Universidad Zaragoza
1542

Proyecto Fin de Carrera

Generación automática configurable de feeds

Autor/es

Carlos Serrano Sanz

Director/es y/o ponente

Marcos Latorre Díez

Eduardo Mena Nieto

Escuela de Ingeniería y Arquitectura

2014-2015

GENERACIÓN AUTOMÁTICA CONFIGURABLE DE FEEDS

RESUMEN

El trabajo realizado ha consistido en realizar un módulo nuevo en un producto propio de la empresa Hiberus Tecnología. El producto se trata de una aplicación web llamada Hermes que sirve para gestionar todos los aspectos relativos a la configuración de un portal de comercio electrónico (productos, pedidos, contenidos…).

El módulo realizado se encarga de externalizar la información de los productos de los portales de comercio electrónico de manera configurable por el administrador de cada portal. La externalización se traduce a creación de archivos y su compartición. Estos archivos se denominan feeds y contienen la información necesitada para cada situación.

El objeto de la generación de feeds es la compartición de los mismos con plataformas externas para que estas dispongan de la información deseada, es por eso que la personalización de estos feeds es tan importante, ya que cada plataforma demanda la información de manera única.

El módulo permite la configuración del esqueleto del fichero generado, con que datos se va a rellenar, el formato del fichero resultante, que productos aparecerán en cada fichero…. También permite generar los feeds en diferentes idiomas según se requiera.

Otra funcionalidad que ofrece el módulo es la frecuencia de generación de los feeds. Ya que cada plataforma puede demandar la renovación de la información compartida cada una cierta frecuencia, el módulo permite generar cada feed con una frecuencia distinta y configurable según se requiera.

Por último, el sistema cuenta con una funcionalidad de backup, que almacena información acerca de cada proceso de generación de un feed, de modo que si hay algún fallo es más fácil detectarlo. Además guarda una copia de cada feed generado por si se necesita recuperarlo.

TABLA DE CONTENIDOS

1. Introducción

En este capítulo se explican los aspectos principales del desarrollo del Proyecto Fin de Carrera. Se detalla el contexto en el que se ha llevado a cabo, las características fundamentales del mismo y se exponen el resto de secciones que forman esta memoria.

1.1. Contexto del proyecto

El proyecto se ha desarrollado en la empresa Hiberus Tecnología, como Proyecto Fin de Carrera de Ingeniería Informática, de la Escuela de Ingeniería y Arquitectura de Zaragoza.

En el desarrollo del proyecto han intervenido dos empresas de distintas características. Estas empresas han sido Hiberus Tecnología como se ha comentado previamente e Imaginarium. Hiberus ha hecho el papel de empresa desarrolladora del proyecto e Imaginarium ha participado como cliente.

Hiberus Tecnología es una compañía especializada en la consultoría de negocio y la prestación de servicios tecnológicos y outsourcing. Es la compañía de tecnología líder del Valle del Ebro, referente en el mercado español y en pleno proceso de expansión en el mercado latinoamericano.

Mientras que Imaginarium es una de las marcas líderes en el sector infantil a nivel mundial, que se caracteriza por ser pionera en el concepto del juego educativo.

Estas dos empresas van unidas de la mano por el ámbito web. Hiberus Tecnnología es la empresa encargada de desarrollar y mantener el portal web de Imaginarium (tanto en el dominio español como en el resto de dominios que ofrece Imaginarium), y todo lo que le rodea desde hace más de ocho años.

Es por eso, que este proyecto nació a través de una petición de Imaginarium para mejorar su experiencia web, y no hubiese sido posible sin su dedicación y participación.

1.2. Descripción del proyecto y objetivos principales

El objetivo principal de este proyecto ha sido ofrecer una solución tecnológica a una petición muy concreta realizada por la empresa Imaginarium a Hiberus: externalizar los productos vendidos en su portal web.

La aplicación web de Imaginarium está soportada por un producto propio de la empresa Hiberus. Este producto se llama Hermes. Hermes es una plataforma de comercio electrónico que permite modelar y alinear la estrategia offline de una compañía con su presencia online.

¿Quiere decir esto que antes de este proyecto, Imaginarium no externalizaba sus productos web? Naturalmente sí que lo hacía, pero no existía un sistema personalizado para que los propios empleados de Imaginarium pudiesen hacerlo. Es decir, si Imaginarium quería externalizar sus productos web con cualquier otro portal de venta online, debía pagar un desarrollo nuevo ya que cada portal requería una forma peculiar de tratar con los datos.

En este proyecto, se ha desarrollado una herramienta integrada en Hermes, capaz de generar y tratar dichos datos de una forma totalmente personalizable, de modo que ahora Imaginarium puede externalizar los datos de sus productos web sin el requerimiento de nuevos desarrollos.

Otro objetivo principal, es por supuesto, la satisfacción del cliente. Imaginarium ha depositado su confianza en Hiberus para llevar a cabo este proyecto y es por eso que se tiene como compromiso que quede satisfecho con el resultado del proyecto. El éxito del proyecto afianzará las relaciones entre ambas empresas.

Este proyecto también servirá como retroalimentación tanto para la aplicación Hermes, como para otras empresas que tengan como base esta herramienta, ya que podría ofertarse en un futuro a estas empresas.

1.3. Estructura de la memoria

En este apartado se va a describir como está estructurada esta memoria. La memoria está dividida en dos partes: la memoria principal y los anexos.

La memoria principal se divide a su vez en diferentes capítulos. Este primer capítulo ha servido como introducción, en el que se ha descrito el contexto del proyecto y sus objetivos principales.

En el segundo capítulo se describen en detalle los primeros pasos que se han seguido para la realización del mismo: la captura de requisitos, los estudios previos realizados y la planificación del proyecto.

