

# **Sistema Administración de Base de Datos del Departamento de Audiovisuales Comunicación Interna de TMB**

Ernesto Oliva Barrera  
ETIS

Antoni Oller Arcas  
David Bayo Julve

**Data Lliurament**  
16 junio de 2006

---

Indice.....	2
Plan de trabajo.....	3
Introducción.....	3
Organización del Equipo.....	3
Recursos del Software.....	4
Definición de Tareas.....	5
Calendario del Proyecto.....	6
Temporalización y Documentación.....	7
Project.....	8
Descripcion del Modelo de Negocio.....	9
Introducción.....	9
Modo de Trabajo Actual.....	10
Situación Actual.....	12
Imagen de las Tablas.....	14
Interficie Actual.....	17
Situación de Futuro.....	18
Caso práctico.....	19
Ejemplo de DVD.....	21
Diagrama de Clases.....	22
Glosario de términos.....	23
Diagramas de Casos de Uso.....	24
Objetivos.....	25
Diseño.....	26
J2EE.....	27
Modelo de desarrollo J2EE.....	28
Struts Implementacion del patrón MVC en java.....	31
Capa de presentación.....	34
Diagrama Grafico de las funciones de pantalla.....	35
Interficie Menu Principal.....	37
Interficie Nuevo.....	38
Interficie Listar.....	39
Interficie Modificar y Borrar.....	40
Diagramas de Secuencias.....	41
Diagramas de Colaboración simplificados.....	51
Diagramas de componentes.....	57
Base de Datos.....	58
Modelo Lógico.....	59
SGBD utilizado.....	60
Modelo de datos.....	61
Implementación.....	62
Struts.....	63
Persistencia temporal.....	66
Capa de negocio.....	68
Conclusiones.....	70
Bibliografía.....	71

## Plan de trabajo

### Introducción

El departamento de Audiovisuales de TMB dedicado a la elaboración de reportajes, videos y publicaciones sobre todas las actividades de la empresa se plantea la necesidad de tener una aplicación vía Web donde se pueda acceder de una manera fácil y cómoda a la base de datos de sus productos, esta funcionalidad tendrá como usuarios en una primera fase el propio departamento y en una segunda fase para usuarios que accedan desde diferentes puntos accesos (Kioscos) repartidos por toda la ciudad.

El departamento necesita una base de datos correctamente creada, actualmente tienen una BD en Access, sin relaciones, la interficie que usan es compleja y muy primitiva.

Como objetos persistentes tienen los videos ya tratados/editados y el resto de objetos importantes para la correcta función del Departamento los tienen solo de manera física.

Nuestra propuesta consiste primero en crear una Base de Datos relacional , además de tener en cuenta la inmensa cantidad de productos audiovisuales que va generando el departamento, es que este proyecto sea fácilmente reutilizable para poder seguir ampliando las funcionalidades de la aplicación sin tener que hacer grandes cambios, para por ejemplo, además de los miembros del departamento, usuarios externos puedan visualizar estos trabajos desde varios puntos de acceso que proporciona la empresa por toda la ciudad.

El proyecto, por cuestiones de temporalización, será desarrollar la primera fase que consistirá en la creación de la Base de Datos del Departamento y una aplicación sobre plataforma J2EE que vía Web acceda a dicha base de Datos.

Esta aplicación Web permitirá a los miembros del departamento administrar la base de datos, creación, lecturas, modificación, eliminación de productos y sobre todo la necesidad más imperativa del departamento que se puedan realizar búsquedas entre los diferentes objetos con respuestas rápidas y eficaces.

### Organización del equipo:

El Equipo que desarrollará este proyecto estará formado por *Ernesto Oliva Barrera, Antoni Oller y David Bayo*.

*Ernesto Oliva Barrera* estudiante realizando el Trabajo de fin de Carrera, se encargará de hacer el trabajo planificado.

*Antoni Oller* por parte de la Universitat Oberta de Catalunya, supervisará y asesorará al estudiante.

*David Bayo* por parte de Transportes Metropolitanos de Barcelona, supervisará y asesorará al estudiante.

## Recursos de Software:

Para la realización del proyecto contamos con varios recursos, para facilitar el trabajo a realizar.

.El campus UOC permitirá la comunicación entre el estudiante y el consultor de la UOC.

.La empresa TMB mediante un convenio de colaboración ofrecerá al estudiante tutor y prácticas en sus instalaciones.

Software:

- JDK 1.4 version 1.4.2\_09
- Eclipse, versión 3.0 \*\*
- JBoss, versión 4.0.2 \*\*
- Ant, versión 1.6.5 \*\*
- MySQL, versió 5.0.18 \*\*
- MySQL connector Java, versió 3.1.12 \*\*
- MySQL Administrator versió 1.1.9 \*\*
- MySQL Query Browser versió 1.1.20 \*\*
- JBoss-IDE Eclipse 1.5 \*\*

\*\* El software puede estar sujeto a cambios, dependiendo del departamento de informática de TMB.

**Definición de tareas:**

- .Plan de trabajo
  - Introducción
  - Organización y software
  - Calendario
- .Análisis
  - Recogida de Información
    - Modelo del dominio
    - Modelo del Negocio
  - Casos de Uso
  - Diagrama de Entidades
- .Diseño
  - Diagramas
    - Base de Datos
      - Diagrama de Clases y Relaciones
    - Diagrama de Clases
    - Diagrama de Casos de Usos
    - Diagrama de Colaboraciones
  - Recogida y Documentación de Requisitos de la Interficie de Usuario
    - Documentación de las tareas actuales y futuras.
    - Especificaciones de Usabilidad.
- .Implementación
- .Pruebas de Integración
- .Redacción de de la memoria
- .Formación en J2EE.

### Calendario del Proyecto

Tarea	temporalización	Inicio	Fin
Plan de Trabajo	6 días	6/03/06	12/03/06
Introducción	2 días	6/03/06	8/03/06
Organización y software	2 días	8/03/06	10/03/06
Calendario	2 días	10/03/06	12/03/06
Análisis	21 días	13/03/06	2/04/06
Recogida de Información	10	13/03/06	22/03/06
Modelo del Dominio	6	13/03/06	18/03/06
Modelo de Negocio	4	19/03/06	22/03/06
Casos de Uso	5	23/03/06	27/03/06
Diagrama de Entidades	6	28/03/06	2/04/06
Diseño	19	03/04/06	21/04/06
Diagramas			
Base de Datos			
Diagrama de Clases y Relaciones	8	03/04/06	10/04/06
Diagrama de Clases	4	11/04/06	14/04/06
Diagrama de Casos de Uso	3	15/04/06	17/04/06
Diagrama de Colaboraciones	4	18/04/06	21/04/06
Recogida y Documentación de Requisitos de la Interficie de Usuario	3	22/04/06	24/04/06
Documentación de las tareas actuales y futuras.	1	22/04/06	22/04/06
Especificaciones de Usabilidad	2	23/04/06	24/04/06
Implementación	35	25/04/06	29/05/06
Pruebas de Integración	10	30/05/06	8/06/06
Redacción de de la memoria	8	9/06/06	16/06/06
Formación en J2EE.	340 horas	13/03/06	17/07/06

### Temporalización y Documentación

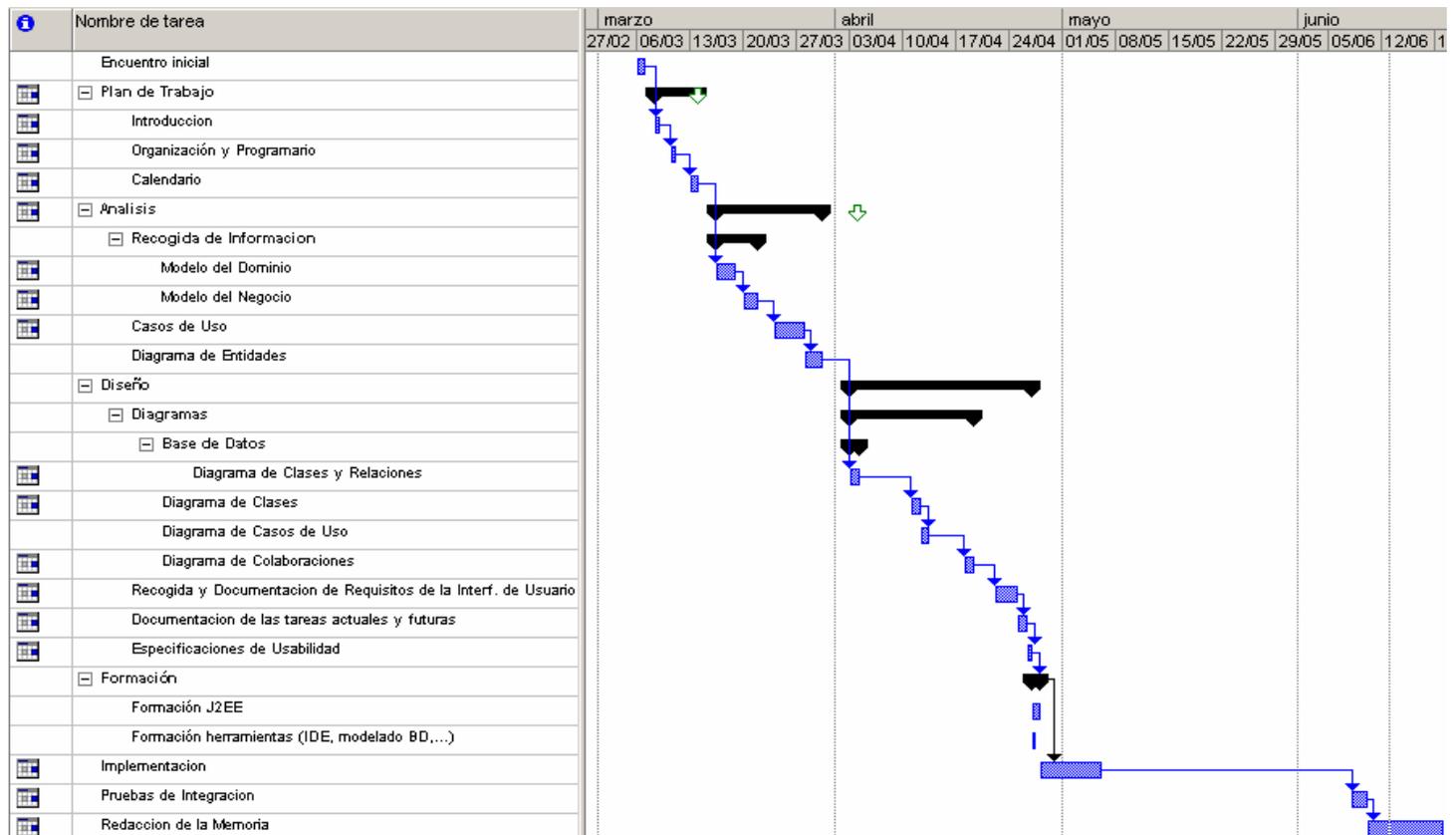
El calendario estará sujeto a cambios según la complejidad de algunas tareas, por lo que es orientativo a excepción de las fechas límites de entregas parciales de cada fase que constarán de las siguientes fechas

Entrega	Fecha
<b>Análisis</b>	<b>03/04/06</b>
<b>Diseño</b>	<b>21/04/06</b>
<b>Implementación</b>	<b>29/05/06</b>
<b>Entrega Final</b>	<b>16/06/06</b>

La entrega final del proyecto constará de la aplicación desarrollada y de la memoria del proyecto.

La memoria del proyecto incluirá toda la información del proceso de desarrollo (análisis, especificación, diseño) además de la información que necesitará el usuario para el correcto funcionamiento de la aplicación (manual de administración, manual de usuario).

### Calendario del proyecto:



Me gustaría comentar que el día 29 de junio me diagnosticaron una supuesta gastroenteritis aguda que resultó ser ,después de los análisis hechos en el Hospital Vall d’Hebron , una salmonella enteritidis, razón por la cual tuve que ,muy a mi pesar ,dejar de trabajar en el proyecto del día 29 de mayo hasta el día 8 de junio.

Por lo que parte, aunque no toda, de los objetivos incumplidos fueron debido a este contratiempo.

## Descripción del modelo de negocio

### Introducción

Transportes Metropolitanos de Barcelona (en adelante TMB), es una empresa ubicada en Barcelona dedicada a prestar servicios de transporte colectivo de viajeros en la ciudad de Barcelona, así como en diversos municipios de su área de influencia.

Dentro de este transporte se enmarca la explotación de todas las líneas de metro, ciertas líneas de autobús y determinados transportes de ocio (Buses turísticos, Tranvía azul,...).

Formando parte de la empresa existe un departamento encargado de producir, almacenar, ceder y recibir material audiovisual. Este departamento, denominado el departamento de Audiovisuales, se plantea la necesidad de tener una aplicación que le permita la catalogación de sus productos audiovisuales. Actualmente se recoge información audiovisual relacionada con el transporte público y la almacena en diversos soportes audiovisuales. Por otro lado ya disponen de una herramienta informática para conocer qué documentos audiovisuales han sido realizados por producción propia o simplemente cedidos / entregados por otras entidades colaboradoras. El problema surge en que la aplicación actual no cubre las necesidades actuales y requieren una nueva aplicación.

A continuación se describe el modo de trabajo actual y la posible situación de futuro, para así acabar obteniendo los requisitos de la futura aplicación.

### **Importancia de los documentos audiovisuales para la TMB:**

Una empresa de las características de TMB, que gestiona la mayor parte del transporte Público en una ciudad de la magnitud de Barcelona, es una empresa dinámica con continuos cambios de infraestructura y modernización de sus instalaciones y vehículos, en constante crecimiento y desarrollo, genera un volumen importante de información, lo cual hace patente la necesidad de documentar estos cambios, ya que esta información, además de ser de utilidad técnica para la empresa, se convierte en el devenir de los años en documentos históricos de la transformación de la ciudad.

Por estos motivos TMB dispone de un departamento propio para la realización, producción y edición de documentos audiovisuales, el cual posee todo el equipo necesario, cintas de varios formatos para grabaciones, cámaras de filmación, equipos de audio, equipos de lectura de cintas de diferentes formatos, equipos para el tratamiento de audio y sonido, ordenadores y el software requerido para las diferentes funciones.

---

### **Modo de trabajo actual**

La empresa encarga al departamento audiovisual la creación de documentos audiovisuales de diferentes temáticas que registran las actividades más significativas de la empresa tales como inauguraciones de nuevas instalaciones, nuevos vehículos, convenciones, formación, remodelación de estaciones de metro, entre otras, así como la posterior edición o tratamiento digital de los mismos. Disponiendo para ambas funciones de todo el equipo necesario.

El departamento está formado por 4 personas:

- Jorge Jiménez jefe del departamento. Además de efectuar las funciones de realizador, producción, tratamiento digital de los documentos elaborados.
- Iris Garcia miembro del equipo de realización y del equipo de postproducción o tratamiento digital de las imágenes,
- Xavier Pomes miembro del equipo de realización y del equipo de postproducción o tratamiento digital de las imágenes.
- Alex Trillo. Se encarga de elaborar los gráficos que aparecen en los documentos audiovisuales ya tratados.

Cuando el departamento se dispone a realizar un documento audiovisual de alguna actividad de la empresa, seleccionan unas cintas en blanco (es el cassette/cinta que solemos poner dentro de una cámara de video para grabar una boda, comunión, fiesta familiares etc.) se les asigna un número de master, fecha, temática y formato. Con el equipo técnico y personal requerido, se desplazan al sitio donde tenga lugar el evento, una vez seleccionados los planos a filmar comienzan a grabar secuencias sobre el tema en concreto.

A estas secuencias se le pone un título, temática, fecha, descripción, formato, duración, y se le asigna un nº de orden (este nº de orden sigue la numeración de todas las secuencias / reportajes que se van haciendo en el departamento) estas secuencias son independientes entre ellas ya que graban diferentes momentos en diferentes tomas del lugar y pueden ser usadas en diferentes documentos cada una según lo requiera.

Estas grabaciones son hechas por el equipo de realización del departamento, otras veces se puede requerir equipos de realización externos.

La cinta donde se efectúan las grabaciones llamada Master de Cámara se puede utilizar en grabaciones de otros temas o lugares, para no desaprovechar los espacios sin grabar de la cinta, por lo que en un Master de Cámara pueden haber secuencias de temáticas diferentes, por ejemplo en una cinta Master de cámara pueden estar grabadas secuencias de la visita de una personalidad del ayuntamiento a una estación de metro, de temática "Institucional" y secuencias o filmaciones de cursos de formación que se les imparte a empleados de la empresa de temática "Videos Didácticos".

Una vez en el departamento ,mediante un ordenador y el software correspondiente ,comienza la postproducción del master de cámara, basado en el tratamiento audiovisual a las secuencias grabadas, que consiste en añadirle sonido, locución ,gráficos ,recorte de planos ,etc. a las imágenes grabadas. El resultado de este tratamiento a cada secuencia, es un video que mantiene el nº de orden de la secuencia o filmación, el título, la descripción, temática, formato, equipo, duración y se le añade la duración y formato propio.

