log interno del sistema operativo. Este subsistema esta formado por el programa eventquery proporcionado por el propio sistema operativo Windows 2000 mediante su kit de recursos. Este subsistema dispone de su propia interfaz, es decir, de operaciones que se pueden pedir desde otro subsistema. En nuestro caso solo se le solicitara una operación, definida mediante el siguiente comando:

```
eventquery.pl -filter ID=10 -format TABLE -verbose System
```

Donde las diferentes opciones se aplicaran bajo estos criterios:

-filter ID=10, filtra los eventos con ID 10 que son los relacionados con la impresión.

-format TABLE especifica el formato de la exportación de datos, en este caso en formato tabla.

-verbose indica que recoja toda la información disponible, no una versión resumida del evento.

System escoge solo los eventos del lóg. de sistema.

La siguiente figura recoge los subsistemas definitivos, incluyendo los subsistemas añadidos en esta fase de diseño:

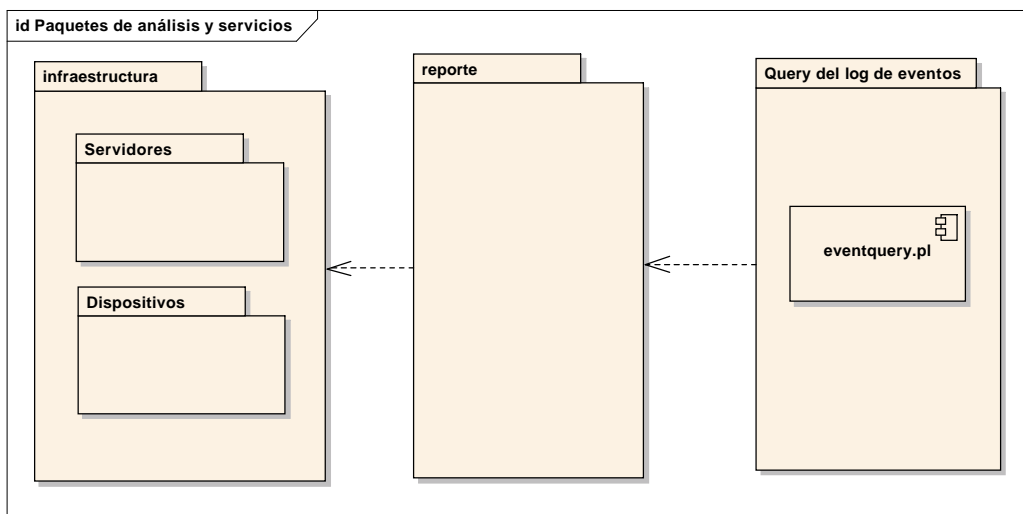


Figura 45. Establecimiento de los subsistemas.

## 3.2 Diseñar casos de uso

En esta etapa de análisis, cuando nos referimos al diseño de los casos de uso, nos estamos refiriendo al diseño de la implementación de estos, para ello, partiremos del diagrama de colaboración resumido efectuado en la etapa de análisis.

La implementación de la funcionalidad de los caso de uso se realiza dentro de las clases de control y de entidades. Es conveniente que sea solo una clase de control la que dirija todo el proceso del caso de uso.

Para especificar los casos de uso utilizaremos especificaciones textuales, podría considerarse como alternativa el uso de diagramas de secuencia, pero consideramos que solo aportarían complejidad, por lo que se opta por la especificación textual.

### Caso de uso 1, "crear servidor"

El usuario selecciona en el menú la opción correspondiente a crear un nuevo servidor, se llama a la operación crear de la clase CreacionServidor, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaCrearServidor.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase CreacionServidor y se devuelve el control al menú.
- 2) Si el usuario confirma la creación, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si se encuentra un servidor con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX ya existe".
  - b) Si no se encuentra el servidor, se llamara a la operación crear de la clase Servidor con los parámetros nombre e ubicación.
- 3) Una vez creado el servidor aparecerá una pantalla modal ofreciendo la posibilidad de añadir dispositivos asociados al servidor recién creado, si en usuario confirma se llamara a la operación crear de la clase CreacionDispositivo descrita en el caso de uso 2.