Finalmente, los anexos complementan la memoria principal para describir con mayor detalle aspectos relevantes del proyecto.

Anexo A: Análisis del sistema.

Anexo B: Desarrollo del sistema.

Anexo C: Base de datos.

Anexo D: Tecnologías utilizadas.

Anexo E: Gestión del proyecto.

Anexo F: Manual de usuario

2. Definición del sistema Product Feed

En este capítulo se presentan las primeras fases del proyecto Product Feed. Estas fases comienzan con la captura de requisitos necesarios para el desarrollo del proyecto, hasta la planificación del mismo.

Todo comienza con la petición que Imaginarium hace a Hiberus como se ha comentado en el capítulo de introducción. A ambas empresas les parece un proyecto tanto viable como necesario. Desde este momento se comienza a analizar la situación para generar una lista de requisitos que deberán cumplirse al finalizar el proyecto.

2.1. Estudios previos

Antes de comenzar con el análisis de requisitos, se hace un estudio previo para ajustarse mucho más tanto al resultado concreto de los requisitos que se obtendrán como al resultado general del proyecto.

Como se ha comentado en la introducción, Imaginarium ya externalizaba sus productos a otros portales. Lo primero que hice fue analizar cómo se estaba haciendo hasta el momento. Me informé preguntando al responsable de la web de Imaginarium. Para cada portal o canal con el que Imaginarium compartía los datos de sus productos existía un proceso particular y único que se ejecutaba periódicamente, cosa lógica ya que los datos de los productos cambian con frecuencia. En todos los casos el resultado del proceso generaba uno o varios ficheros que contenían información de una serie de productos, y se depositaban en una determinada ruta en la misma máquina dónde estaba corriendo la aplicación web.

Además, pregunté a Imaginarium con qué otros canales se pensaban compartir datos en un futuro. Me comentaron algunos que tenían pensados.

El primero de ellos era un buscador que tenían planeado implantar en el portal web. Este buscador era un buscador en tiempo real, por lo que debe nutrirse de información de los productos al instante, no sería eficiente consultar dicha información en bases de datos en el momento de búsqueda, por lo que era preciso que la información de los productos estuviese externalizada en ficheros.

El segundo era una afiliación llamada Affiliate4You. Era una afiliación holandesa, de modo que requería información exclusivamente de los productos que se vendían en la web de Imaginarium de Holanda.

En el último caso necesitaban compartir información de sus productos con un sector de la empresa de Hiberus llamada Semmantica. Semmantica se dedica a la parte analítica y estadística de los portales web, de modo que precisaba información acerca de los productos web de Imaginarium.

Aquí es donde aparece el término product feed. Un product feed es un fichero sin restricciones de formato que contiene información acerca de una serie de productos y sus características. El uso más común de product feed aparece cuando una tienda (generalmente online) decide ampliar sus horizontes y vender sus productos en otros canales. Un ejemplo es el mencionado previamente, en el que Imaginarium quería vender sus productos a través de un afiliado como Affiliate4You. Aunque como se ha mencionado en los otros dos ejemplos, el product feed se puede usar para otros fines, como el caso del buscador, o el caso de Semmantica.

Tras analizar los requerimientos de las tres empresas mencionadas más los requerimientos de los portales con los que ya se estaban externalizando información, se llegó al análisis de requisitos.

2.2. Captura de requisitos

Tras realizar el estudio previo, el objetivo era desarrollar un módulo en Hermes que generase archivos con la información de los productos de Imaginarium.

Esta generación de archivos debería ser totalmente configurable. Estos requisitos de configuración se recogen a continuación, aunque se describirán con más detalle en el Anexo A – Análisis del Sistema:

- El sistema constará de tres sub-módulos:

1. Configuración del canal.

   Este sub-módulo permitirá la creación, modificación, eliminación y clonación de los distintos canales configurables para los que se generarán los feeds.

2. Configuración de la creación del feed.

   Este sub-módulo será el encargado de permitirnos configurar la generación del feed. Se divide en dos apartados a su vez, uno para especificar los atributos que tendrá cada feed y otro que servirá para configurar los referente a la frecuencia de generación del archivo.

3. Listado de feeds

   Este sub-módulo nos permitirá el visionado de los feeds que tenemos configurados actualmente y sus características. Además a partir de este módulo podrán cambiarse la configuración de los feeds generados.

La generación de product feed suele ser un proceso costoso, ya que los archivos generados pueden llegar a contener información de miles de productos. De modo que el módulo también cuenta con un sistema de alarma que enviará un correo si ha habido algún problema o se ha interrumpido algún product feed durante su generación. Además cuenta con un sistema de backup que almacena los ficheros que se generaron con anterioridad.

Estos solo son los requisitos descritos sin entrar mucho en detalle, como se ha dicho, los requisitos se detallarán en el Anexo A - Análisis del Sistema.

3. Descripción detallada del sistema Product Feed

En este apartado de la memoria se describe detalladamente el funcionamiento de la aplicación resultante junto con la arquitectura interna. Este es el resultado de la implementación tras la captura de requisitos, el análisis del sistema y el desarrollo del sistema. Estos aspectos se describen en los anexos: Anexo A – Análisis del Sistema y Anexo B – Desarrollo del Sistema.

El módulo se divide en cuatro apartados, tres visibles al usuario y uno sin interfaz gráfica. Los tres visibles son: la gestión de canales, la gestión del feed y el listado de feeds, mientras que el invisible es el sistema de backup y alertas.

3.1. Gestión de canales

En este apartado de la aplicación se gestiona todo lo referente a cada canal para el que se quiera generar un feed. ¿Por qué se necesitan apartados diferentes para la gestión del canal y la gestión del feed, si el resultado va a ser un fichero?