Del tratamiento del Master de cámara y las secuencias grabadas en él se obtiene una cinta llamada Master de Edición que contiene todas las secuencias grabadas en el master de cámara ya tratadas / editadas, este master de edición mantiene el numero de master ,los números de orden de las secuencias y la temática.

Un Master de Edición proviene de un solo Master de Cámara.

De un master de Edición se pueden obtener / pasar a varios formatos por ejemplo DVCAM, BETACAM o DVD.

Un master de edición es una cinta de un formato a los que solo algunos posibles usuarios podrían visionar, por lo que el departamento a partir de estos master de edición, usando software correspondiente, equipos y ordenador, crea documentos audiovisuales, que pueden tener varios formatos. DVD es uno de los formatos más usados actualmente por el departamento ya que es un formato mucho más exportable a mayor cantidad de posibles usuarios.

Un DVD es un documento audiovisual que trata sobre un solo tema o temática y contiene varios videos (secuencias / reportajes ya editados), un DVD se identifica por número de master, título, formato y fecha.

Ejemplo de un DVD titulo “El bus de Hidrogeno” tiene 4 vídeos que describen cada uno diferentes fases del reportaje, uno describe como se almacena el hidrogeno, otro como se carga este al autobús, recorrido de un bus de hidrogeno por la ciudad.

El propio departamento mediante su grafista, se encarga de la elaboración de las carátulas de presentación de los DVD, en la cual describe el titulo, equipo de producción.

En la misma caja del DVD hay otro DVD llamado Master de Datos que contiene, todo el material utilizado en la realización del DVD ,estos datos están para que se entienda como un puzzle ,fotos ,escenas ,carátulas ,gráficos ,sonidos ,música ,mpg2 mientras en el DVD llamémosle origen están los videos Editados que utilizan todos los datos del Master de Datos. Los elementos de un Master de Datos pueden estar en varios Vds.

## Situación actual

El departamento dispone de dos salas de trabajo, una de 10x10 metros, donde se realiza el tratamiento de las imágenes, en esta sala están dispuestos varios equipos, según su función, estos equipos son variados, equipos para formatos diferentes de cintas conectados a PC, para permitir mediante software y tecnología adecuada el cambio de un formato a otro, equipos para tratamiento de sonido e imágenes y estaciones de trabajo (PCS) del departamento.

Adyacente a esta, poseen una sala de 5x5 metros, donde tienen 5 ordenadores, uno de ellos conectado en red a los kioscos (PCS que la empresa tiene distribuidos por las diferentes instalaciones en la ciudad, conectados a la red interna). En esta misma sala se encuentran distribuidos 3 armarios de dimensiones 1x2 metros, donde almacenan separados según su topología, Master de Cámara, Master de Edición y DVDs.

## Situación tecnológica actual:

El departamento tiene creada una base de datos, usando como herramienta la suite informática Microsoft Office-Access, que consta de 4 tablas sin relaciones entre ellas, las tablas son:

- **Temática** :Son la 7 temáticas de las que actualmente tiene creadas el departamento para clasificar los trabajos .Esta tabla tiene dos campos identificativos

*idtematica* :campo numérico como clave primaria.

*Tipotematica* :campo de texto ,identifica el tipo de temática.

- **Formato**: Son los diferentes formatos con los que trabaja el departamento en los documentos. Sus campos identificativos son:

*Idformato* :campo numérico como clave primaria.

*Tipoformato* :campo textual que identifica el tipo de formato.

- **Cintas de Entrada** :Son los videos o secuencias tratadas ,que ha ido creando el departamento. Sus campos identificativos son:

*n°master* :campo numérico que hace referencia al master de cámara del cual es origen esta secuencia.

*n°orden* :campo numérico correlativo a los videos que ha ido creando el departamento.

*Temática* :campo textual, identifica el tipo de temática del video.

*Fecha*: campo tipo Date, indica la fecha en que se grabó la secuencia.

*Titulo*: campo textual que da titulo a la secuencia.

*Descripción*: campo textual .Describe de que trata la secuencia.

*Duracion*: campo numérico. Indica la duración de la grabación original de la secuencia.

*duracion*: campo tipo Time. Indica el tiempo en horas : minutos : segundos ,que dura la secuencia ya tratada, es decir después de postproducida.

*Formato*: Campo textual. Indica el formato en el que se grabó la secuencia.

*equipo de Producción*: campo textual. Indica las personas ,que en diferentes funciones, participaron en la creación de la secuencia.

Esta tabla contiene 694 registros que son videos(secuencias tratadas) .

- **Producción** :Son los tipos de producción(quien elabora el documento audiovisual).

-interno :si el documento lo ha creado el departamento propio de la empresa.

-externo :lo ha creado una empresa externa.

*Id* :campo numérico .Clave primaria

*Campo1*:campo textual que identifica el tipo de producción.

**Imagen de las tablas:**

**Formatos**

Id Formatos	Formatos
1	VHS
2	DVCAM
3	BETACAM
4	DVD
5	CRW
6	DV

**Producción**

Id	Campo1
1	Externa
2	Interna

**Temática**

Id temática	temática
1	Grupos de Trabajo
2	Convenciones
3	Videos Técnicos
4	Actos Sociales
5	Institucionales
6	Generales
7	Videos Didácticos

**Entrada de Cintas**

NºOrden	1	2	3
titulo	CIEN AÑOS DE TRANSPORTE PUBLICO	FILAMACION PASARELA DE ENERGIA	DESCARRILAMENT ZONA UNIVERSITARIA
temática	Institucionales	Videos didácticos	Videos Técnicos
Descripción	Reseña histórica del Transporte Publico en Barcelona desde sus comienzos hasta los años 70.	Prueba de la pasarela en L5 a diferentes alturas	Incidencia de descarrilo en Zona Universitaria
Equipo de Producción	Locución: José Sánchez/ Laboratorios Fotofilm S.A./ Director de Pladevall/Guion y dirección: Marcell Gili/ Sonido: Sonoblock	Audiovisuales de Comunicación Interna	Audiovisuales de Comunicación Interna
duración	20	30	8
duración	10:33:36		
Fecha	01/01/1973	01/10/1995	01/10/1993
Formato	VHS		VHS
NºMaster	1	2	2

Actualmente los master de cámara(cintas con las secuencias o grabaciones en estado puro(sin tratar)),Master de Edición y DVDs.,están almacenados solo en formato físicamente en cintas, de diferentes formatos. Si bien cierto que las secuencias (videos ya tratados) que componen un Master de Edición y un DVD, están guardadas en la tabla Entrada de Cintas de la base de datos del departamento.

Para más fácil comprensión a partir de ahora llamaremos videos a las secuencias ya tratadas.

El departamento crea videos nuevos, utilizando para este fin, fragmentos de videos de su videoteca, accediendo al master de edición del cuál es componente el video, según se requiera trabajar con un formato u otro, para acceder a un video componente de un master de edición se introduce la cinta Master de edición en el equipo correspondiente y se le efectúa el tratamiento requerido. Una vez creados los nuevos videos, se les da alta en la tabla Cintas de entrada de la base de datos, asignándole sus campos identificativos. Un conjunto de estos videos nuevos pueden servir para la elaboración de un DVD u otro formato.

Para realizar búsquedas de videos ,el departamento utiliza un formulario Access de la base de datos, en el cual se van rellorando los campos de interés especial. Esta interfície presenta problemas de interacción tales como es poco usable ,poco intuitiva ,no funcional e ineficiente.

**Captura de pantalla de la interficie actual:**

Entrada de Cintas

Nº Orden: [ ] Título: CIENT AÑOS DE TRANSPORTE PUBLICO Duración: 20 Fecha: 1/01/1973

Temática: [ ] Formato: 1 Formato2: 2

Contenido de la Cinta: Reseña histórica del Transporte Público en Barcelona desde sus comienzos hasta los años 70.

Actores Participantes: [ ]

Equipo de Producción: Locución: Jose Sánchez / Laboratorios Fotofilm S.A. / director de Fotografía: Tomas Pedevell / Guión y dirección: Marcel·l Gil / Sonido: Sonoblock

Tipo de Producción: [ ] Idioma: Catalano

Master 1

Audiovisuales - Comunicación Inter

Análisis del formulario usado como interficie para las búsquedas:

Los campos no están agrupados lógicamente.

La secuencia de los campos no es lógica.

El diseño no es visualmente agradable.

Las etiquetas o nombres de campos ,no son fácilmente reconocibles.

## **Situación de futuro**

El departamento se plantea la necesidad de tener una aplicación para acceder de forma fácil ,cómoda , rápida y segura a su catálogo de productos audiovisuales con diferentes objetivos como por ejemplo la administración de sus productos como crear ,modificar ,actualizar ,leer estos productos .La interficie de esta aplicación debe ser bastante fácil de usar para un usuario novel y con la necesaria potencia para un usuario experto.

Otra de las funcionalidades más necesitadas del departamento es la búsqueda de productos concretos, pueden ser en videos sin tratar, o videos ya tratados o editados, para la creación, elaboración de nuevos, las respuestas de estas búsquedas deben ser lo más rápida y precisa posible.

Actualmente se tienen almacenados en armarios, productos tales como, Master de cámara ,Master de edición ,DVDs .Teniendo en cuenta el volumen de información que anualmente genera la empresa TMB ,dentro de pocos años no tendrán espacio físico para almacenar sus productos ,por lo que necesitarán la informatización de los mismos para su mejor conservación ,además de prevenir posibles daños que le puedan ocurrir a estos con el transcurrir del tiempo.

Otra futura funcionalidad sería la de extender a toda la red interna de la empresa el acceso a estos productos desde otros departamentos con los permisos correspondientes, para que de esta forma un departamento concreto que requiera el uso de un documento audiovisual, validándose correctamente pueda acceder y hacer el uso necesario.

La información del ciudadano es uno de los retos que tiene planteados TMB ,razón por la cual ,sería interesante poner al alcance del público un catálogo de videos de varios temas de interés general ,por ejemplo videos históricos, de esta manera los usuarios externos podrían visionar ,por ejemplo ,como eran los tranvías, autobuses y trenes de metro antiguos ,teniendo en cuenta el volumen de información que genera TMB y su política de aproximación al público.

El departamento de audiovisuales proyecta generar de manera automática informes periódicos de todos los productos de la futura base de datos del departamento ,de estos obtener estadísticas, para planificar futuras acometidas y usarlas como información de referencia del departamento.

### **Caso práctico:**

Al departamento se le encarga filmar las instalaciones de la nueva cochera de Horta ,para estas grabaciones se requiere la intervención de cómo mínimo 2 miembros del equipo de producción ,el realizador y un cámara .Al tratarse de un reportaje de todas las instalaciones de un CON (centro operativo de negocios) ,donde hay variadas instalaciones como sala de mandos, sala de conductores, oficinas de administración, taller mecánico, aparcamientos de autobuses ,el equipo de producción de Audiovisuales puede necesitar más de una cinta para las filmaciones previniendo una larga duración del material a cubrir.

Una vez en la cochera deciden comenzar a filmar una instalación o zona por ejemplo la sala de conductores, que es el área que tienen los conductores para comer, ver la tele, leer ,cuando están de guardia o a punto de comenzar un servicio. El equipo comienza una secuencia grabando a dos conductores en las maquinas de café ,otro conductor tecleando en un Kiosco(PC conectado a la red interna de la empresa),el objetivo es tener un panorama visual de esta sala y todos sus espacios, se termina la secuencia ,a esta secuencia se le asigna un numero de orden (correlativo a todas las secuencias que se han grabado hasta hoy por el departamento) ,se le asigna una temática “Institucionales”,un titulo “Sala de conductores CON HORTA” ,descripción “conductores tomando café en la sala de conductores de la cochera de Horta”,fecha y equipo de producción(equipo que ha participado en la filmación).

El objetivo siguiente es filmar el taller mecánico de la cochera. Comienzan otra secuencia grabando a un mecánico cambiándole una pieza mecánica a un autobús dentro de un foso, se filma la sala de herramientas, las oficinas del taller, se graba también un autobús entrando al taller con una toma exterior comprobándose el sistema automática de las puertas. El objetivo es filmar las zonas de trabajo del taller mecánico. De la misma manera se graban imágenes de las diferentes zonas e instalaciones de la cochera de Horta.

Después de haber filmado todas las áreas de la cochera grabando las zonas representativas de la misma se da por terminada la filmación.

Al master de cámara (cinta que se ha usado para las grabaciones) además del número se le asigna fecha de filmación y la duración total de las secuencias, para de esta manera saber que espacio de cinta queda libre para posibles futuras grabaciones ,también se necesita saber la temática de la cinta ,según el caso puede tener varias y el equipo que ha participado en la elaboración de dicho reportaje ,desde la persona que lo produce, realiza ,guionista ,el cámara y el técnico de sonido ,la mayoría de las veces los miembros del departamento ejercen varias de estas funciones a la vez.

Una vez en el departamento ,se introduce el Master de Cámara en un equipo conectado a un PC que permite el tratamiento de las filmaciones ,a estas filmaciones se necesitan añadir sonido ,recortar planos de las imágenes ,borrar tomas falsas ,añadir música ,locución ,gráficos ,textos ,todo esto es técnicamente llamado postproducción o tratamiento digital de las imágenes.

Este tratamiento del master de cámara se guarda o se graba mediante equipo y software del departamento ,en otra cinta nueva, esta cinta se le denomina Master de Edición que contiene las mismas grabaciones o secuencias del Master de Cámara pero ya tratadas audiovisualmente. Al master de edición se le mantiene el número de master y todos los datos identificativos de las secuencias. Estos datos se escriben en la caja de la cinta.

Una vez se ha finalizado el proceso de postproducción del master de cámara de la cochera de Horta ,esta cinta se almacena en un armario disponible para master de cámara en una habitación del departamento y se comienza a trabajar con el master de edición.

El master de edición pasa a tener la función de copia original de secuencias ya tratadas, y posteriormente será utilizado para la creación de documentos en varios formatos.

Cuando se trata o postproduce una filmación se obtiene como resultado un video con música ,gráficos ,locución ,textos incluidos ,que proviene de una secuencia o filmación y mantiene su número de orden ,título ,descripción ,temática ,fecha ,formato y se le añade la duración de la edición o tratamiento del video .

Es en este momento cuando los videos se guardan en la videoteca, base de datos del departamento creada utilizando como herramienta la suite informática MS Access ,y es el conjunto de todas las secuencias ya tratadas o editadas del departamento en formato mpeg.

Un miembro del departamento se encarga de dar de alta al video en la videoteca ,para esto accede a la base de datos instalada en un PC y comienza a rellenar un formulario este formulario consta de todos los campos que identifican a un video o secuencia tratada, tales como ,título ,descripción ,etc. Se le asigna el número de master (número de master de cámara del que provienen) y el numero de orden se lo otorga la base de datos de manera automática ,tal como lo actualmente tienen creada esta tabla de registros llamada Cintas de Entrada. Este número de orden no depende del master que provienen ,sino que es correlativo a las secuencias creadas por el departamento.

Al departamento se le encarga elaborar varios DVD sobre la cochera de Horta ,la empresa decide este formato ya que debido a los cambios tecnológicos en la sociedad ,la implantación de este formato y su bajo coste ,es uno de los más usados actualmente ,por lo que será más exportable y arribará a mayor cantidad de usuarios.

El departamento mediante el uso de ordenador y software adecuado crea los DVD usando como fuente de origen el Master de edición donde se encuentran las diversas secuencias tratadas o editadas que se grabaron en la cochera de Horta ,copiando los secuencias en el nuevo DVD .

A este DVD se le asigna un titulo de DVD “Cochera de Horta”,y se le añaden las secuencias ya tratadas o videos en formato mpeg ,con un índice y titulo propio.