### Caso de uso 2, "crear dispositivo"

El usuario selecciona en el menú la opción correspondiente a crear un nuevo dispositivo, se llama a la operación crear de la clase CreacionDispositivo, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaCrearDispositivo.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase CreacionDispositivo y se devuelve el control al menú.
- 2) Si el usuario confirma la creación, se llamara a la operación buscar de la clase Dispositivo con el nombre de dispositivo como parámetro. Si el

caso de uso no es una extensión de "crear servidor" previamente se llamara a la operación buscar de la clase servidor, pues el dispositivo ha de estar asociado aun servidor existente.

- a) Si no se encuentra un servidor con el nombre adecuado, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX ya existe".
- b) Si se encuentra el servidor, se llama a la operación buscar de la clase Dispositivo, si se encuentra un dispositivo con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El dispositivo XXXXX ya existe".
- c) Si no se encuentra el dispositivo, se llamara a la operación crear de la clase Dispositivo con los parámetros recogidos de la pantalla PantallaCrearDispositivo.

### Caso de uso 3, "modificar servidor"

El usuario selecciona en el menú la opción correspondiente a modificar un servidor, se llama a la operación crear de la clase ModificacionServidor, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaModificarServidor.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase ModificacionServidor y se devuelve el control al menú.
- 2) Si el usuario confirma la modificación, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si se encuentra un servidor con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX ya existe".
  - b) Si no se encuentra el servidor, se llamara a la operación modificar de la clase Servidor con los parámetros recogidos de la pantalla PantallaModificarServidor.

### Caso de uso 4, "consultar servidor"

El usuario selecciona en el menú la opción correspondiente a consultar un servidor, se llama a la operación crear de la clase ConsultacionServidor, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaConsultarServidor.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase ConsultacionServidor y se devuelve el control al menú.
- 2) Si el usuario confirma la consulta, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si no se encuentra un servidor con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX no existe".



- b) Si se encuentra el servidor, se abrirá una pantalla con el formato PantallaConsultarServidor y se ofrecerá la posibilidad de imprimir los datos mediante la operación imprimir.
- c) Una vez mostrados los datos, se ofrecerá la posibilidad de modificar el servidor, para ello se llamara a la operación crear de la clase ModificacionServidor, descrita en el caso de uso 3.

#### Caso de uso 5, "borrar servidor".

El usuario selecciona en el menú la opción correspondiente a borrar un servidor, se llama a la operación crear de la clase EliminacionServidor, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaBorrarServidor.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase EliminacionServidor y se devuelve el control al menú.
- 2) Si el usuario confirma la eliminación, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si no se encuentra un servidor con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX no existe".
  - b) Si se encuentra el servidor, se abrirá una pantalla con el formato PantallaBorrarServidor y se llamara a la operación borrar de la clase servidor.
  - c) Una vez eliminado el servidor se han de eliminar los dispositivos asociados a el para lo cual se llamara a la operación crear de la clase EliminacionDispositivo, descrita en el caso de uso 6, para cada uno de los dispositivos asociados al servidor borrado.

#### Caso de uso 6, "borrar dispositivo"

El usuario selecciona en el menú la opción correspondiente a borrar un nuevo dispositivo, se llama a la operación crear de la clase EliminacionDispositivo, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaBorrarDispositivo.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase EliminacionDispositivo y se devuelve el control al menú.
- 2) Si el usuario confirma la eliminación, se llamara a la operación buscar de la clase Dispositivo con el nombre de dispositivo como parámetro. Si el caso de uso no es una extensión de "borrar servidor" previamente se llamara a la operación buscar de la clase servidor, pues el dispositivo ha de estar asociado aun servidor existente.
  - a) Si no se encuentra un servidor con el nombre adecuado, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX ya existe".

- b) Si se encuentra el servidor, se llama a la operación buscar de la clase Dispositivo, si no se encuentra un dispositivo con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El dispositivo XXXXX no existe".
- c) Si se encuentra el dispositivo, se llamara a la operación borrar de la clase Dispositivo con los parámetros recogidos de la pantalla PantallaBorrarDispositivo.