Porque un canal nos determina una serie de requerimientos, que normalmente se traduce a que el feed contenga para cada producto una serie de atributos, pero podemos necesitar para un mismo canal diferentes feeds, con diferentes productos en cada feed, o en diferentes idiomas.

Veámoslo con un ejemplo. Por ejemplo, se quiere generar un product feed para Google Shopping. En la hoja de requerimientos se piden las siguientes pautas:

- El fichero resultante deberá tener formato .xml
- El fichero deberá comenzar con el tag <products> y acabar con el tag </products>.
- Cada producto deberá comenzar con el tag <product> y cerrarse con el tag </product>.
- Todos los tags de atributos deben comenzar con la cadena de texto g:
- Los atributos id_pmm, nombre y precio son obligatorios.
- Los atributos stock y categoría son recomendados.

Y en el documento que proporciona Google Shopping se indica un ejemplo de product feed resultante:

```
<products>
    <product>
        <g:id_pmm> 17728 </g:id_pmm>
        <g:nombre> kikonico </g:nombre>
        <g:precio> 17.95 </g:precio>
    </product>
    <product>
        <g:id_pmm> 17729 </g:id_pmm>
        <g:nombre> pakonico </g:nombre>
        <g:precio> 18.95 </g:precio>
    </product>
</products>
```

Figura 1. Estructura product feed Google Shopping

Como podemos observar en la Figura 1**,** el esqueleto del feed es muy concreto para Google Shopping, igual que para el resto de feeds que necesitemos generar. Pero es probable que necesitemos distintos feeds para cada canal que contengan diferentes productos, o la información en distintos idiomas. Es por eso que se necesita configurar las características de cada canal por separado a las características de cada feed.

Para la gestión de canales, como se analizó y diseñó (Figuras 2 y 3) existen cuatro operaciones: crear, modificar, eliminar y clonar.