**Ejemplo del DVD:**

Cochera de Horta

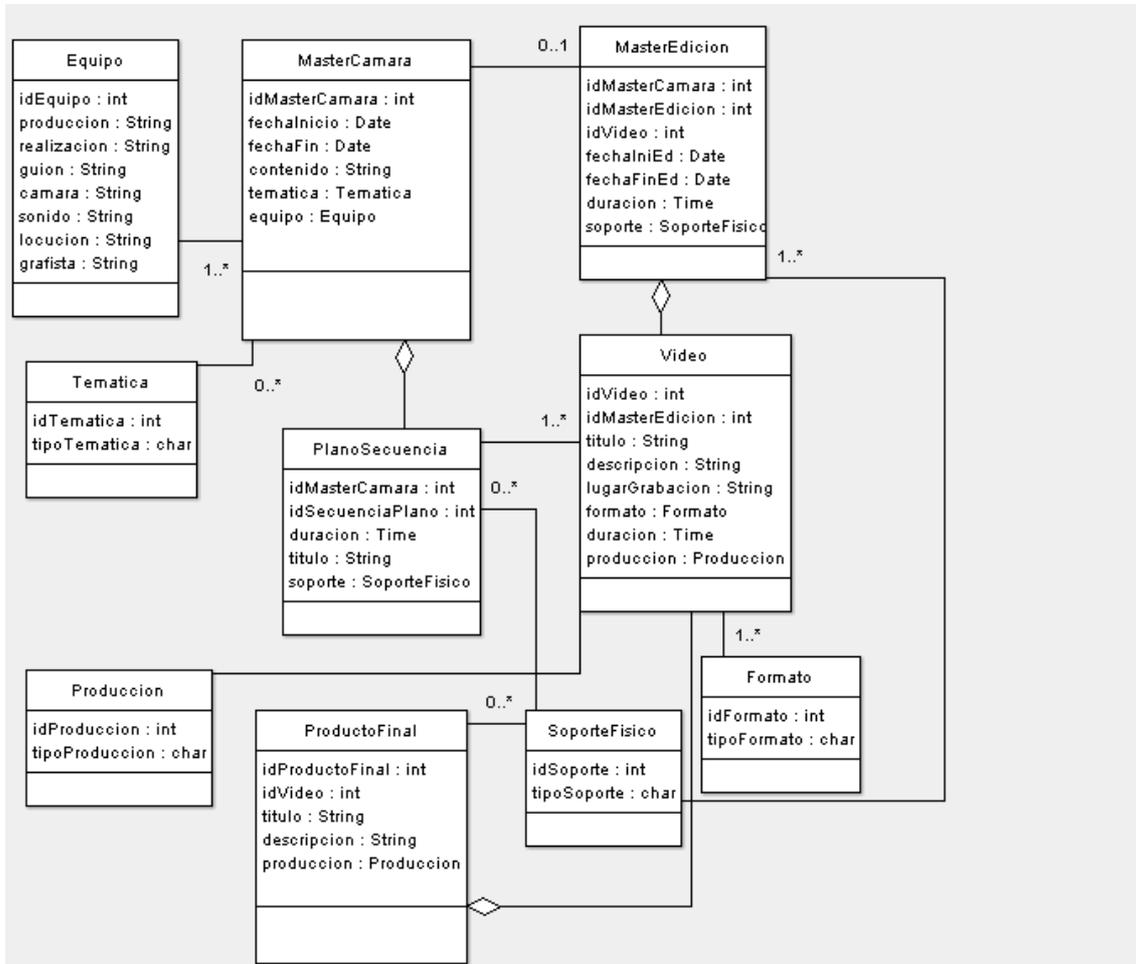
1. sala de conductores
2. Taller Mecánico
3. Oficinas
4. Salas de mandos.

En la carátula o caja de presentación del DVD ,están en titulo del DVD , los títulos de cada video, además del equipo de realización.

Cada uno de estos videos está guardado en la videoteca o base de datos con sus propios atributos identificativos como número de orden ,título ,descripción ,etc.

La cantidad de DVD a crear depende de las necesidades de la empresa o del departamento que los solicita.

Diagrama de Clases:



## Glosario de Términos:

**Master de Cámara:** Suma de los planos secuencia que se obtienen bajo un solo encargo (ejemplo: descarrilamiento L-1, nueva cochera de Horta,...).

**PlanoSecuencia:** Son filmaciones que se realizan en un determinado lugar o momento. En lenguaje técnico son planos o secuencias. Todos los planos secuencia pertenecen a un solo master de cámara.

**Master de Edición:** Puede tener varios Soportes Físicos, en el se guarda el tratamiento que se les da a las secuencias o planos, convirtiéndolas en videos.

**Video:** Es el resultado del tratamiento a una o varias secuencias o planos ,para obtener un documento visual con sonido, imágenes, locución, música.

**Producto Final:** Es un conjunto de videos sobre una temática que se guarda en uno o varios soportes Físicos, formando un producto para ser distribuido.

**Temática:** Temáticas sobre las que tratan los documentos audiovisuales del departamento.

**Equipo:** Grupo de personas que participan en la realización (Filmación) y posterior producción de los documentos audiovisuales.

**Formato:** Tipo de formato de los documentos audiovisuales (avi, mpg, ...).

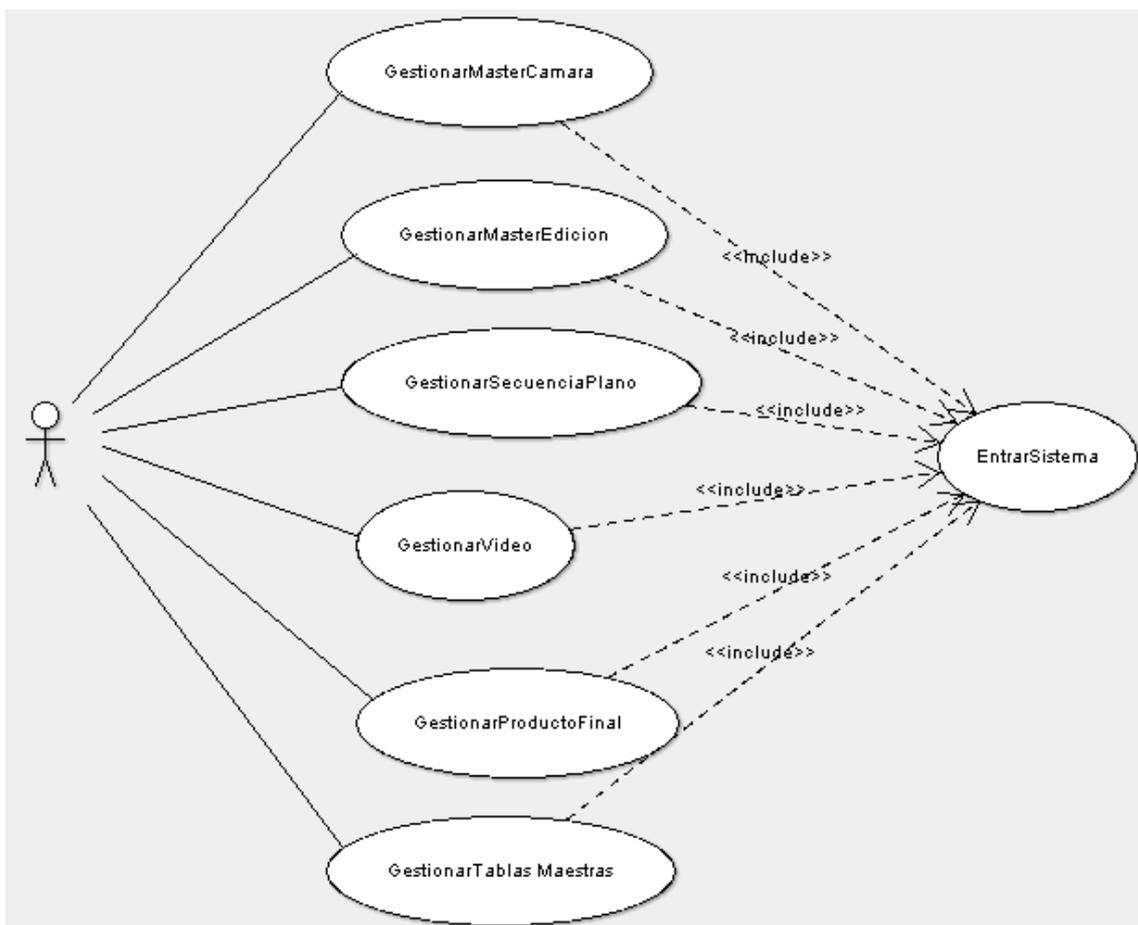
**Soporte Físico:** Tipo de soporte en el que se almacenan los documentos audiovisuales (dvd, vhs, ...).

**Producción:** Identifica el tipo de producción ,interna si el producto lo ha realizado el departamento de audiovisuales de la empresa ,externa si lo ha realizado una empresa exterior

En este diagrama mostramos primeramente esquema generalizado en la descripción textual, detallamos ampliamente todos los casos de usos.

En el plan de trabajo comentaba en la situación de futuro nuestro deseo de ampliar las funcionalidades de la aplicación ,pero debido a la temporalización del Trabajo de fin de carrera en un principio crearíamos funcionalidades a solo en el ámbito del departamento ,una vez cumplidas las fechas límites de las entregas de las diferentes fases del proyecto, lo ampliaríamos con lo cual crearíamos más actores y casos de uso de los que comentamos en este documento.

Diagrama de Casos de Usos:



- Para realizar el análisis se mantuvieron reuniones con el usuario actual y de futuro de la aplicación
- La mayor parte de la memoria se escribió durante el período de análisis
- Para representar en modo gráfico el análisis se utilizó la metodología UML, llevando a cabo diagramas de uso, componentes y clases.

## Objetivos

Inicialmente se propusieron una serie de objetivos que debía alcanzar este TFC además de la consecución del mismo.

### Objetivos alcanzados

- Adquisición de conocimientos teóricos y prácticos en la implantación de aplicaciones reales, usando la arquitectura J2EE y siguiendo un patrón de diseño, en nuestro caso Modelo – Vista – Controlador (MVC).
- Realización de un software que tuviese cabida en un entorno real de explotación.

### Objetivos no resueltos

- Sistema de autenticación controlado por el servidor de aplicaciones.
- Generación de informes de los documentos audiovisuales
- Visualización de los contenidos multimedia en la propia aplicación.

## **Fase Diseño**

## J2EE

J2EE, la plataforma creada por SUN en el año 1997 es según nuestra opinión la que ofrece mejores perspectivas de desarrollo para empresas que quieran basar su arquitectura en productos basados en software libre. J2EE, nos ofrece entre otras las siguientes ventajas:

- Soporte de múltiples sistemas operativos: Al ser una plataforma basada en el lenguaje Java, es posible desarrollar arquitecturas basadas en J2EE utilizando cualquier sistema operativo donde se pueda ejecutar una máquina virtual Java.
- Organismo de control: La plataforma J2EE está controlada por el JCP[1], un organismo formado por más de 500 empresas. Entre las empresas que lo forman están todas las más importantes del mundo informático ( SUN, IBM, Oracle, SAP, HP, AOL, etc. ) lo que garantiza la evolución de la misma.
- Competitividad: Muchas empresas crean soluciones basadas en J2EE y que ofrecen características como rendimiento, precio, etc., muy diferentes. De este modo el cliente tiene una gran cantidad de opciones a elegir.
- Madurez: Creada en el año 1997 como respuesta a la tecnología MTS de Microsoft, J2EE tiene ya cinco años de vida y una gran cantidad de proyectos importantes a sus espaldas.
- Soluciones libres: En la plataforma J2EE es posible crear arquitecturas completas basadas única y exclusivamente en productos de software libre. No sólo eso, sino que los arquitectos normalmente disponen de varias soluciones libres para cada una de las partes de su arquitectura.

Aún así, la plataforma de J2EE también tiene desventajas, algunas importantes:

- Depende de un único lenguaje: La plataforma J2EE depende exclusivamente del lenguaje Java. Sólo se puede utilizar este lenguaje para desarrollar aplicaciones lo que puede suponer un gran problema si nuestro equipo no dispone de los conocimientos suficientes o tiene otras preferencias.
- Complejidad: Aunque no es una plataforma tan compleja como CORBA, no existe un VB .NET en Java. La creación de aplicaciones bajo J2EE requiere normalmente desarrolladores más experimentados que los necesarios para desarrollar bajo .NET
- Heterogeneidad: Existe una gran heterogeneidad en las soluciones de desarrollo. No existe en J2EE un simil a Visual Studio .NET. La gran cantidad de herramientas disponibles causa confusión dentro de los desarrolladores y puede crear dependencias dentro de las empresas.

## **Introducción a J2EE**

Existe mucha confusión, sobre todo entre la gente alejada del mundo de Java, sobre lo que es en realidad J2EE. La confusión más habitual es pensar que J2EE es un producto concreto que distribuye SUN Microsystems, como hace con su JDK, y que te puedes descargar desde su página Web. Nada más lejos de la realidad. No existe un J2EE concreto, no puedes ir a la página Web de SUN Microsystems y descargar "el J2EE".

J2EE es una especificación, un JSR (concretamente el JSR-151[2] ), que define una plataforma de desarrollo empresarial, a la que llamaremos la plataforma J2EE. La plataforma J2EE[3] está formada varios componentes:

- Un conjunto de especificaciones.
- Un test de compatibilidad, el J2EE Compatibility Test Suite ( CTS ).
- La implementación de referencia de J2EE.
- Un conjunto de guías de desarrollo y de prácticas aconsejadas denominadas J2EE BluePrints.

---

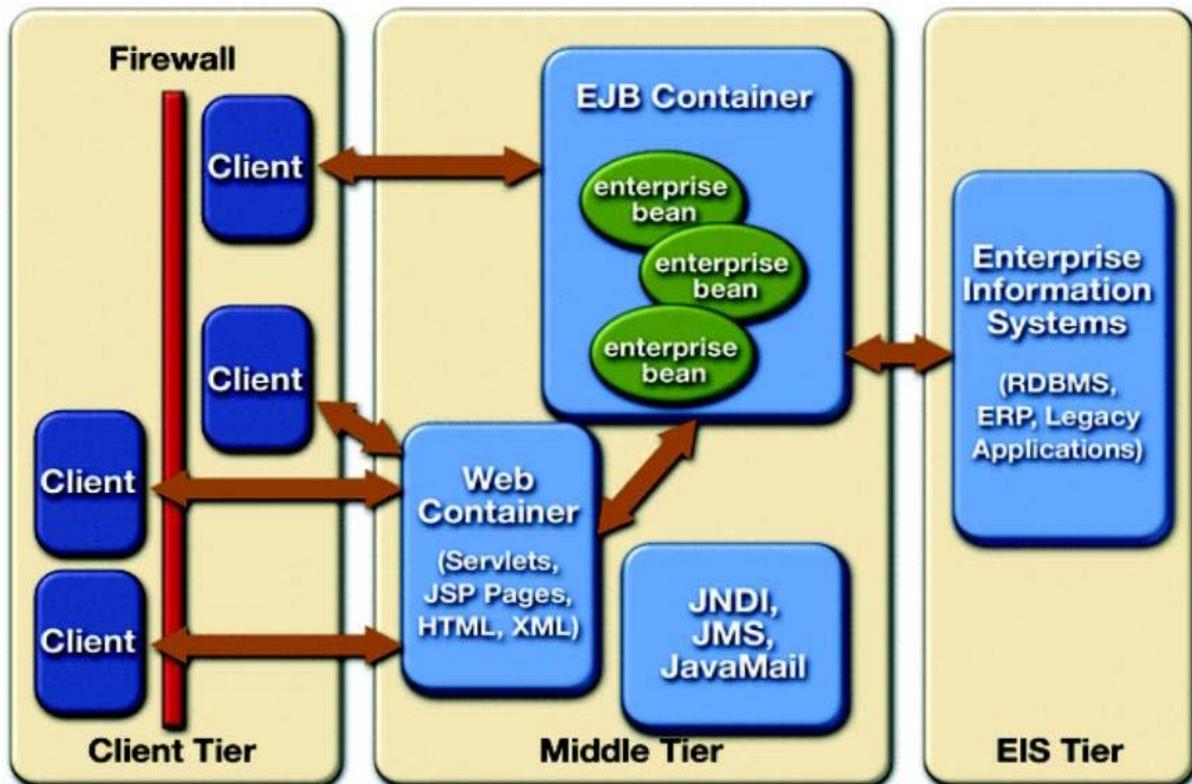
## **El modelo de desarrollo de J2EE**

Para finalizar con esta breve descripción de la plataforma J2EE vamos a ver algunos conceptos básicos sobre el modelo de desarrollo de aplicaciones bajo esta plataforma. Realmente abordar el modelo de desarrollo de J2EE nos llevaría varios libros completos y aún nos dejaríamos partes sin tratar, es por ello que en los siguientes párrafos se va ver por encima como sería una arquitectura típica basada en esta tecnología.

La plataforma de J2EE define un modelo de programación encaminado a la creación de aplicaciones basadas en n-capas. Típicamente una aplicación puede tener cinco capas diferentes:

- Capa de cliente: Representa el interfaz de usuario que maneja el cliente.
- Capa de presentación: Representa el conjunto de componentes que generan el la información que se representará en el interfaz de usuario del cliente. Típicamente se creará a través de componentes basados en Servlets y JSP.
- Capa de lógica de negocio: Contiene nuestros componentes de negocio reutilizables. Normalmente se forma a partir de componentes EJB.
- Capa de integración: Aquí se encuentran componentes que nos permiten hacer más transparente el acceso a la capa de sistemas de información. Por ejemplo este es el lugar idóneo para implementar una lógica de objetos de acceso a datos, DAO (Data Access Objects).
- Capa de sistemas de información: Esta capa engloba a nuestros sistemas de información : bases de datos relacionales, bases de datos orientadas a objetos, sistemas legacy, etc.