#### Caso de uso 7, "consultar dispositivo"

El usuario selecciona en el menú la opción correspondiente a consultar un dispositivo, se llama a la operación crear de la clase ConsultacionDispositivo, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaConsultarDispositivo.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase ConsultacionDispositivo y se devuelve el control al menú.
- 2) Si el usuario confirma la consulta, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si no se encuentra un servidor con el nombre adecuado, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX no existe".
  - b) Si se encuentra el servidor, se llama a la operación buscar de la clase Dispositivo, si no se encuentra un dispositivo con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El dispositivo XXXXX no existe".
  - c) Si se encuentra el dispositivo, se abrirá una pantalla con el formato PantallaConsultarDispositivo y se ofrecerá la posibilidad de imprimir los datos mediante la operación imprimir.
  - d) Una vez mostrados los datos, se ofrecerá la posibilidad de modificar el dispositivo, para ello se llamara a la operación crear de la clase ModificacionDispositivo, descrita en el caso de uso 8

#### Caso de uso 8 "modificar dispositivo".

El usuario selecciona en el menú la opción correspondiente a modificar un servidor, se llama a la operación crear de la clase ModificacionDispositivo, esta operación realizara la instanciacion de la clase, abriendo una ventana e instanciando el formato de pantalla PantallaModificarDispositivo.

- 1) El usuario podrá cancelar, mediante el botón correspondiente, momento en el cual se llama a la operación borrar de la clase ModificacionDispositivo y se devuelve el control al menú.
- 2) Si el usuario confirma la modificación, se llamara a la operación buscar de la clase Servidor con el nombre de servidor como parámetro.
  - a) Si no se encuentra un servidor con el nombre adecuado, se abre una ventana con el formato PantallaError mostrando el mensaje "El servidor XXXXX no existe".

- b) Si se encuentra el servidor, se llama a la operación buscar de la clase Dispositivo, si no se encuentra un dispositivo con el mismo nombre, se abre una ventana con el formato PantallaError mostrando el mensaje "El dispositivo XXXXX no existe".
- c) Si se encuentra el dispositivo, se abrirá una pantalla con el formato PantallaModificarDispositivo y se llamará a la operación modificar de la clase Dispositivo.

#### Caso de uso 9, "generar log".

En este caso el usuario es el propio sistema, por lo que no accede al menú, aunque su rol puede extenderse a un usuario "real" que si seleccionaría del menú la opción correspondiente. El sistema llamaría a la operación crear de la clase GeneracionLog, esta operación realiza la instanciación de la clase.

- 1) Se llamará a la operación consulta de la clase Log para obtener los datos del Log.
- 2) Con los datos del Log obtenidos se llamará a la operación crear de la clase GeneracionFichero para generar el fichero de texto con los datos del Log del sistema.
- 3) Si la generación del fichero ha sido correcta, se procederá a borrar el Log del sistema mediante la operación borrar.

#### Caso de uso 10, "exportar log".

En este caso el usuario es el propio sistema, por lo que no accede al menú. El sistema llamaría a la operación crear de la clase ExportacionLog, esta operación realiza la instanciación de la clase.

- 1) Se llamará a la operación crear de la clase GeneracionFichero para copiar el fichero de cada localización al servidor central desde el que se efectuara la carga a la BD.
- 2) Una vez copiado el fichero, se procede a su carga en la BD, mediante la operación crear de la clase Sumarizacion, descrita en el caso de uso 11. esta operación también efectuara una sumarización en la BD para evitar guardar un detalle superior a tres meses según requerimientos.
- 3) Si la sumarización, carga en la BD, ha sido correcta, se procederá a borrar el fichero del sistema mediante la operación borrar.

#### Caso de uso 11, "sumarizar".

Una vez copiado el fichero, se procede a su carga en la BD mediante la clase sumarización (operación crear), esta operación también efectuara una sumarización en la BD para evitar guardar un detalle superior a tres meses según requerimientos.