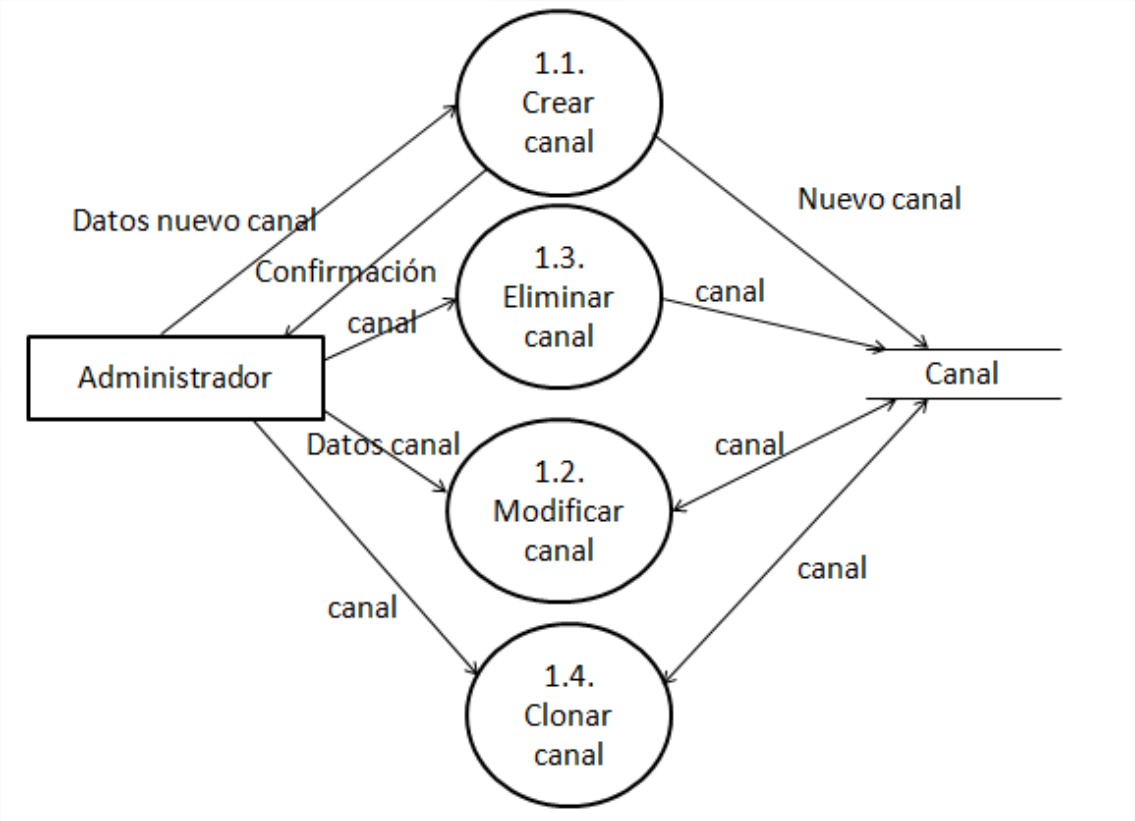


Figura 2. DFD Nivel 2 – gestión canales

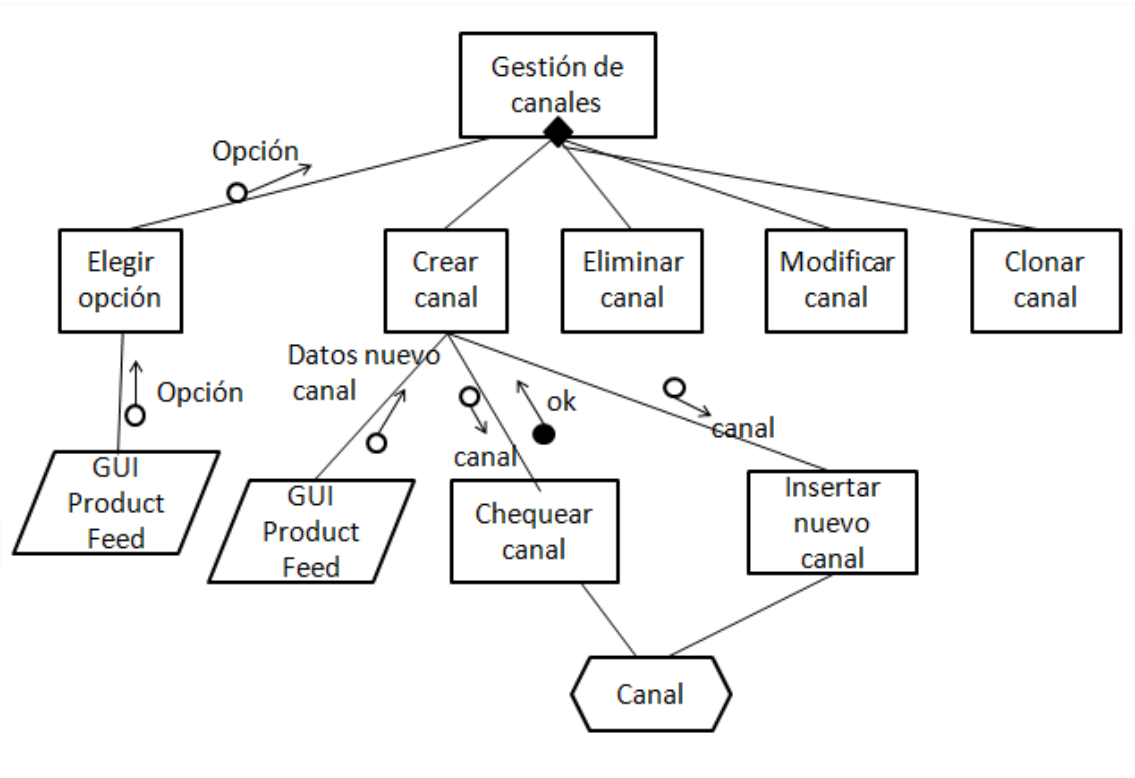


Figura 3. DDE gestión de canales

Para la creación de un canal se obtienen a través de la interfaz gráfica las siguientes características: nombre, nombre_tag, precio_tab, producto_tag, cabecera_tag. Esto resulta en el siguiente tipo de entidad creado en la base de datos para almacenar la persistencia de canales (Figura 4). Si llevamos esto al ejemplo de la figura 1 el nombre no es relevante para generar el fichero. El nomrbe_tag correspondería a “g:” que es la cadena de texto con la que comienzan todos los tag. Precio_tag no es usado en este fichero. Este atributo se creó para algo muy concreto: se usa en el caso de que se seleccionen como atributos el precio del producto y el precio rebajado, y en el caso de que no exista precio rebajado el atributo precio se nombrará con lo que contenga Precio_tag. Producto_tag es la cadena de texto que se usa para abrir y cerrar los tags de cada producto, en el caso del ejemplo corresponde a la cadena “product”. Finalmente Cabecera_tag corresponde al inicio y cierre del fichero, en este caso “products”.

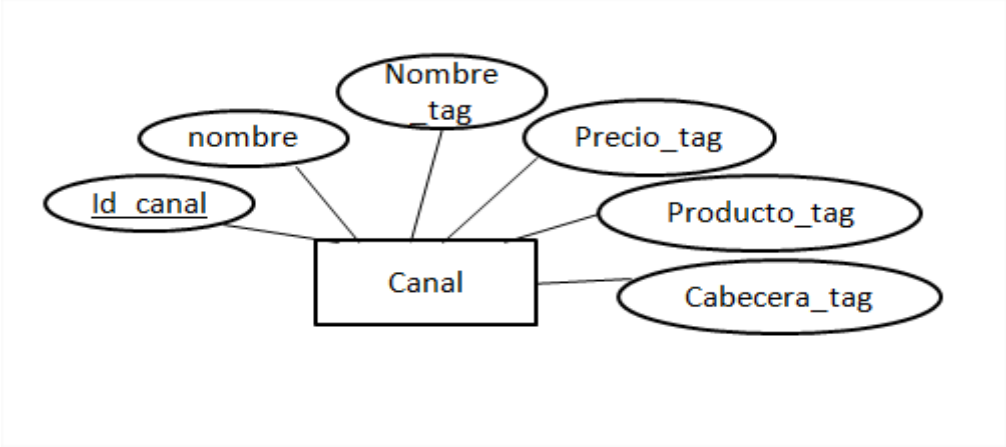


Figura 4. Tipo de entidad canal

El identificador id_canal se genera automáticamente en cada inserción a esta tabla. Con esta información no tenemos completo un canal, faltan los atributos que va a tener dicho canal, de modo que se ha generado otro tipo de entidad “Atr_canal” ya que tiene sus características propias (Figura 5).