Las ventajas de un modelo como este son muy importantes. Al tener las capas separadas tenemos que existe poco acoplamiento entre las mismas, de modo que es mucho más fácil hacer modificaciones en ellas sin que interfieran en las demás. Todo esto redundando en la obtención de mejoras en cuanto a mantenibilidad, extensibilidad y reutilización de componentes. Otra de las ventajas que obtenemos es que se promueve la heterogeneidad de los clientes ya que añadir nuevos tipos de cliente ( móviles, set-top-boxes, PCs, etc.) se reduce a añadir nuevas capas de interfaz de usuario y presentación, sin tener que modificar todo el resto de capas.



En la figura 2 se puede ver lo que sería una posible arquitectura J2EE. El modelo que aparece en la figura está dividido en varias capas, con una separación clara entre presentación, lógica de negocio y sistemas de información empresariales. En este caso además, podemos ver como se trata de un modelo mixto. En la parte de arriba tenemos que se recorre un camino por una estructura de tres capas ( aplicación - lógica de negocio - sistemas de información ), mientras que por el segundo camino recorreremos una estructura de cuatro capas ( aplicación - lógica de presentación - lógica de negocio - sistemas de información ).

Este modelo es sólo un ejemplo, en muchos casos hasta puede que sea demasiado complejo y nos conformemos con utilizar únicamente la capa de Servlets/JSP para acceder a nuestros sistemas de información, en otros casos prescindiremos de la capa Web, en otros casos utilizaremos servicios Web para acceder a la lógica de negocio, o crearemos capas de persistencia intermedia con patrones de acceso a datos. Las combinaciones son prácticamente ilimitadas.

Como ya hemos dicho, el modelo de desarrollo con J2EE está basado en componentes reutilizables, con el objetivo de aumentar la reusabilidad de las aplicaciones. Estos componentes, además, gracias a las especificaciones, son intercambiables entre servidores de aplicaciones, por lo que la portabilidad de nuestros desarrollos es máxima. Si hay un tipo de componentes que requieren una atención especial estos son los Enterprise Java Beans. Se trata de objetos distribuidos, que como hemos dicho contienen la lógica de negocio de nuestras aplicaciones y que hacen transparente al programador operaciones como la persistencia, la seguridad, la gestión de transacciones, etc.

En cuanto a la integración con otros sistemas, gran parte del modelo de J2EE ( en especial los EJB ) está basado en CORBA lo que quiere decir que es posible comunicarse sin ningún tipo de problema a través de IIOP con otras aplicaciones creadas en otros lenguajes diferentes de Java y viceversa. Las posibilidades de integración todavía son mayores si consideramos que a partir de enero del 2003 ( cuando salga J2EE 1.4 ) cualquier EJB que hayamos creado se podrá exponer como un servicio Web. Por último otra tecnología que merece la pena nombrar es la de los conectores, que por explicarlo de una manera clara, son un puente entre J2EE y sistemas legacy ( por ejemplo un conjunto de ficheros de datos en COBOL ), Los conectores tienen un API estándar que nos permite acceder a estos sistemas legacy de una manera transparente y sin apenas esfuerzo.

Para finalizar, otra de las grandes ventajas de J2EE viene del hecho de que está basado en Java, ya que la variedad de clientes que pueden conectarse a una arquitectura J2EE es inmensa, desde aplicaciones de escritorio, hasta móviles, pasando por televisiones, PDAs, etc., cualquiera de estos productos puede conectarse ya sea a través de RMI-IIOP, HTTP, SOAP, XML-RPC, etc., lo que nos da una gran gama de posibilidades en cuanto a la conectividad de nuestros sistemas.

## STRUTS, implementación del patrón MVC en Java

### 1. INTRODUCCIÓN

Este documento describe el patrón de diseño MVC (Model View Controller). Como ejemplo de implementación se utiliza el framework Struts, que facilita el desarrollo de aplicaciones Web en Java basadas en MVC.

En el ámbito del desarrollo Web se siguen unas pautas que tratan más o menos de conseguir un desarrollo estructurado de las aplicaciones, donde la verificación de sesión se centraliza y cada caso de uso se distingue claramente. Utilizando Struts ese "más o menos" se convierte en una arquitectura completamente estructurada que divide perfectamente lógica de negocio (Model), presentación (View) y control de flujo de aplicaciones (Controller). En muchos desarrollos Web se diseña consciente o inconscientemente siguiendo este patrón, por tanto la adopción del modelo Struts no debiera suponer un quebradero de cabeza.

Además hay que tener en cuenta que Struts nos da parte del trabajo hecho, funciona correctamente, y nos da ciertos extras. Teniendo en cuenta además las aportaciones de grupos de desarrollo (taglibs nuevos, etc..), la adopción de esta plataforma puede resultar muy interesante.

Parte del éxito de Struts se debe a que el desarrollo de aplicaciones Web resulta ser un caos y cada grupo de desarrollo lo hace a su manera. Al menos para entorno puramente Web Struts le está comiendo terreno a los EJB y está consiguiendo que los desarrollos sigan unas pautas estándar, lo que facilita que en el cambiante mundo de la ingeniería informática donde impera el movimiento de recursos humanos el mantenimiento y mejora de una aplicación Web sea más llevadero.

### 2. MVC: MODEL VIEW CONTROLLER

MVC o Model view Controller es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software. El paradigma MVC consiste en dividir las aplicaciones en tres partes:

- Controlador
- Modelo
- Vistas.

El **controlador** es el encargado de redirigir o asignar una aplicación (un modelo) a cada petición; el controlador debe poseer de algún modo, un "mapa" de correspondencias entre peticiones y respuestas (aplicaciones o modelo) que se les asignan.

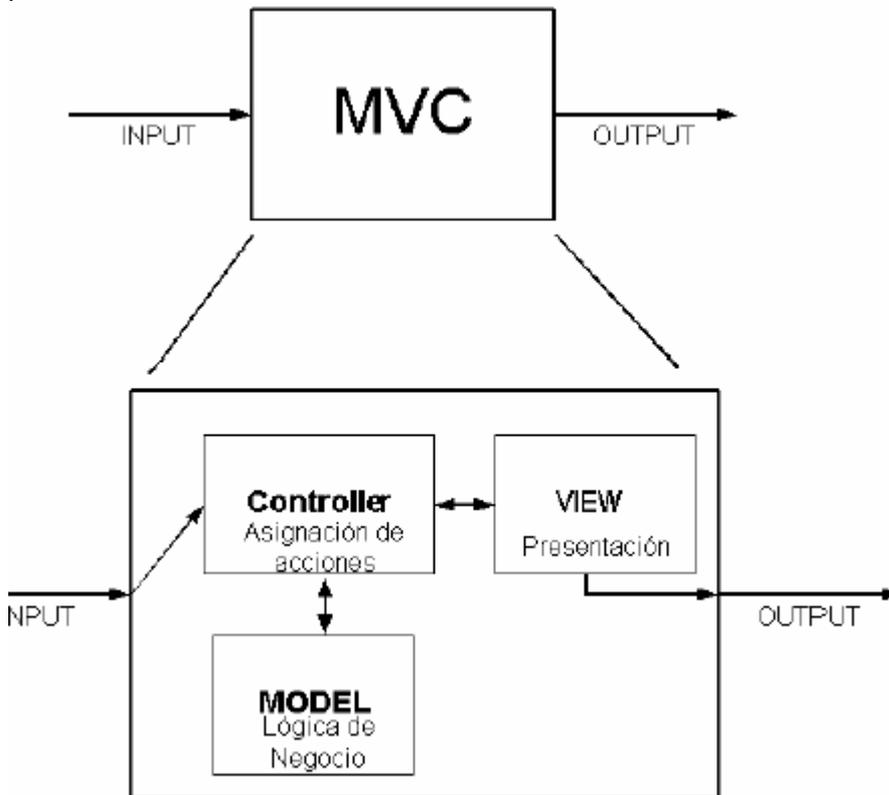
El **modelo** sería la aplicación que responde a una petición, es la lógica de negocio a fin de cuentas.

Una vez realizadas las operaciones necesarias el flujo vuelve al controlador y este devuelve los resultados a una **vista** asignada.

Vemos las diferencias que supone el modelo con los modelos convencionales.

Volviendo al esquema más básico de programa, tenemos una entrada o parámetros que llegan (INPUT), se procesan y se muestra el resultado (OUTPUT).

En el caso del patrón MVC el procesamiento se lleva a cabo entre sus tres componentes. El controller recibe una orden y decide quien la lleva a cabo en el modelo. Una vez que el modelo (la lógica de negocio) termina sus operaciones devuelve el flujo vuelve al controller y este envía el resultado a la capa de presentación.



El Controller en cierta forma debe tener un registro de la relación entre ordenes que le pueden llegar y la lógica de negocio que le corresponde (Es como una operadora de teléfono que recibe una petición y une dos líneas). En el siguiente gráfico se representa ese funcionamiento:

¿Que ventajas obtenemos de este modelo? Obviamente una separación total entre lógica de negocio y presentación. A esto se le pueden aplicar opciones como el multilinguaje, distintos diseños de presentación,.. etc. sin alterar la lógica de negocio. La separación de capas como presentación, lógica de negocio, acceso a datos es fundamental para el desarrollo de arquitecturas consistentes, reutilizables y más fácilmente mantenibles, lo que al final resulta en un ahorro de tiempo en desarrollo en posteriores proyectos.

- El diseño de la aplicación se basó en el modelo de 3 capas:
  - Presentación  
Para la capa de presentación se pensó en un cliente ligero y fácilmente accesible desde cualquier centro de TMB o fuera de TMB, de cara a facilitar a otros usuarios la consulta de datos en la aplicación
  - Negocio  
La lógica de negocio se ejecuta en el servidor de aplicaciones, mediante componentes ejb (enterprise java bean)
  - Datos o persistencia  
Se dividió la persistencia en dos subcapas: la persistencia no volátil (base de datos relacional) y la volátil en el servidor de aplicaciones (mapeo O-R) para mejorar la eficiencia

#### Persistencia

- La persistencia de datos se realizó en base de datos relacional. Se probó con MySQL e Informix.
- La persistencia temporal en servidor se realizó con Hibernate.
- Para la modelización de datos se optó por la herramienta Sybase PowerDesigner.

#### Negocio

- La lógica de negocio se encuentra en diez entidades y existen dos sesiones que envuelven 5 entidades cada una, actuando como fachada.

#### Presentación

Para la capa de presentación implementamos el patrón de diseño MVC mediante el framework Struts 1.2:

- Se consensuó con el usuario que la aplicación fuese Web, por lo que cliente era el navegador. Antiguamente eran formularios Access
- Se estudió con el usuario todas las pantallas que debía tener la aplicación, así como el flujo de navegación entre éstas.
- Esto permite que sea visualmente atractiva y cómoda
- Se adapta al estilo corporativo

## *Capa de presentación*

Requisitos de la interficie de usuario:

El objetivo de la aplicación era que fuese lo mas sencilla posible, cómoda y de fácil aprendizaje.

En un principio tenía pensado unas ideas de pantalla concretas siguiendo los criterios que comentabamos anteriormente pero al conocer la librería Displaystags que se rige por la idea de presentación que tenía y la necesaria para la mayoría de usuarios, decidí acometer pequeños cambios en la idea de presentación.

La librería Displaytags, junto con Struts nos facilitan la creación de este tipo de interficies, proporcionándole al usuario tener todas las posibles funciones de la aplicación a la vista de una manera muy clara, es decir creando interficies totalmente WYSWYG (“what you see ,what you get”) lo que ves , lo que obtienes.

A continuación presentamos las pantallas y los ligeros cambios acometidos en referencia a la idea anterior de pantallas.

En un principio teníamos la idea de la pantalla inicial tal como muestra la figura de la pagina siguiente:

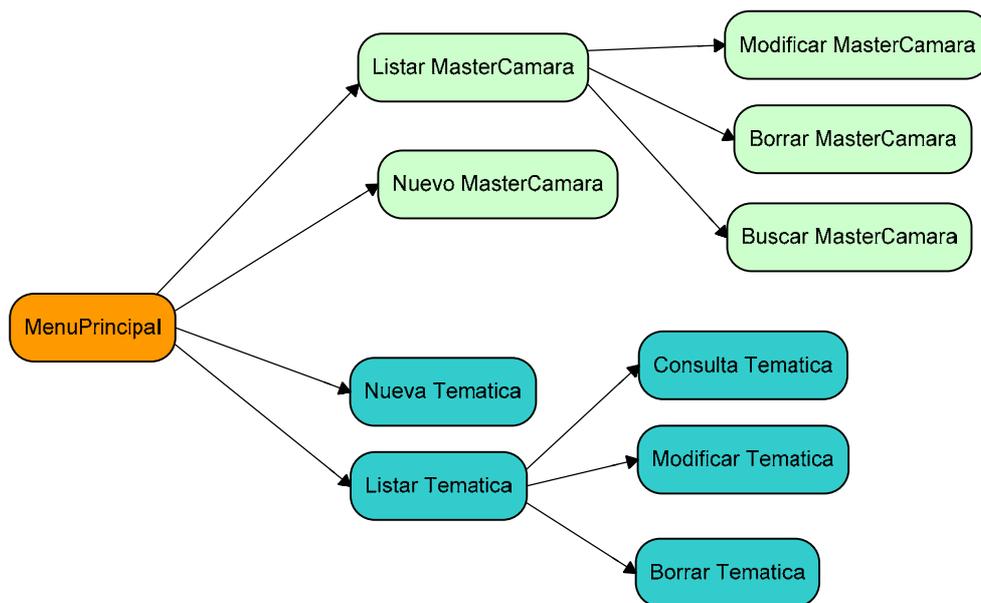
Diagrama grafico de las funciones de pantallas



Una vez conocida la librería Displaystags junto con Struts ,me costó poco decidirme por usar ambas, como podemos observar en la figura siguiente ; nos muestra ya en una sola pantalla dos de las funciones más importantes Listar y Nuevo ,nuestro objetivo es, si la temporalización nos lo permite agregar una columna al Menú Principal con la función Buscar ,ya que esta librería nos ofrece un método de busca que no me convence ,una vez listado una colección de objetos la búsqueda es visual dentro de la lista, aunque este método es relativamente cómodo y muy intuitivo, no es práctico cuando se tienen grandes colecciones de objetos ya que deberíamos ir buscando en paginas de listas, hasta ver el que nos interesa, como es lógico opto por el método de implementar una función buscar en la cual una vez presionado nos muestra un formulario (que ya tenemos implementados, en las funciones Nuevo , Modificar y Borrar) en la que rellenamos los campos que nos interesen en la búsqueda.

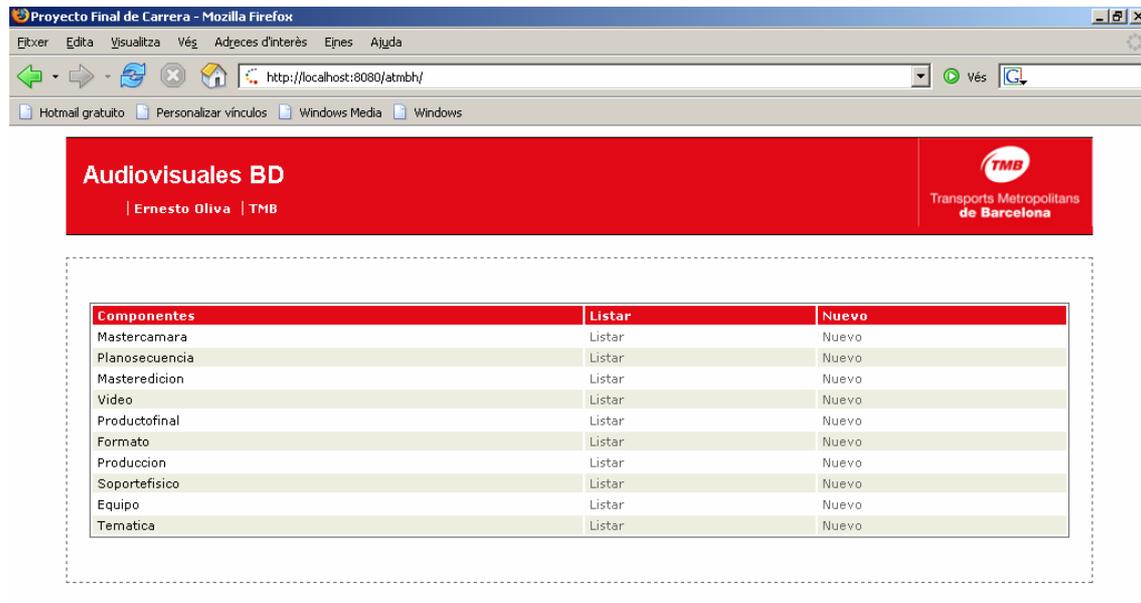
Una vez decidido el uso de esta librería optamos por realizar un ligero cambio de funciones de pantallas como muestra la figura siguiente:

Diagrama grafico reducido de las funciones de pantallas



## Diseño de la Interficie Menú Principal

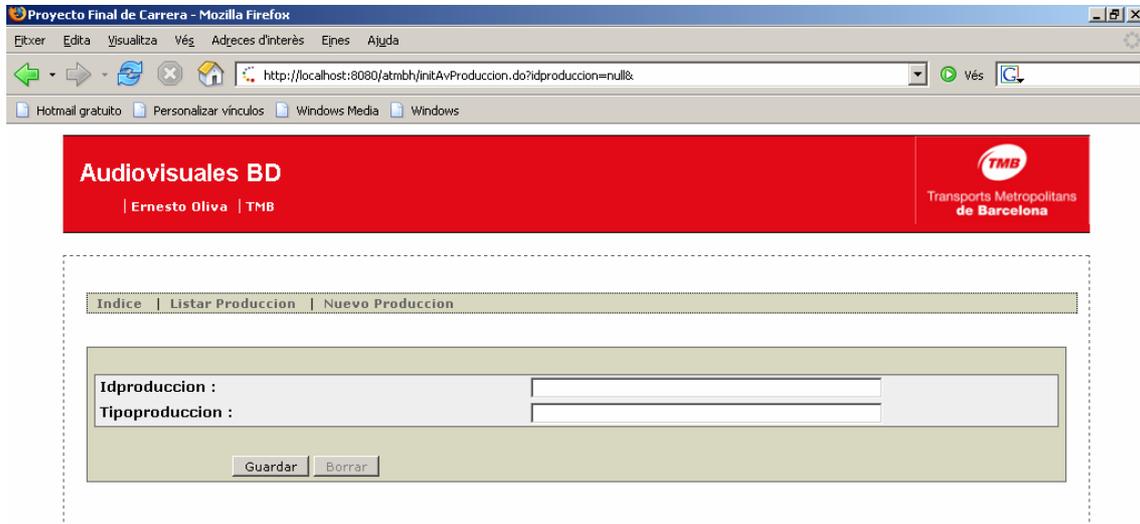
En esta pantalla Menú Principal ya podemos listar todas los Objetos que tenemos en cada tabla y podemos crear estos, todo esto con un golf de ejecución mínimo es decir un solo clic.