### 3.3 Obtener diagrama estático de diseño

La obtención del diagrama estático de diseño se podría ir obteniendo conforme se avanza en el diseño de los casos de uso. En nuestro caso lo realizaremos una vez finalizado el diseño de los casos de uso. Este diagrama no contendrá todas las clases que conformaran el software, por ejemplo no contendrá las clases de la interfaz grafica, en caso de realizar el desarrollo final con una herramienta orienta a objetos (con lo que se generaran muchas mas clases en la fase de desarrollo), para evitar aportar una complejidad innecesaria no contemplaremos estas clases.

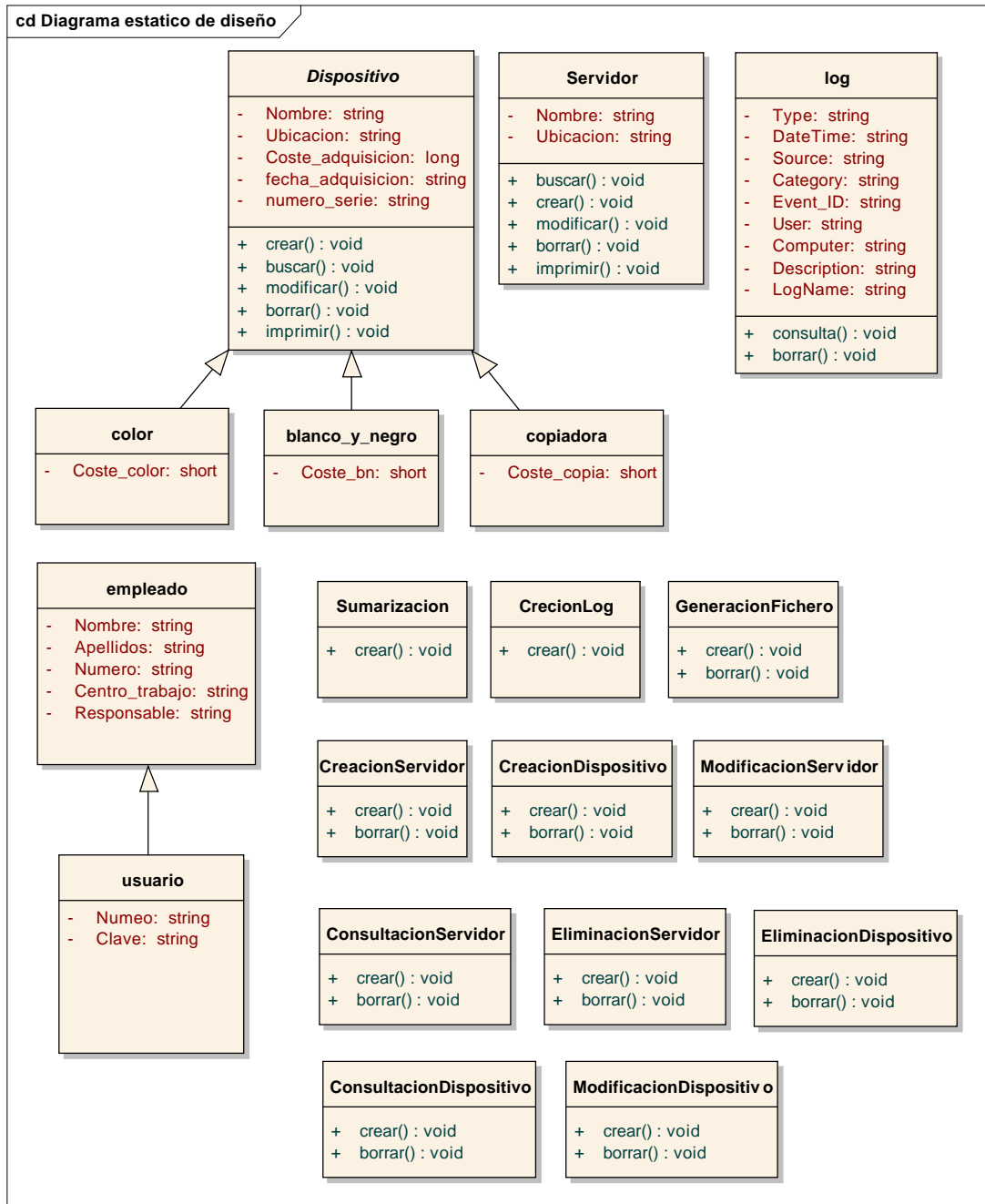


Figura 46. Diagrama estático de diseño.