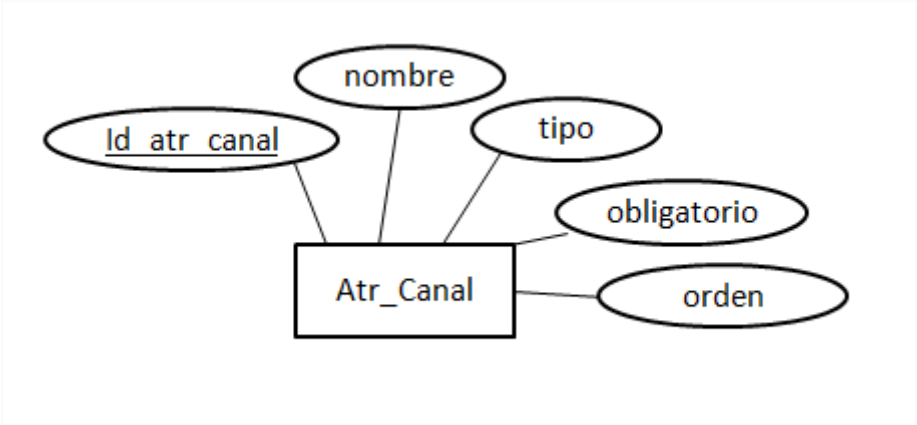


Figura 5. Tipo de entidad atr_canal

Para cada atributo del canal se almacena un identificador que se genera automáticamente, el nombre de dicho atributo, su tipo, su orden respecto a los demás atributos y su obligatoriedad. Estos datos son los necesarios para generar el esqueleto del fichero. Si observamos la figura 1 serían necesarios tres nuevos atributos: id_pmm, nombre y precio. Para almacenar el nombre se insertaría un nuevo atr_canal en la base de datos, que estaría relacionado con el canal que se ha insertado previamente. Como nombre tendría “nombre” y como tipo “texto”. Como orden tendría el 2, ya que hay tres atributos y en el fichero el nombre aparece en segundo lugar. Para saber su obligatoriedad se deberían consultar las especificaciones del canal para ver si dicho atributo debe aparecer en el fichero obligatoriamente.

Además los atributos pueden tener un atributo padre o estar contenidos dentro de otro atributo. Es por esto que se generaron dos relaciones nuevas en la entidad (Figura 6)

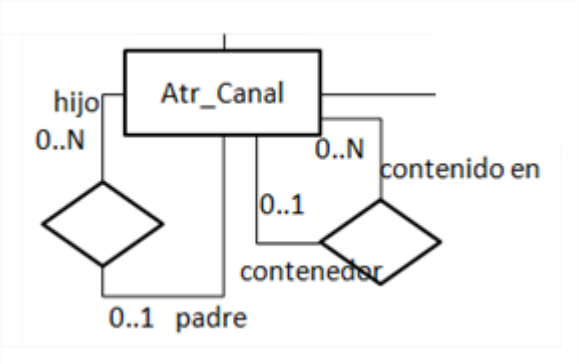


Figura 6. Elaciones Atr_canal

Con esta estructura podremos generar los ficheros con unos atributos que contengan otros o que tengan hijos. En las Figuras 7 y 8 se muestra un ejemplo de atributos con hijos y atributos contenidos en otros atributos respectivamente.

```
<product>
    <id_pmm> 17728 </id_pmm>
    <precio>
        <valor> 9.95 </valor>
        <divisa> EUR </divisa>
    </precio>
</product>
```

Figura 7. Atributos anidados

```
<product>
        <id_pmm> 17728 </id_pmm>
        <precio divisa = “EUR””> 9.95</precio>
</product>
```

Figura 8. Atributos contenidos en

La modificación de un canal se hace a través de la misma interfaz gráfica que la creación. Cuando se accede a la modificación del canal se consultan de la base de datos los datos de dicho canal y sus atributos y se muestra al usuario. Si se realiza algún cambio sobre el canal o sobre alguno de sus atributos se realiza la operación de actualización de esa fila en la base de datos.

La eliminación de un canal es sencilla. Se deben consultar primero todos los atributos relacionados con ese canal de la tabla “Atr_canal” y se deben eliminar. Una vez se han eliminado los atributos del canal se procede al borrado del canal de la tabla “Canal”.

Para realizar la clonación del canal se consulta el canal que se quiere clonar y se crea una entrada en la tabla “Canal” con los mismos datos, aunque distinto identificador y distinto nombre. El identificador se autogenera, y el nombre para hacerlo distinguible al usuario se le añade al nombre original la cadena “Copia de” que posteriormente puede modificarse. Se barajó la posibilidad de no copiar los atributos de canal sino relacionar los mismos atributos con la copia creada del canal. Pero se descartó ya que no se podrían hacer cambios en los atributos de un canal sin que eso afectase a los atributos de la copia. Es por eso que se decidió que cada canal tuviese sus propios atributos aunque algunos fuesen iguales que los de otro canal.

Una vez configurado todo lo relevante al canal, ya tenemos preparado el esqueleto que tendrán todos los feeds generados para ese canal.

Ahora queda indicarle con qué datos se va a generar cada feed. Esto se configura en la gestión de generación de feed.

3.2. Gestión de generación del feed

Para la generación del feed se deben tener en cuenta dos aspectos importantes, con qué datos se va a generar el feed y cada cuanto tiempo se generará.

Estos dos aspectos se aprecian en el análisis y diseño realizados (Fguras 9 y 10).