*Audiovisuales Comunicacion Interna TMB*

## Diseño de la Interficie Nuevo

En la figura siguiente mostramos un ejemplo del Diseño de la pantalla Nueva Produccion para mostrar la funcionalidad Nuevo:

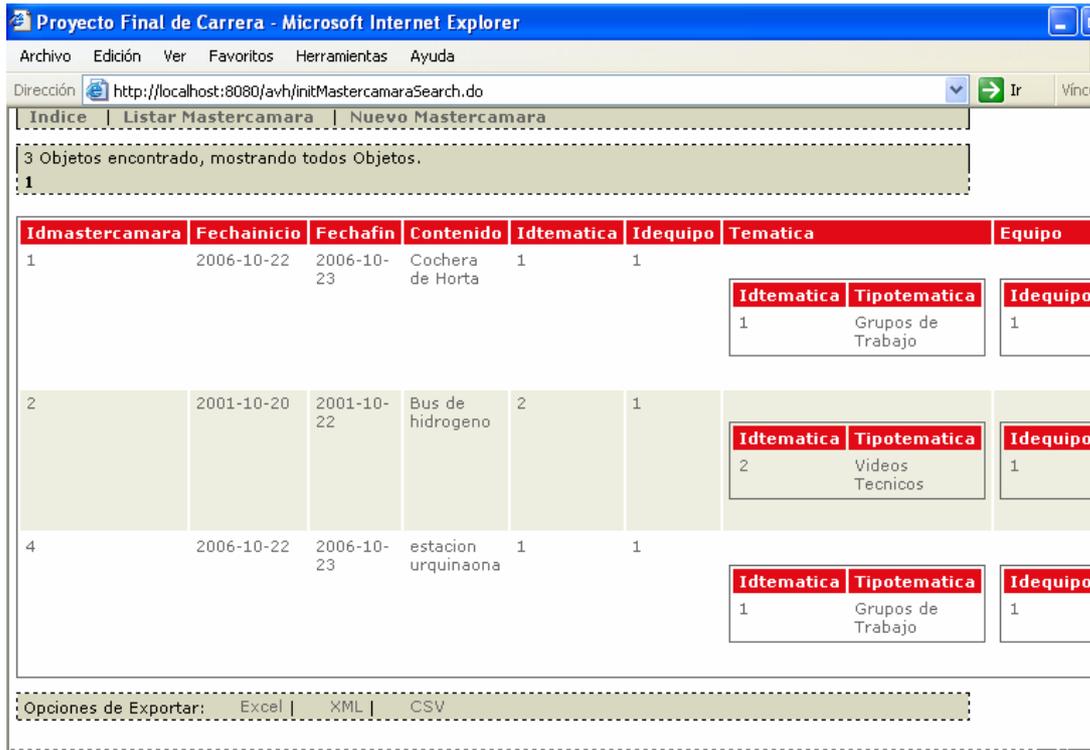


*Audiovisuales Comunicacion Interna TMB*

Como podemos observar tenemos en esta pantalla la posibilidad de volver al menú principal o Índice y la de Listar las producciones existentes.

## Diseño de la Interficie Listar:

En un modelo WYSWYG esta interficie es una de las importantes ya que está incluida en la mayoría de funciones o lo que es lo mismo para realizar una función primero se lista ,a partir de la lista podemos seleccionar los objetos que necesitemos y ejecutar la acción pertinente, este método aún siendo muy cómodo e intuitivo tiene sus desventajas cuando las colecciones de objetos son de volúmenes considerables ya que para acceder a estos tendríamos que recorrer varias páginas.



Como podemos observar una entidad que tenga un atributo <<type>> de otra entidad nos mostrará en el campo toda la colección de objetos que exista.

Todos los atributos de la entidad poseen un link mediante el cual se realiza la búsqueda o findby[atributo clicado].

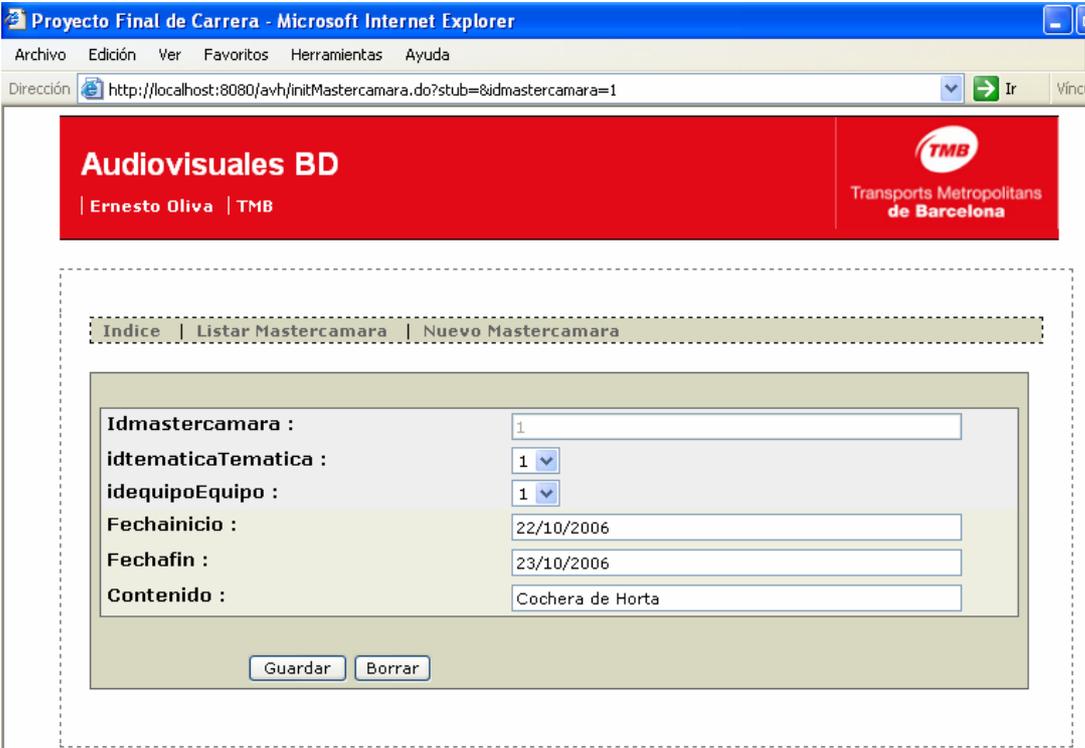
Tenemos a pie de pagina varias opciones de exportar el resultado obtenido a varios formatos como Excel, XML o CSV.

Clicando cualquier campo de la tabla podemos acceder a dicho objeto con la opción de Modificarlo o Borrarlo.

## Diseño de la Interficie Modificar y Borrar:

Para el diseño de estas Interficies usamos la misma composición de la interficie Nuevo pero en este caso la acción cambia ,es decir activamos, en el caso de borrar ,según la acción predecesora activamos la acción Borrar en el botón.

En el caso de Modificar también varia la acción según la acción predecesora.



The screenshot shows a web browser window titled 'Proyecto Final de Carrera - Microsoft Internet Explorer'. The address bar contains the URL 'http://localhost:8080/avh/initMastercamara.do?stub=&idmastercamara=1'. The page header is red and contains the text 'Audiovisuales BD' and 'Ernesto Oliva | TMB'. The TMB logo and 'Transports Metropolitans de Barcelona' are also visible. Below the header is a navigation bar with links: 'Indice | Listar Mastercamara | Nuevo Mastercamara'. The main content area is a form with the following fields:

Idmastercamara :	<input type="text" value="1"/>
idtematicaTematica :	<input type="text" value="1"/>
idequipoEquipo :	<input type="text" value="1"/>
Fechainicio :	<input type="text" value="22/10/2006"/>
Fechafin :	<input type="text" value="23/10/2006"/>
Contenido :	<input type="text" value="Cochera de Horta"/>

At the bottom of the form are two buttons: 'Guardar' and 'Borrar'.

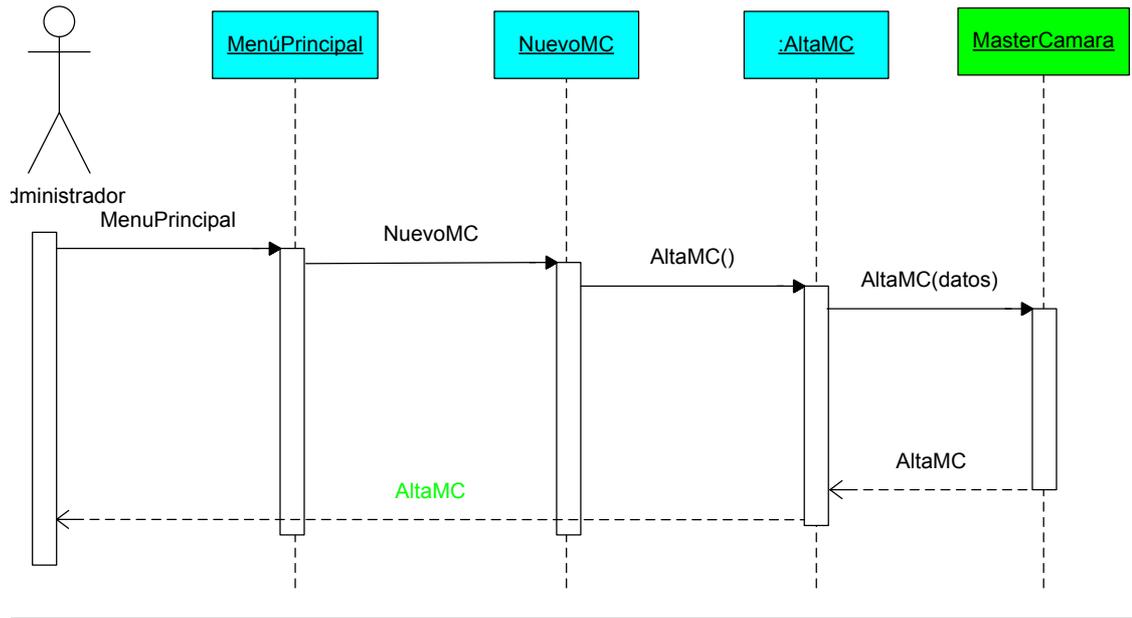
Cuando clicamos un campo de la tabla que muestra la lista de la pantalla Listar MasterCamara ,en nuestro caso hemos clicado “Cochera de Horta” la aplicación busca ejecutando la accion findbycontenido,de esta manera accedemos al Master Camara, una vez aquí podemos modificarlo o borrarlo ,tal como se muestran los botones activados, Guardar y Borrar.

Como podemos observar desde estas pantallas podemos volver al Menú Principal o Índice y a listar los objetos de la entidad en la cual estamos trabajando.