### 3.4 Diseñar la persistencia

El diseño de la persistencia se realizara basándonos, según requerimientos aportados en la fase de recogida y documentación de requisitos, en una base de datos relacional, concretamente Microsoft SQL-Server.

#### Diseño de la Base de datos.

Para diseñar la base de datos, en un primer momento eliminaremos la herencia, posteriormente se realizara el diagrama Entidad-Relación equivalente al diagrama estático de las clases persistentes para finalmente obtener la especificación de la base de datos relacional.

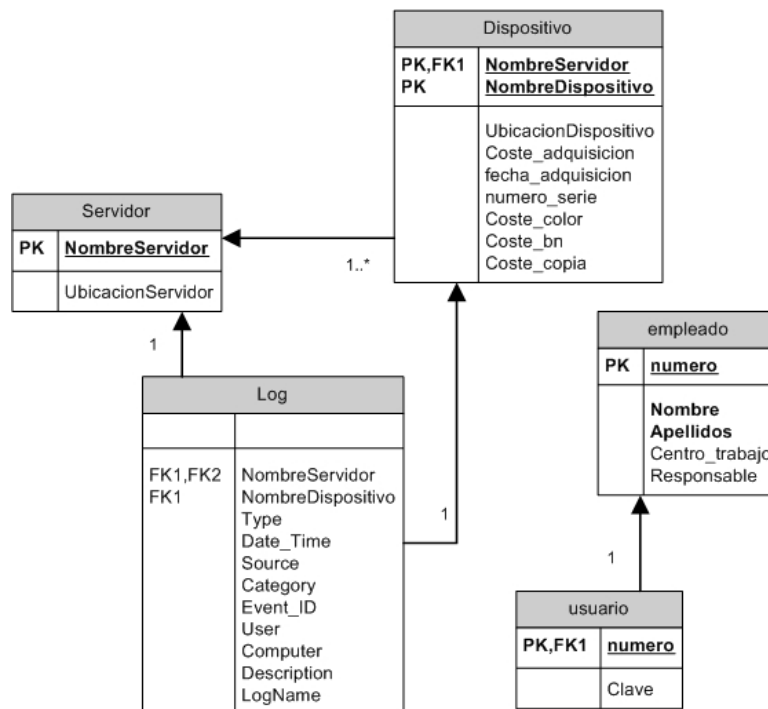
- Supresión de la herencia.

Atendiendo al diagrama estático de diseño, mostrado en la figura 46, existen dos jerarquías de herencia encabezadas por las clases Dispositivo y empleado.

En el primer caso para eliminar la herencia se va a optar por una sola tabla que incorpore los atributos de las subclases, incorporar una tabla para cada subclase desperdiciaría mucho espacio al estar la mayor parte vacías por cada elemento de la superclase dienta situación si se incorporase una tabla para cada subclase.

En el caso de la superclase empleado, aun cuando según los requerimientos es una tabla que ya se encuentra en el sistema de gestión de bases de datos existente, la elección seria clara, pues la subclase solo se daría para un reducido conjunto de elementos de la superclase, la elección mas idónea seria por tanto una tabla para la superclase y otra para la subclase. De cara a facilitar la comprensión del diseño de la BD incorporaremos la tabla empleados en el diagrama Entidad-Relación.

- Diagrama Entidad-Relación.



- Base de datos Relacional.

La descripción de la base de datos la realizaremos ateniéndonos al sistema de gestión de base de datos usado, Microsoft SQL-Server:

```

CREATE TABLE Servidor (
  NombreServidor varchar(50)
  ,UbicacionServidor varchar(50) NULL
  ,CONSTRAINT PK_Servidor PRIMARY KEY (NombreServidor)
)
go
  