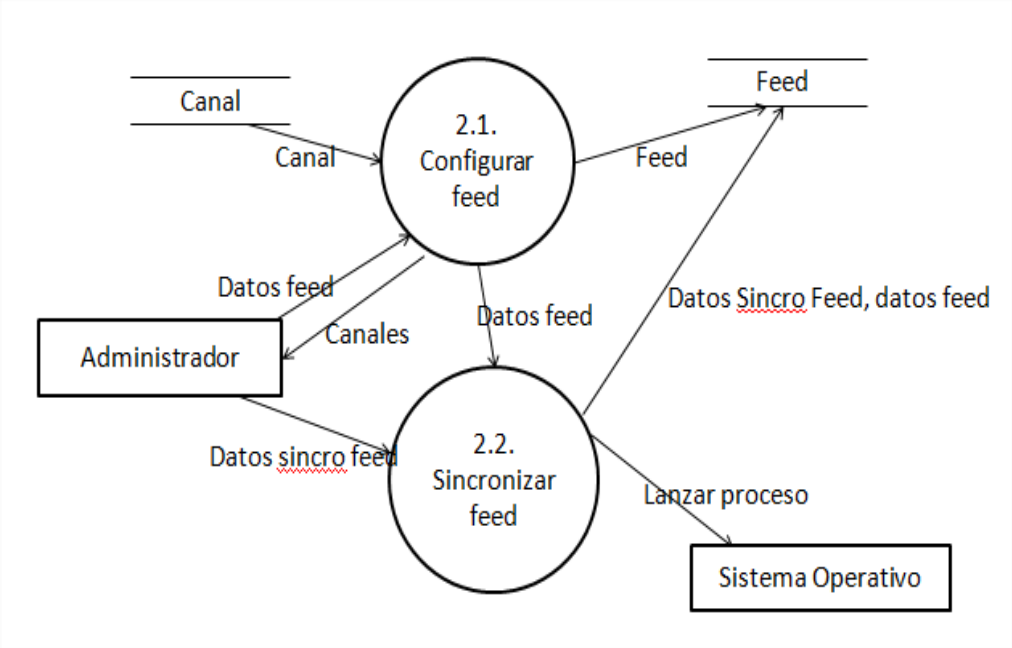


Figura 9. DFD Nivel 2 – gestión de generación del feed

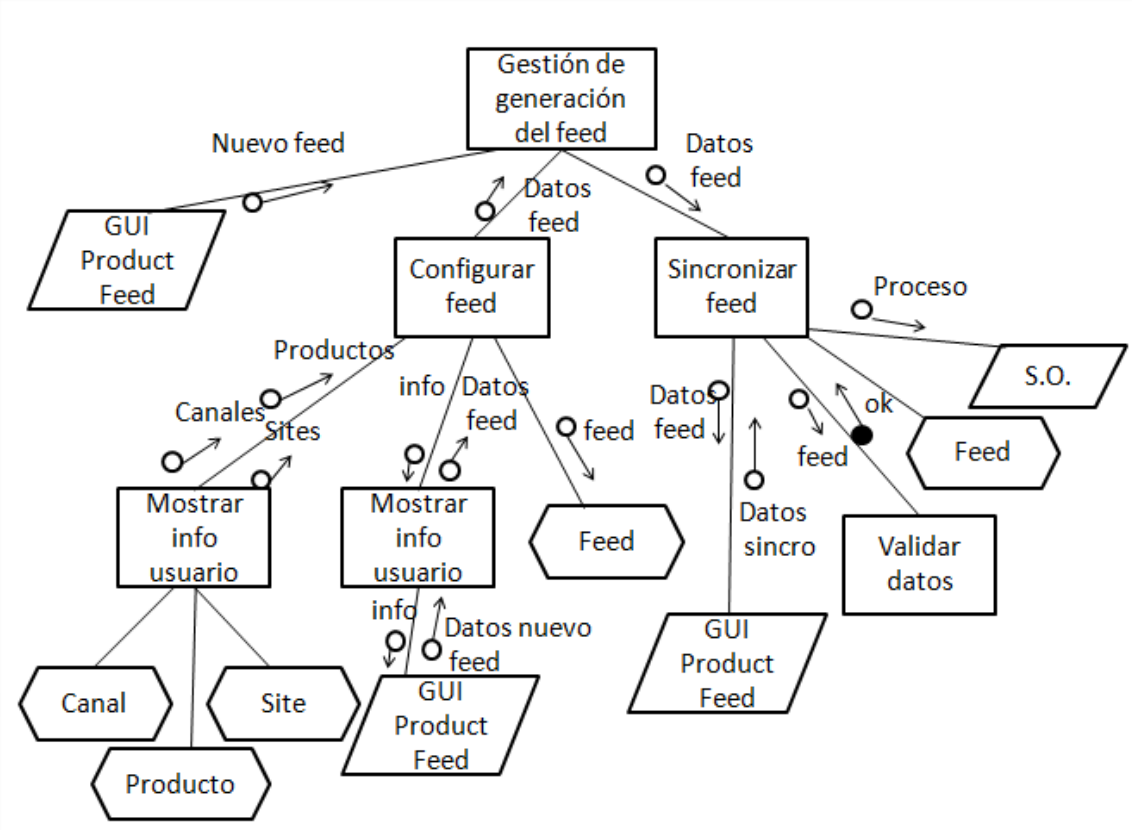


Figura 10. DDE gestión de generación de feed

Para almacenar la información de los feeds se ha generado un tipo de entidad denominada “Feed” (Figura 11).

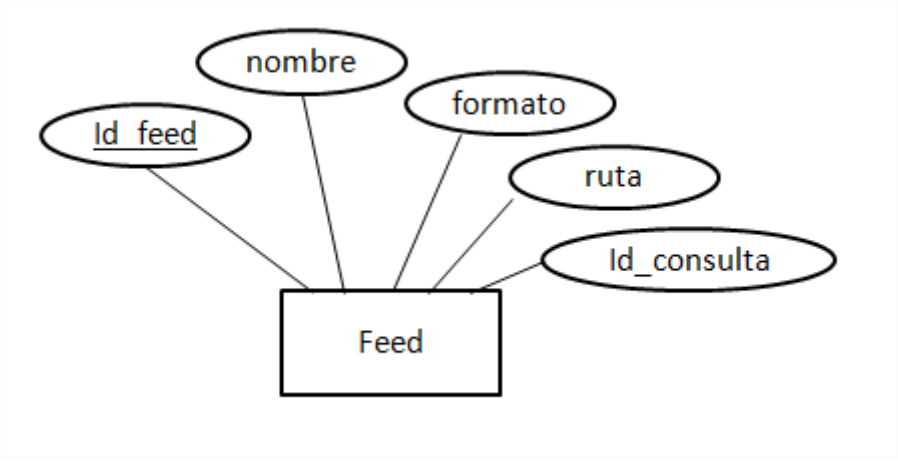


Figura 11. Tipo de entidad feed

El identificador del cada feed se generará automáticamente cada vez que hagamos una inserción en la tabla. Para cada feed nos interesará almacenar su nombre, el formato en el que se generen los archivos, la ruta dónde se depositarán los mismos y el id_consulta. El id_consulta se utiliza para saber que productos deben aparecer en dicho feed. Esto se explicará más adelante en este capítulo.