## **Diagramas de Secuencias**

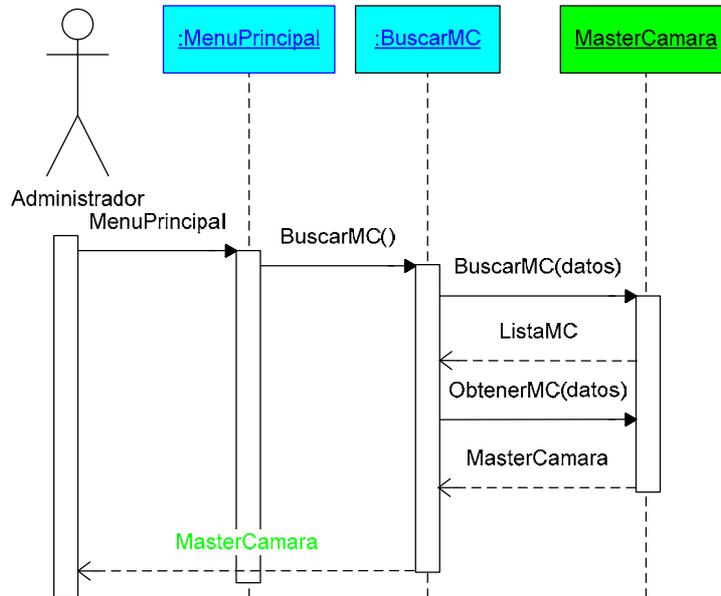
## 1-Alta Master de Cámara

### Caso de Uso : Alta MC



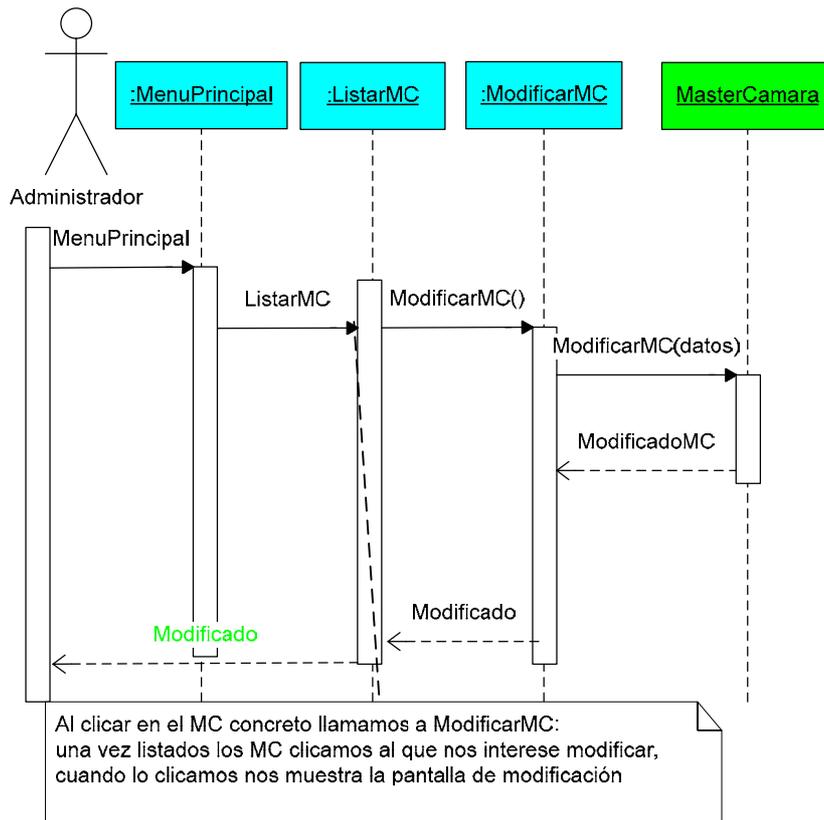
## 2- Consultar Master de Cámara

Caso de Uso : Consultar Master de Cámara



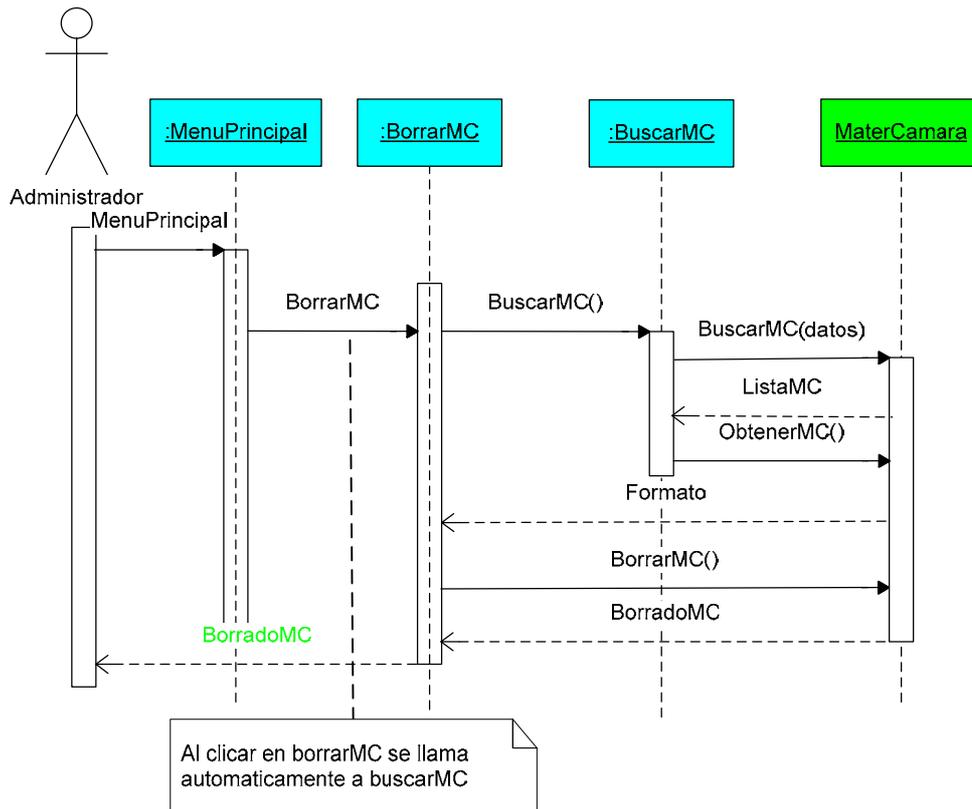
### 3 - Modificar Master de Cámara

Caso de Uso : Modificar Master de Cámara



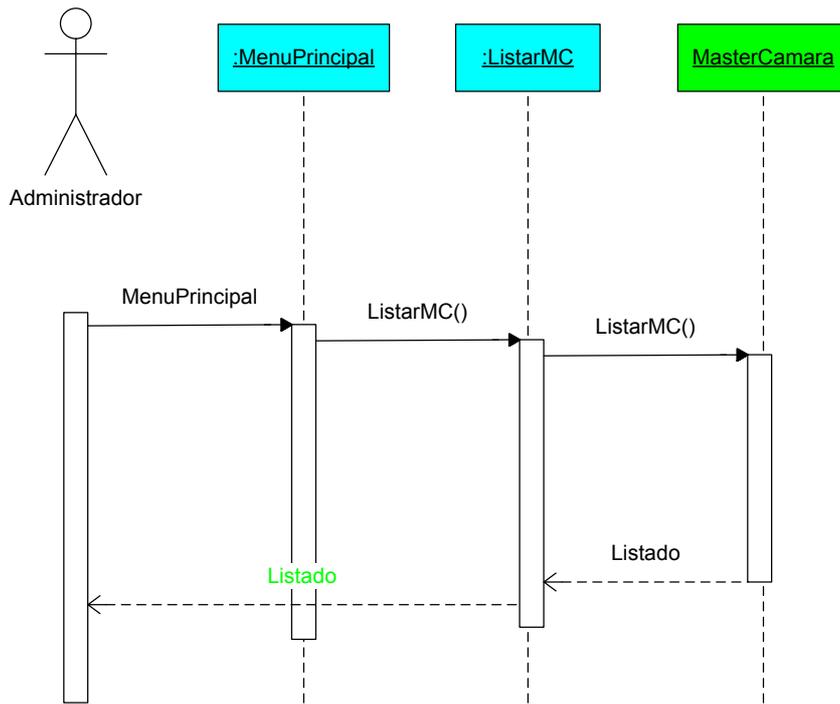
## 4 - Borrar Master de Cámara

### Caso de Uso : Borrar Master de Cámara



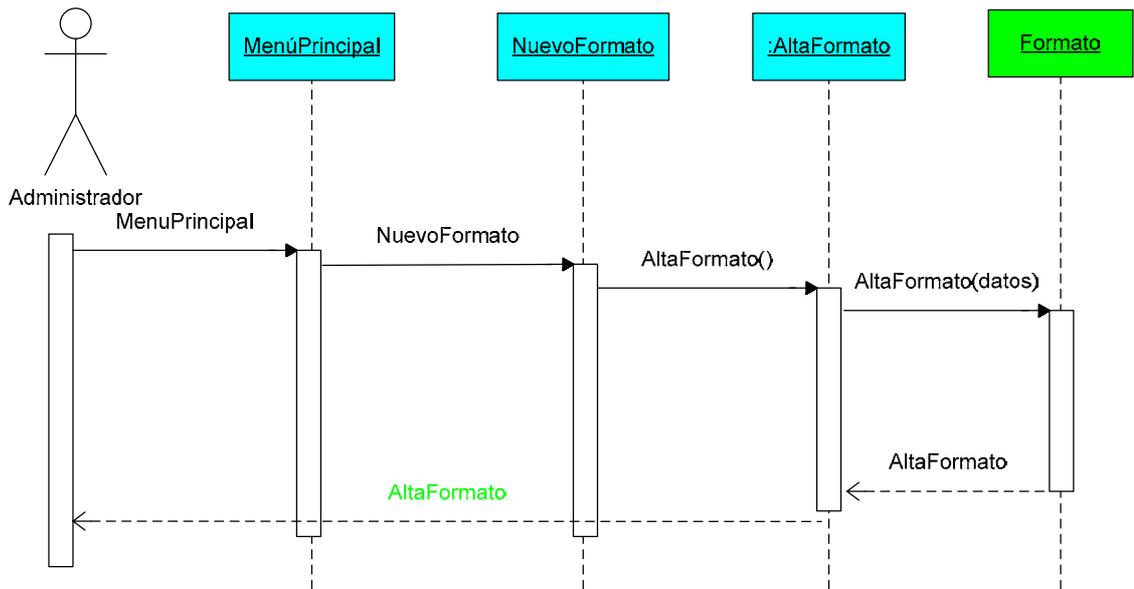
5 - Listar Master de Cámara

Caso de Uso : Listar Master de Cámara



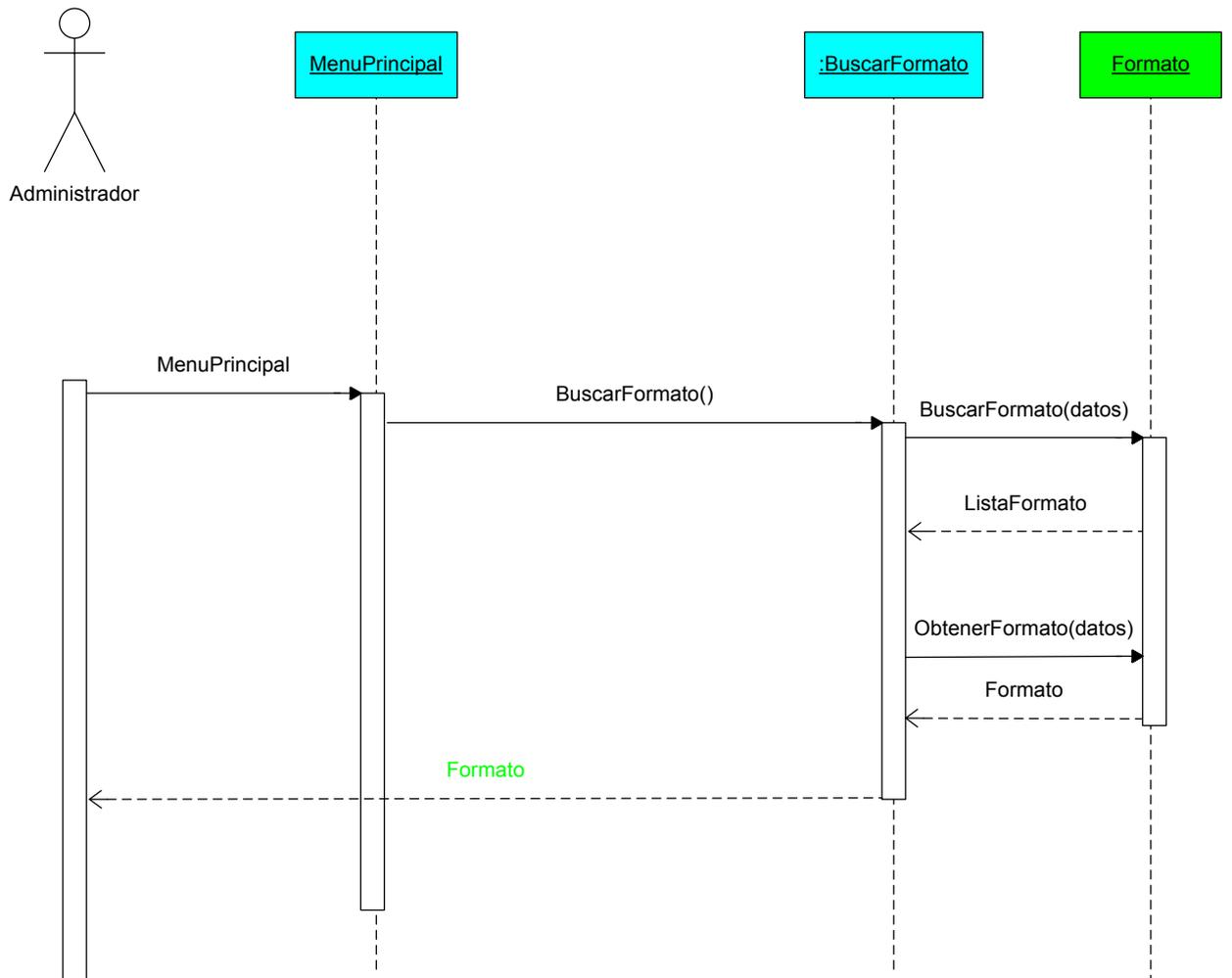
6 - Alta Formato

Caso de Uso : Alta Formato



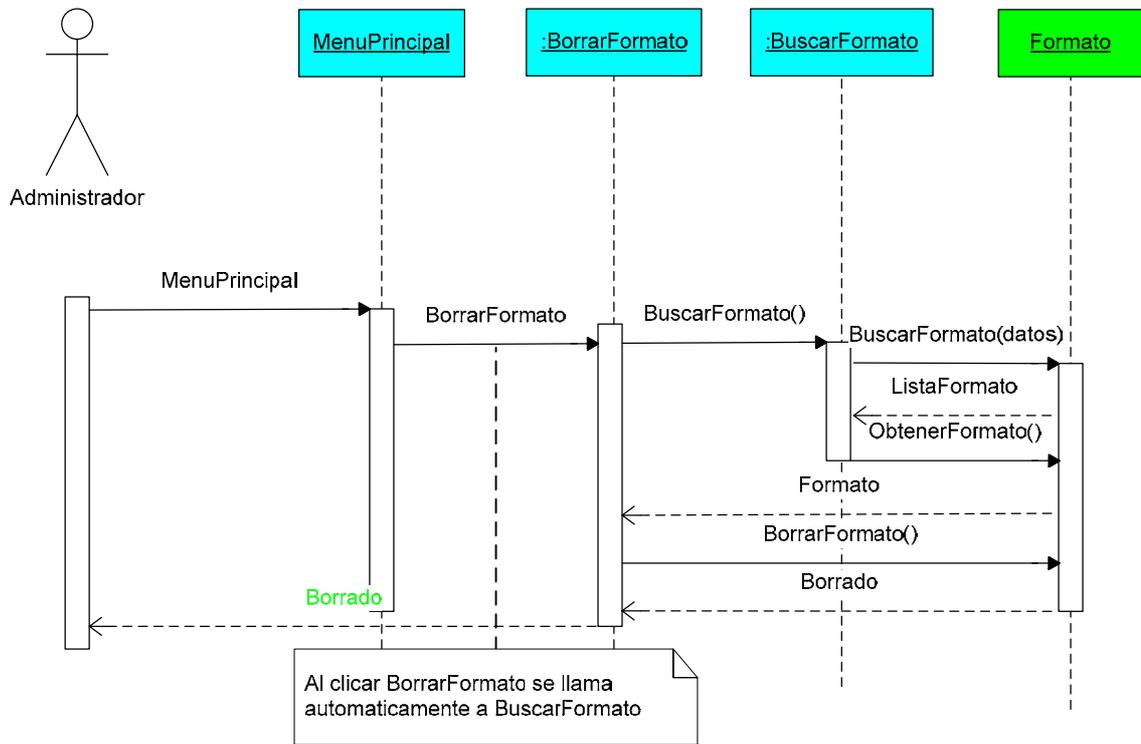
7 - Consulta Formato

Caso de Uso : Consulta Formato



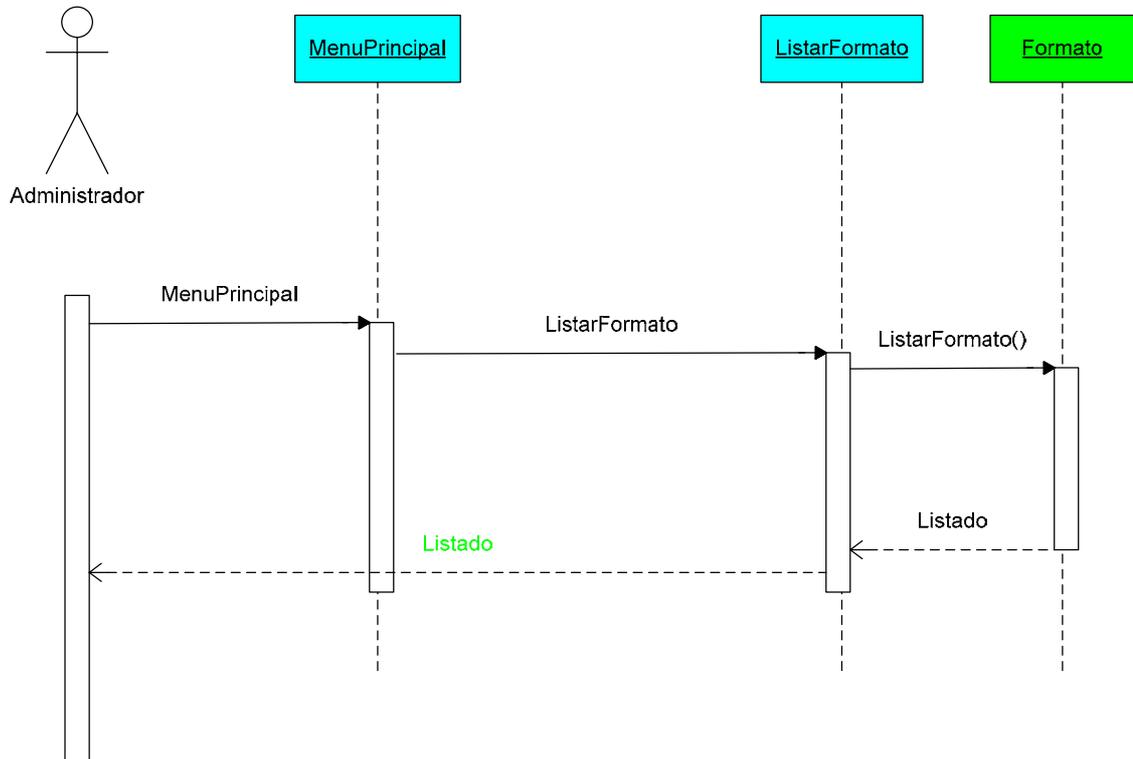
8 - *Borrar Formato*

Caso de Uso : Borrar Formato



9 - Listar Formato

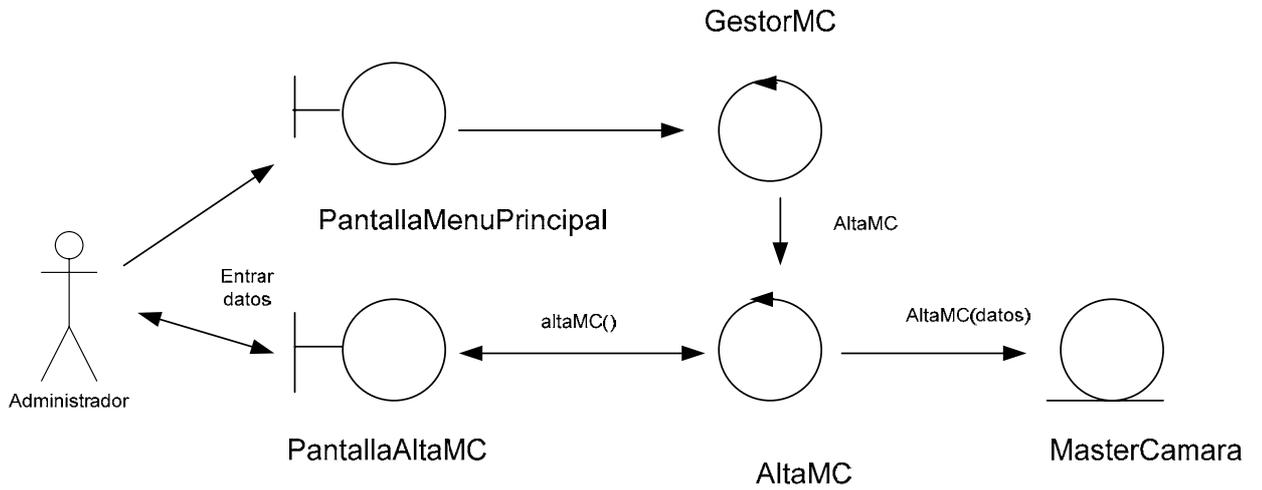
Caso de Uso : Listar Formato



## **Diagramas de Colaboración simplificados**

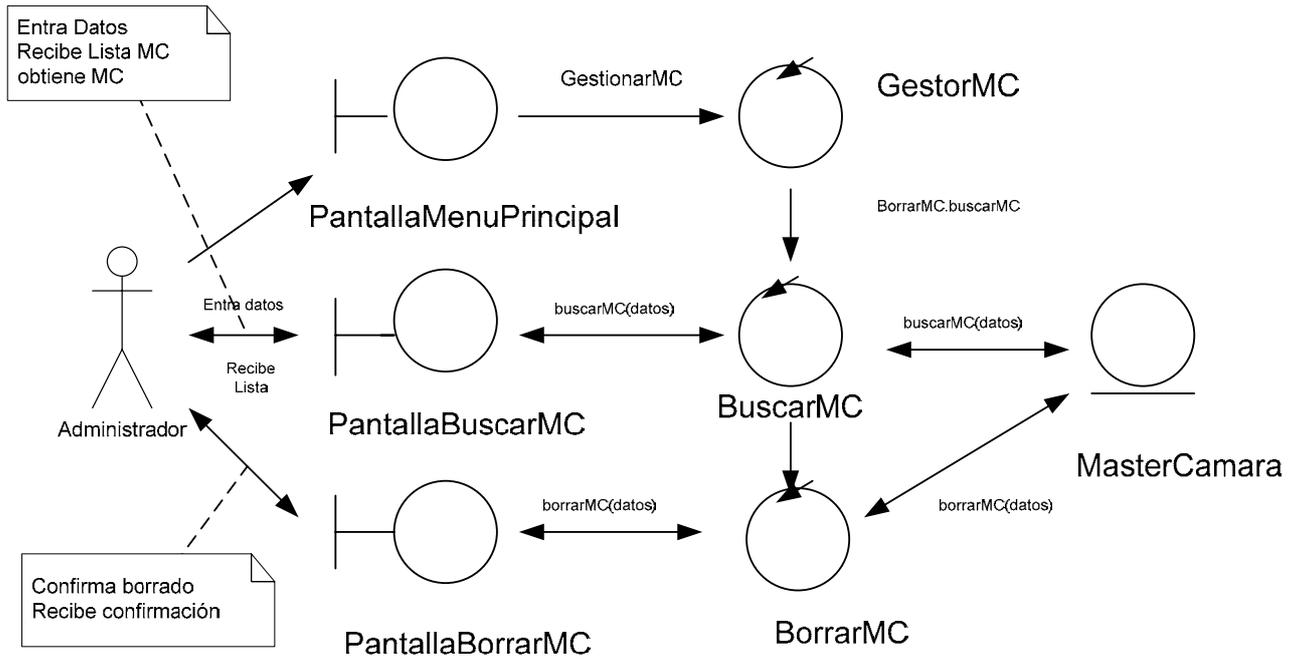
## 1 - Alta Master de Cámara

Caso de Uso : Alta Master de Cámara



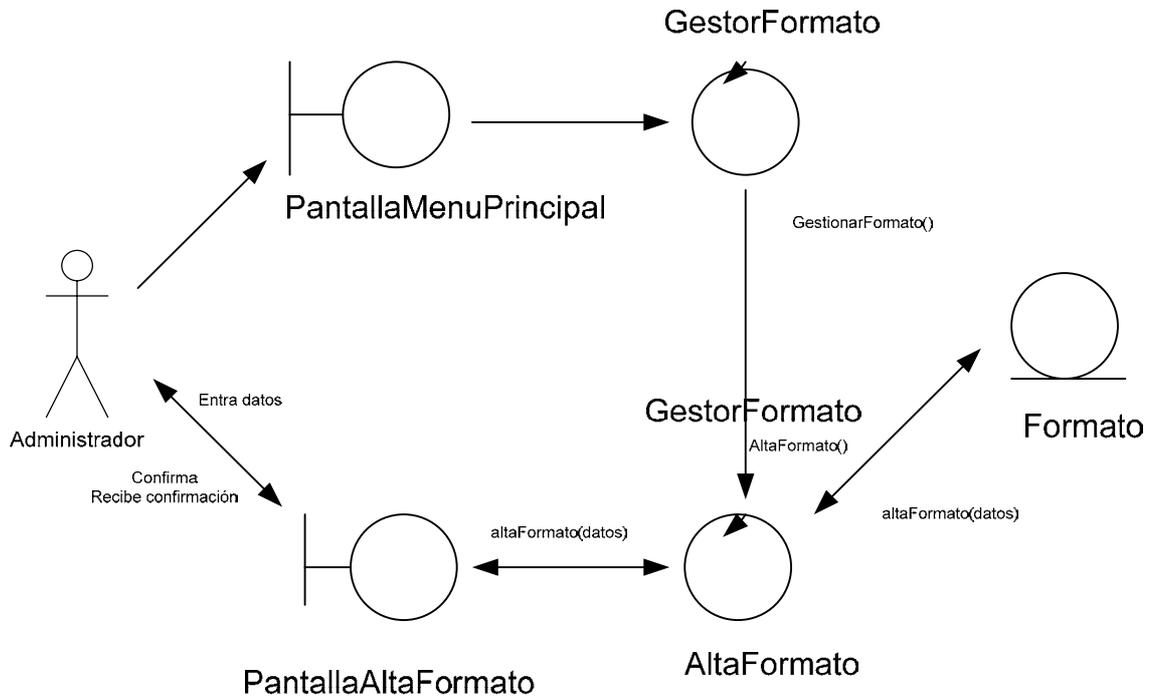
## 2-Borrar Master de Cámara

Caso de Uso Borrar Master de Cámara



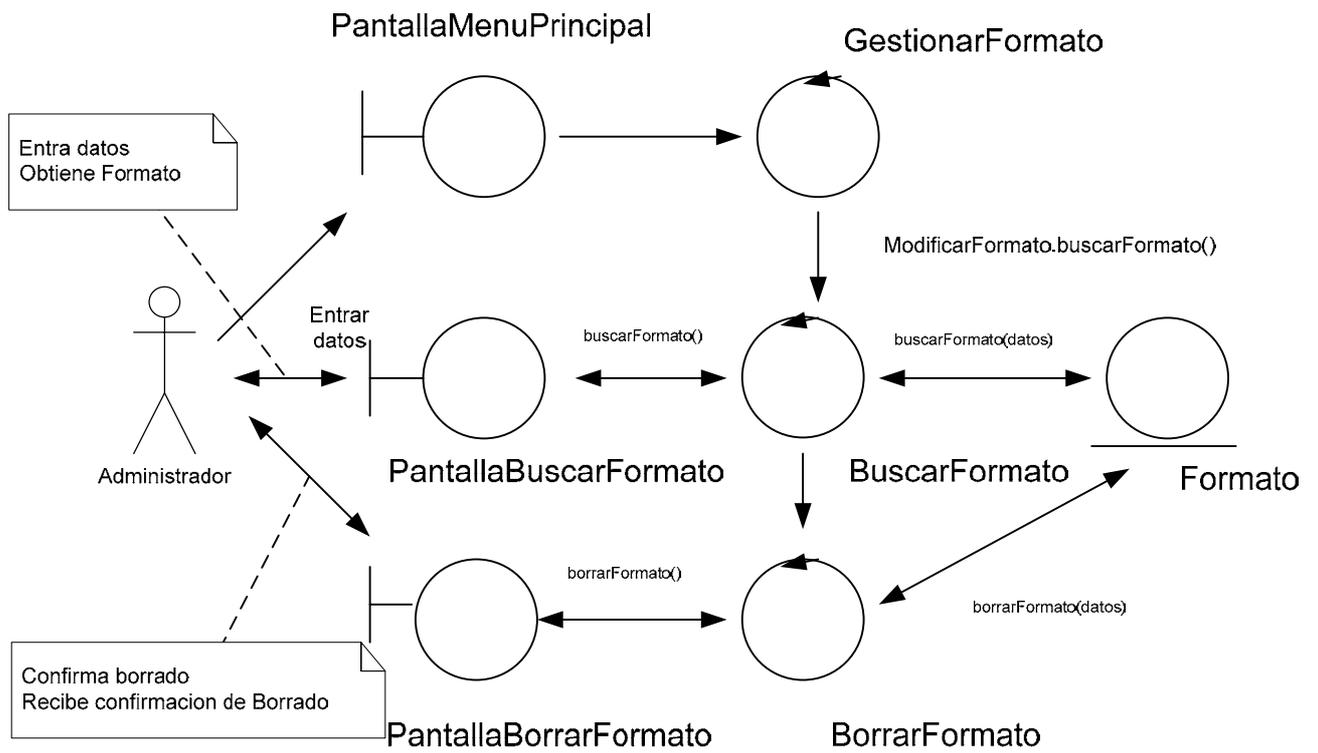
### 3-Alta Formato

Caso de Uso : Alta Formato

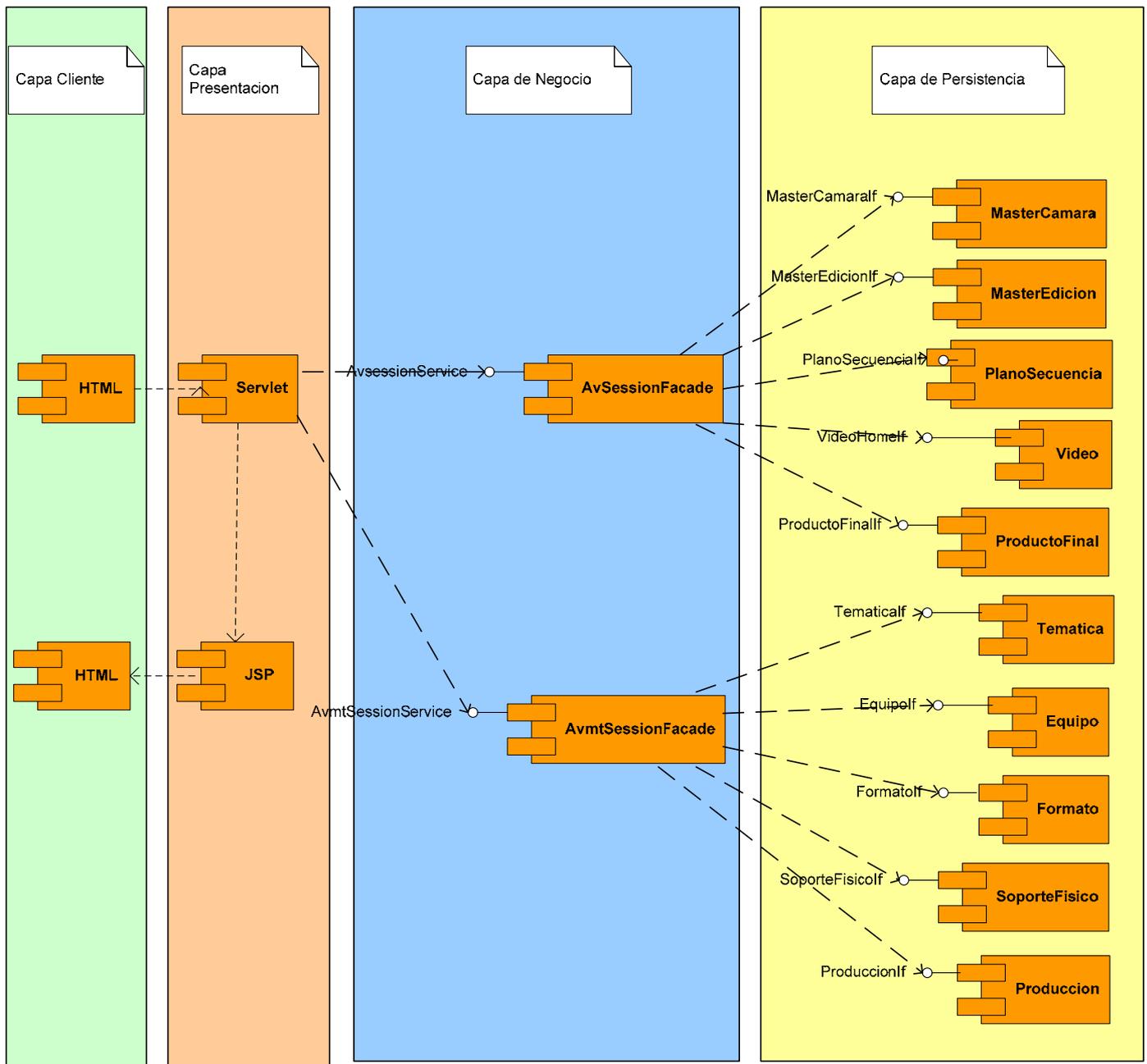


### 4-Borrar Formato

#### Caso de Uso Borrar Formato

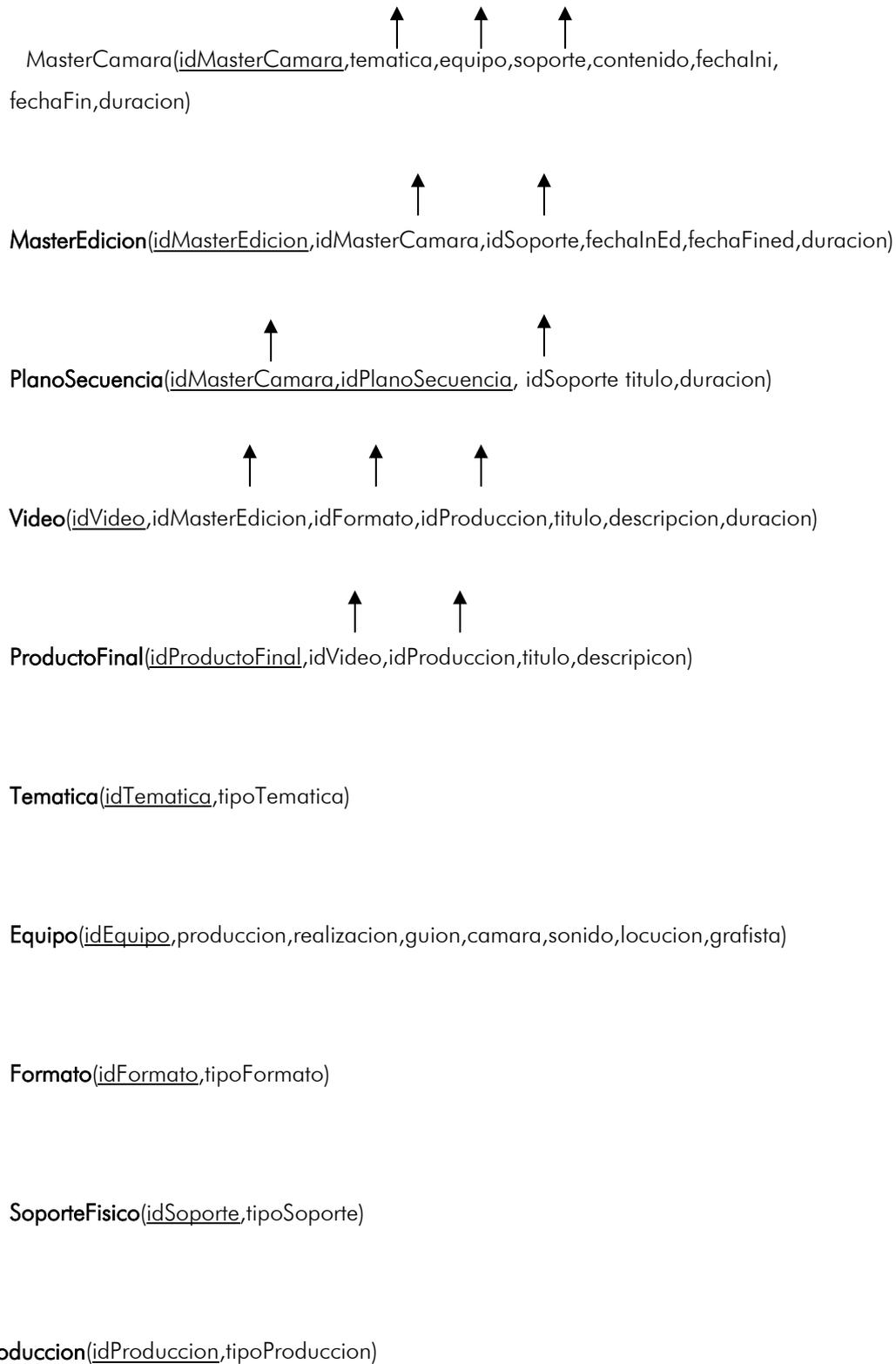


## Diagramas de Componentes



## **Base de Datos**

*Modelo Lógico*



### *SGBD UTILIZADO*

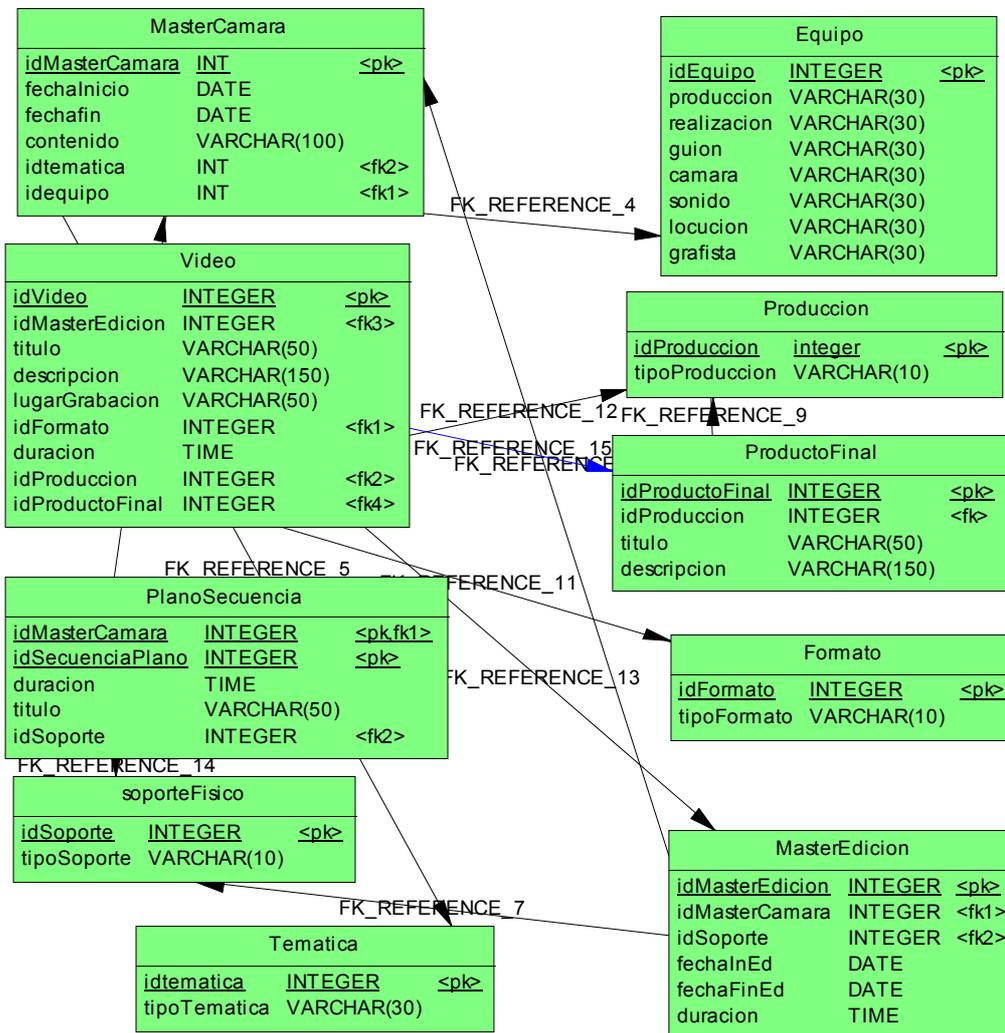
Para implementar la persistencia de datos se ha elegido el *SGBD* (Sistema Gestor de Bases de Datos) *MySQL Server 4.x*. Este *SGBD* reúne todos los requisitos necesarios para la implementación del proyecto. Algunas de estas características son:

- Integridad referencial
- Control de transacciones (*BEGIN*, *COMMIT* y *ROLLBACK*)
- Surrogates (claves sustitutivas)
- Todo aquello soportado por *SQL Standard*

Ver Anexo-Scripts-sql

Diseño de la Persistencia

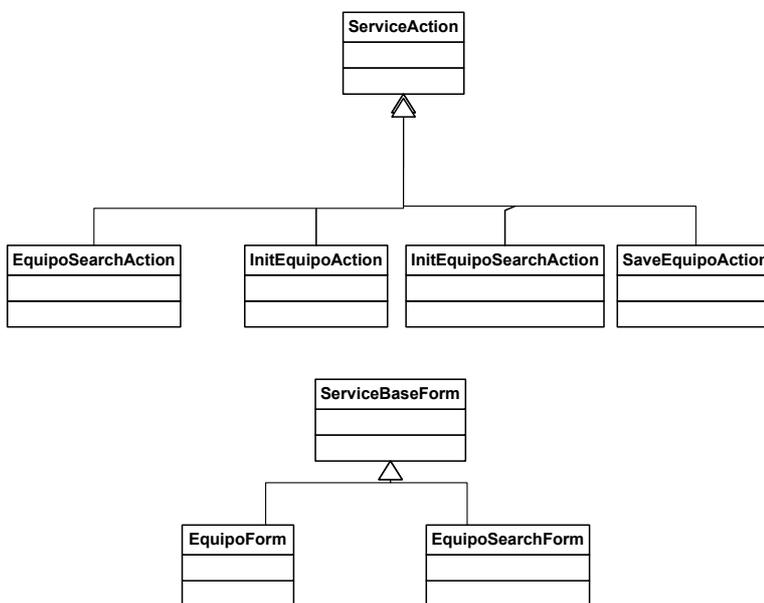
Modelo de Datos:



## Implementación

- La plataforma utilizada para implementar la aplicación es Java. La especificación dentro de la tecnología Java ha sido J2EE, que está pensada para cubrir el ciclo de vida del software a nivel empresarial.
- La especificación J2EE es una tecnología abierta basada en Java inicialmente desarrollada por Sun Microsystems.
- El modelo de desarrollo escogido dentro de J2EE es el MVC (Modelo Vista Controlador).  
MVC o Model view Controller es un patrón de diseño aportado originariamente por el lenguaje SmallTalk a la Ingeniería del Software. El paradigma MVC consiste en dividir las aplicaciones en tres partes (Controlador, Vista, Modelo).
- El software está pensado para trabajar con:
  - El servidor de aplicaciones Jboss
  - Los SGBD MySQL e Informix.
- La mayoría del software utilizado es de código abierto por lo que el coste del desarrollo de la aplicación es muy reducido.

- Struts es un framework que implementa el patrón de diseño MVC.
- De cada objeto:
  - Se ha implementado una clase Action(En nuestro caso delegamos a varias clases) que representará la acción propia .
  - Una clase Form que representa el formulario con los setter y getters de cada atributo además de los metodos getList y getListsize().para obtener una coleccion del objeto y obtener el numero de objetos y otra clase para la búsqueda.



En Struts necesitamos lo siguiente:

1. -Una clase Action (nuestra clase comienza por init) encargada de cargar el formulario.
2. -Un Bean tipo Form cuyos campos u atributos de instancia coinciden con los campos del formulario. (al ser tipo JavaBean tendrá los correspondientes métodos set y get para los atributos) . Este form y sus atributos son un requisito imprescindible.
3. -Una JSP que contiene el formulario citado. Los campos deben coincidir con los definidos en el Bean de formulario
4. -Una clase Action (su nombre podría empezar por **Save**) encargada de pasar la instancia del formulario al Bean de Logica de Negocio.