```

```

CREATE TABLE Dispositivo (
  NombreServidor varchar(50)
  ,NombreDispositivo varchar(50)
  ,UbicacionDispositivo varchar(50) null
  ,Coste_adquisicion float null
  ,fecha_adquisicion datetime null
  ,numero_serie varchar(50) null
  ,Coste_color float null
  ,Coste_bn float null
  ,Coste_copia float null
  ,CONSTRAINT PK_Dispositivo PRIMARY KEY (NombreServidor,NombreDispositivo)
  ,CONSTRAINT FK2_Dispositivo FOREIGN KEY (NombreServidor) REFERENCES Servidor(NombreServidor)
)
go
  
```

```

CREATE TABLE Log (
  NombreServidor varchar(50)
  ,NombreDispositivo varchar(50)
  ,Type varchar(50)
  ,Date_Time varchar(50)
  ,Source varchar(50)
  ,Category varchar(50)
  ,Event_ID varchar(50)
  ,User varchar(50)
  ,Computer varchar(50)
  ,Description varchar(50)
  ,LogName varchar(50)
  ,CONSTRAINT PK_Log PRIMARY KEY (NombreServidor,NombreDispositivo)
  ,CONSTRAINT FK1_Log FOREIGN KEY (NombreServidor) REFERENCES Servidor(NombreServidor)
  ,CONSTRAINT FK2_Log FOREIGN KEY (NombreServidor,NombreDispositivo)
  REFERENCES Dispositivo(NombreServidor,NombreDispositivo)
)
go
  
```

```

CREATE TABLE empleado (
  numero varchar(50)
)
  
```

### 3.5 Diseñar interfaz de usuario

En nuestro caso, no aplicamos ninguna guía de estilo, pues no se nos ha especificado así en la fase de recogida y documentación de requisitos. Aun así, a continuación presentaremos una guía para representar los diferentes datos, diálogos y diseño de ventanas para la posterior construcción de la aplicación. Resaltar que será una guía muy poco restrictiva en tanto no se conoce la herramienta final de codificación.

- Presentación de datos.

Una correcta presentación de los datos exige que cada dato se presente de la misma manera en todos los formularios en lo que se presente, además, cada dato, según su topología se habrá de presentar de la siguiente forma:

- Los datos de tipo texto se presentaran en áreas de texto con barras de desplazamiento en aquellos casos en que su extensión exceda del tamaño de área.
- Los campos de longitud fija se presentaran como campos de texto en caso de ser de salida, en caso contrario, y solo si la lista de posibles valores es reducida se presentaran como un combo.
- En cualquier caso si el dato a presentar es de tipo booleano se usaran check boxes.

- Implementación de diálogos.

Los menús se corresponderán a los casos de uso descritos mas aquellas opciones propias de la aplicación como son salir y el menú de ayuda en línea. La presentación será con un menú de barra, con despleables para "sistema", "servidores", "dispositivos", "log" y "ayuda". No se considera necesario la presencia de una barra de herramientas.

El menú se presentara en la parte superior izquierda de la pantalla.

- Formato de ventanas.

El formato de ventana estará fuertemente ligado a la herramienta final de codificación, en toda caso será el Standard Windows, tal y como se presentan en el apartado 2.8 "análisis de la interfaz de usuario"

### 3.6 Diseño de los subsistemas

Los subsistemas coincidirán con los paquetes que previamente se han establecido en la etapa de análisis, estos paquetes son los siguientes:

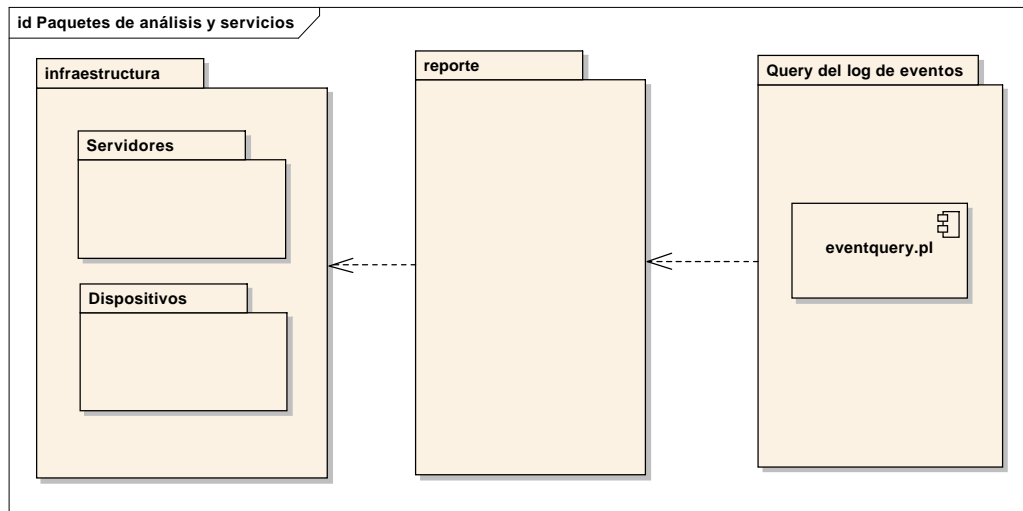


Figura 45. Establecimiento de los subsistemas.

El contenido de los subsistemas es el siguiente:

- Servidores

Comprende la clase Servidor y los casos de uso crear servidor, modificar servidor, consultar servidor y borrar servidor.

- Dispositivos

Comprende la clase Dispositivo y los casos de uso crear dispositivo, borrar dispositivo, consultar dispositivo y modificar dispositivo.

- Reporte

Comprende la clase log y los casos de uso generar log, exportar log y sumarizar.

- Query del log de eventos

Subsistema que forma parte del propio sistema operativo de servidor de impresión, en concreto del kit de recursos de Windows 2000, en nuestro caso estará formado por la utilidad eventquery.pl



## 4. Conclusión

---

Este trabajo fin de carrera ha supuesto, al menos para mi, dos conclusiones bien diferenciadas. Por un lado y desde una perspectiva de la ingeniería del software ha sido revelador acerca del actual uso de metodologías en la producción de software. No es este el primer proyecto de producción software en el que participo, sin embargo, si ha sido el primero en el cual he aplicado de manera rigurosa la metodología UML en el ciclo de vida del software. Esto me ha permitido comprobar la importancia de su uso, que invariablemente aporta ventajas, tanto desde un punto de vista de calidad del software como desde la perspectiva del ahorro de costes.

Por otro lado, este proyecto esta basado en un problema real, que afecta a una empresa real, y que puede ser replicado a otras empresas con suma facilidad. El proyecto ha servido para realizar el análisis y diseño de un sistema que facilitara la toma de decisiones enfocadas al ahorro u optimización de recursos, en este caso la impresión, sin por ello mermar la calidad del servicio prestado.

No quisiera terminar este apartado dedicado a las conclusiones sin mencionar la labor de síntesis que este trabajo fin de carrera ha supuesto respecto al conjunto de mis estudios de ingeniería técnica en informática en la UOC.

## 5. Agradecimientos

---

Uno de los pilares de este proyecto fue trabajar con datos reales, partiendo de una problemática real. Esto no habría sido posible si la colaboración de la empresa 3M España, y en concreto la figura de José Miguel Olivares como director del departamento de tecnologías de la información.

Jordi Fernández González ha dirigido este proyecto, resolviendo mis dudas aportándome claridad en los momentos necesarios.

Mi familia que ha soportado que les robara tiempo para dedicarlo a este proyecto.

GRACIAS A TODOS

## 6. Bibliografía

---

Kruchten, Philippe.

**Rational Unified Process, The: An Introduction, Third Edition**

Addison Wesley. Diciembre, 2003

G. Booch, J. Rumbaugh, I. Jacobson.

**El Lenguaje Unificado de Modelado.**

Addison Wesley Iberoamericana, 1999.

G. Booch. Benjamin/Cummings.

**Object-Oriented Analysis and Design.**

1994.

G. Booch, I. Jacobson and J. Rumbaugh.

**The UML Specification Document.**

Rational Software Corp., 1997.

I. Jacobson.

**Object-Oriented Software Engineering: A Use Case Driven Approach**

Addison-Wesley, 1992.

C. Larman.

**UML y Patrones.**

Prentice Hall, 1999.

J. Rumbaugh [et al].

**Object-Oriented Modeling and Design.**

Prentice- Hall,1991.

**UML Resource Center.** Rational Software.

<http://www.rational.com/uml/>