Un feed puede pertenecer únicamente a un canal de los que se han configurado previamente. Para cada feed no se tiene porqué generar únicamente un archivo, sino que se genera un archivo por cada site de los que se selecciona para el feed. Un site es un sitio web. Se decidió generar un fichero por cada site ya que cada site tiene asociados una serie de productos. Además cada site tiene su propio idioma y era de utilidad poder generar feeds en distintos idiomas. De modo que entre la entidad “feed” y “site” nace la relación “feed_site” (Figura 12) que traduciremos cada relación como un fichero que se generará.

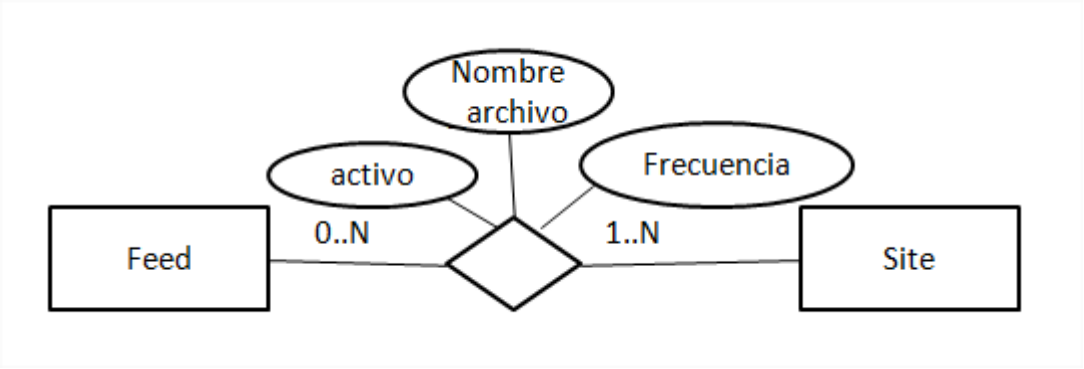


Figura 12. Relación feed-site

Para cada feed-site guardaremos si está activo o no, el nombre del archivo que se generará y la frecuencia de generación. La frecuencia de generación es una expresión cron. Las expresiones cron son expresiones que denotan una frecuencia, por ejemplo, todos los días a las 8.00 a.m. Son usadas frecuentemente por el sistema operativo para realizar funciones de mantenimiento o funciones programadas en una frecuencia determinada. Cuando se termina la configuración de generación del feed se crea un trigger. Un trigger se traduce a un disparador que ejecuta tarea en el sistema operativo cada cierto tiempo. Este tiempo vendrá establecido por la generación cron que hemos generado al captar la frecuencia que se ha configurado a través de la interfaz gráfica. La tarea que se ejecutará no será otra que la generación del feed. Aunque tras generar el trigger con la expresión cron, debemos guardar la expresión en la base de datos por si se para el sistema. Si no se guardase la generación del feed funcionaría correctamente hasta que hubiese un problema en el sistema o se tuviese que reiniciar. Es por eso que existe otro trigger que

se activa al iniciar la aplicación, que lee todos los feed_sites actuales y pone en marcha sus triggers.

Otro aspecto importante es el mapeo de los atributos del canal con los valores que tendrán en el feed. En el ejemplo de la figura 1, en el apartado de canal se explicaba que estructura iba a tener el fichero con sus tags, pero no se ha dicho nada de el valor que tendrá cada atributo dentro del fichero. De aquí nace una relación ternaria entre el tipo de entidad “feed” que será en cada caso el feed que se esté configurando, el tipo de entidad “atr_canal” que vendrá dado por el canal que se haya seleccionado para el feed a configurar y un nuevo tipo de entidad denominado “mapeo_atr (Figura 13)”.

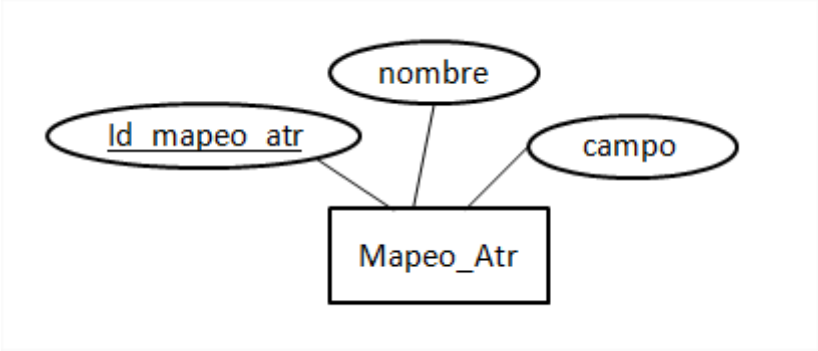


Figura 13. Tipo de entidad Mapeo_atr

Esta tabla es algo intuitivo para saber el valor que se debe seleccionar para cada atributo canal dado un feed. Se pensó en un principio relacionar con los atributos directamente de la base de datos de Imaginarium, pero fue tarea imposible ya que estaban totalmente dispersos y cada atributo tenía estructuras diferentes. De modo que en el proceso de generación del feed se consulta esta tabla y dependiendo del valor de “nombre” y “campo” se hace una consulta al valor que represente.