5. -Un Bean de lógica de negocio (El que hace algo con los datos del formulario) nuestros sessions.

Fragmento de **ServiceAction** clase abstracta que define dos metodos statics para obtener el session que corresponda :

```
public static AvsessionService getAvsessionService() {
    try {
        return (AvsessionService)
ServiceLocator.getService(ServiceLocator.AVSESSION_SERVICE);
    } catch (Exception e) {
        return null;
    }
}

/**
 * Obtiene la interficie session.
 * @return AvmtsessionService interface
 */
public static AvmtsessionService getAvmtsessionService() {
    try {
        return (AvmtsessionService)
ServiceLocator.getService(ServiceLocator.AVMTSESSION_SERVICE);
    } catch (Exception e) {
        return null;
    }
}
```

Al clicar en la vista **index** sobre listar estamos llamando la accion `init[entidad]search`

```
<a href='initMastercamaraSearch.do'>Listar</a>
</TD>
<TD>
    <a href='initMastercamara.do?idmastercamara=null&'>Nuevo</a>
</TD>
</TR>
<TR class=even>
<TD> Planosecuencia</TD>
<TD>
    <a href='initPlanosecuenciaSearch.do'>Listar</a>
</TD>
<TD>
    <a href='initPlanosecuencia.do?idsecuenciaplano=null&'>Nuevo</a>
```

### **InitTematicaSearchAction:**

```
public class InitTematicaSearchAction extends ServiceAction {

    private static Logger log = LogService.getLogger(InitTematicaSearchAction.class);
    .....
    public ActionForward doPerform(ActionMapping mapping, ActionForm actionForm,
HttpServletRequest request,
        HttpServletResponse response) throws Exception {
        Collection tematicaList =
            getAvmtsessionService().getTematicaList();
```

Obtenemos DataObjects.

### SaveProduccionAction (fragmento)

```
public class SaveProduccionAction extends ServiceAction {
    private transient Logger log = LogService.getLogger(this.getClass());
    public ActionForward doPerform(ActionMapping mapping, ActionForm form,
    HttpServletRequest request,
        HttpServletResponse response) throws Exception {

        log.info("start doPerform");
        HttpSession session = request.getSession();
        ProduccionForm subform = (ProduccionForm) form;
        String idproduccion = subform.getIdproduccion();
        if (idproduccion != null && (idproduccion.equalsIgnoreCase("null") ||
idproduccion.length() < 1)) {
            idproduccion = null;
            subform.setIdproduccion(null);
        }
        com.audiovisuales.session.AvmtsessionService facade = getAvmtsessionService();
```

### FormatoForm

```
public final class FormatoForm extends ServiceBaseForm {
```

Ademas de los getters y setters

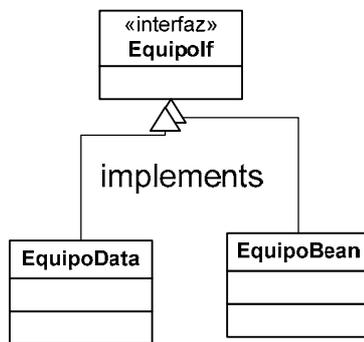
```
public int getListSize() throws Exception {
    return getAvmtsessionService().getFormatoListSize();
}

/**
 * obtiene una coleccion de todos los objetos formato.
 *
 * @throws Exception exception
 * @return a collection of FormatoData objects.
 */
public Collection getList() throws Exception {
    Collection result = getAvmtsessionService().getFormatoList();
    if (result.size() > 0) {
        return result;
    } else {
        return new ArrayList();
    }
}
}
```

## Persistencia temporal

- Como mapeador objeto-relacional nos decidimos por implementar el api-Hibernate 3
  - Se ha creado un bean por cada objeto que implementa la interficie \*if que es la interficie que expone los datos persistentes, esta inteficie la implementa una clase \*Data generando de esta manera los Value Object que pasaran de capa en capa.

PD:\* nombre del objeto.



## ProduccionData (fragmento)

Extiende el Value Object

```
public class ProduccionData implements ProduccionIf, Serializable {

    public ProduccionData(com.audiovisuales.entity.produccion.ProduccionIf value) {
        setIdproduccion(value.getIdproduccion());
        setTipoproduccion(value.getTipoproduccion());
    }
    public java.lang.Integer getIdproduccion() {
        return idproduccion;
    }

    public void setIdproduccion(java.lang.Integer idproduccion) {
        this.idproduccion = idproduccion;
    }

    public java.lang.String getTipoproduccion() {
        return tipoproduccion;
    }

    public void setTipoproduccion(java.lang.String tipoproduccion) {
        this.tipoproduccion = tipoproduccion;
    }
}
```

## ProduccionIf

La interficie ProduccionIf interfaces muestra los datos persistentes  
Setters y getters.

## ProduccionBean

El entity Produccion

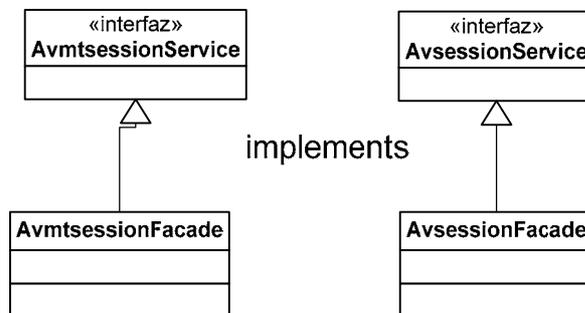
```
public class ProduccionBean implements Serializable,  
com.audiovisuales.entity.produccion.ProduccionIf {
```

```
@hibernate.class
```

```
*     table="produccion"
```

## Capa de Negocio

Los sessions



### AvmtsessionFacade

Envuelve los entities formato, produccion ,tematica ,equipo, soportefisico.

```
public class AvmtsessionFacade implements AvmtsessionService ,
LocatableService {
```

```
public com.audiovisuales.entity.formato.FormatoIf
addFormato(com.audiovisuales.entity.formato.FormatoIf model) throws
GenericBusinessException {
    com.audiovisuales.HibernateQueryHelper hibernateTemplate = new
com.audiovisuales.HibernateQueryHelper();
    FormatoBean value = new FormatoBean();
    try {
        value.setIdformato(model.getIdformato());
        value.setTipoformato(model.getTipoformato());
        // Now save the data.
        hibernateTemplate.save(value);
        return getFormato(value.getPrimaryKey());
    } finally {
        log.debug("finished
addFormato(com.audiovisuales.entity.formato.FormatoIf model)");
    }
}
```

```
public void saveFormato(com.audiovisuales.entity.formato.FormatoIf
model) throws GenericBusinessException {
    // We have to create an ejb object:
    FormatoBean data = new FormatoBean();
    com.audiovisuales.HibernateQueryHelper hibernateTemplate = new
com.audiovisuales.HibernateQueryHelper();
    data.setIdformato(model.getIdformato());
    data.setTipoformato(model.getTipoformato());
    try {
        // Now update the data.
        hibernateTemplate.update(data);
    } finally {
        log.debug("finished
saveFormato(com.audiovisuales.entity.formato.FormatoIf model)");
    }
}
```

```
public void deleteFormato(java.lang.Integer id) throws
GenericBusinessException {
    com.audiovisuales.HibernateQueryHelper hibernateTemplate = new
com.audiovisuales.HibernateQueryHelper();
    try {
        // First get the data.
        FormatoBean bean = (FormatoBean)
hibernateTemplate.get(FormatoBean.class, id);
        hibernateTemplate.delete(bean);
    } finally {
        log.debug("finished deleteFormato(java.lang.Integer id)");
    }
}
```

### AvmtsessionFacade

Envuelve los entities MasterCamara, MasterEdicion , PlanoSecuencia,  
Video ,ProductoFinal.

## Conclusiones.

- Es difícil implicar a los usuarios en el proyecto
- Los patrones de diseño (MVC, Mapeo O-R Hibernate, Struts) y los frameworks me han facilitado el desarrollo y me han permitido el aprendizaje de experiencias adquiridas por otros desarrolladores
- Las librerías creadas por la comunidad (Hibernate, Struts, Displaytags.) me han evitado el desarrollo propio, el testing y el mantenimiento de librerías propias, por lo que han reducido el tiempo de desarrollo y facilitan el mantenimiento si el proyecto lo recoge otra persona
- La arquitectura J2EE es idónea para el desarrollo de aplicaciones distribuidas a nivel empresarial que requieran escalabilidad, integración, consistencia, robustez y seguridad
- He empleado herramientas para todo el ciclo de vida del software (ms project, powerdesigner, word, eclipse, frameworks y librerías) que en algunos casos desconocía

## Bibliografía

- J2EE Design and Development –Rod Johnson
- Mastering Enterprise JavaBeans-Ed Roman
- Hibernate: A J2EE™ Developer's Guide- Will Iverson

## Recursos de Internet

- [www.java.sun.com](http://www.java.sun.com)
- [www.programacion.com/java/tutorial/struts/](http://www.programacion.com/java/tutorial/struts/)
- [http://www.hibernate.org/hib\\_docs/v3/reference/en/html/tutorial.html#tutorial-firstapp-helpers](http://www.hibernate.org/hib_docs/v3/reference/en/html/tutorial.html#tutorial-firstapp-helpers)
- [www.w3schools.com](http://www.w3schools.com)
- [Jakarta.apache.org/struts](http://Jakarta.apache.org/struts)