Finalmente nos queda saber que productos aparecerán en cada fichero generado. Se dan cuatro posibilidades de seleccionar los productos:

- Consulta de productos. Se pueden seleccionar un conjunto de atributos a través de una consulta de atributos personalizada. Esto es una utilidad que ya incorporaba Hermes con otras utilidades, de modo que se no va a indagar mucho en ello. Simplemente, si se elige esta opción aparecerán en el feed los productos del resultado de la consulta de productos seleccionada.
- Categorías ecommerce. Las categorías ecommerce son aquellas categorías que aparecen en la web de Imaginarium. Por lo que cada categoría ecommerce tiene una serie de atributos asociados a la misma.
- Categorias pmm. Las categorías pmm son las categorías que usa Imaginarium internamente, las que utilizan en su ERP. No son más

que otra forma de agrupar productos. Se ha habilitado la posibilidad de elegir los dos tipos de categoría ya que según qué tipo de empleado de Imaginarium utilice la herramienta puede estar más familiarizado con las categorías pmm o con las categorías ecommerce.

- Todos los productos. Esta opción mostrará en el feed la información de todos los productos que estén disponibles para la venta en cada site.

Esta información se almacena en el tipo de entidad “Feed” (Figura 11). Si es una consulta de productos se almacena el id de la consulta de productos que ya existe en Hermes. Si se trata de todos los productos se almacena un 0. Si es por categorías pmm se almacena un -1 y un -2 si son categorías ecommerce. Existe una relación entre “Feed” y la tabla “categoría_pmm” y otra entre “Feed” y “categoría_ecommerce” para almacenar que categorías están relacionados con los feed.

.

3.3. Listado de feeds

El listado de feeds simplemente muestra la información acerca de cada “feed_site” actualmente en el sistema. Para mostrar esta información se accede a las tablas mencionadas en los apartados anteriores. También se muestra la fecha de la última vez que se generó cada fichero y la próxima vez que se hará. Esto se calcula a través de la expresión cron que se generó cuando se configuró el site y se guardó en la base de datos.

Adicionalmente, si un proceso está generándose actualmente se muestra una barra de progreso para informar de su avance. Se ha creado un monitor en la aplicación para controlar el progreso de todas las tareas que estén generando feeds. Un monitor es un elemento de programación para el control de sistemas concurrentes estudiado en la asignatura Programación se Sistema Concurrentes y Distribuidos. Básicamente se utiliza para almacenar la información de cada proceso y que esté accesible desde cualquier otro proceso de la aplicación. De modo que cada vez que se activa un proceso de generación de feed, informa al monitor de que ha comenzado y va informando del avance del proceso. Es por eso que se puede mostrar en el listado de feeds el avance de cada proceso en tiempo real, solo hay que consultar el monitor en cada momento y este nos informa de los procesos activos.

3.4. Histórico, sistema de backup y alertas

Este módulo sin interfaz gráfica es el encargado de gestionar un sistema de datos históricos, un sistema de backup y un sistema de alertas.

Cada vez que se genera un fichero, automáticamente se genera una entrada en la base de datos que utiliza la aplicación web de Imaginarium. Se guarda en una tabla información acerca del proceso, del feed, si ha tenido éxito o no entre otras cosas. Se detalla más esta información en el Anexo C – Base de Datos.

A su vez, cada vez que se genera un fichero, se almacena el fichero generado en una carpeta de backup. Esto se realiza ya que cada vez que se repite un proceso de generación de feed, ya que el nombre del archivo es el mismo, se sobrescribe el fichero anterior. Es por eso que cada vez que se genera un fichero, queda guardada una copia en una carpeta llamada “backup” en la misma ruta dónde se estuviese generando el mismo fichero.

Finalmente, se ha configurado un sistema de alarma que se ejecuta si ha habido algún tipo de problema durante la generación de un fichero. Este sistema de alarma envía un correo electrónico a Imaginarium yal responsable técnico de Hiberus con información detallada acerca del proceso que falló. Esta información incluye el nombre del feed que falló, así como la fecha y el falló que lanzó la máquina virtual java.

4. Conclusiones

4.1. Posibles mejoras y ampliaciones

Una ampliación casi inmediata será sin lugar a duda trasladar el módulo al producto Hermes genérico. Imaginarium es uno de los clientes más antiguos que usan Hermes, y utiliza una versión de Hermes totalmente a medida. Actualmente se está trabajando en un Hermes genérico que ya está implantado en muchos otros clientes de Hiberus. Es por eso que este módulo probablemente se traslade a este Hermes genérico haciendo alguna adaptación para mejorar el producto ofertado por Hiberus.

Una posible mejora que podría aplicarse al proyecto podría ser la importación de datos a través de ficheros. A veces si un canal requiere de muchos atributos configurarlo es costoso ya que se precisa de información para cada atributo. Es por eso que una posible mejora sería dar la posibilidad al usuario de importar los atributos a través de ficheros externos de formato xml, csv…. Esta mejora mejoraría la experiencia del usuario y agilizaría el proceso de configuración.

4.2. Valoración personal

La realización de este proyecto ha sido una experiencia enriquecedora para mí. Me ha hecho evolucionar tanto en el aspecto profesional como en el personal. En el aspecto profesional, aparte de todas las técnicas de programación aprendidas, he conocido como es un proyecto real en el ámbito de una empresa real y las diferencias existentes entre lo aprendido en la carrera y la experiencia profesional.

También he aprendido la importancia y el papel que tienen el cliente en el proyecto. Al fin y al cabo el proyecto resultante tiene que satisfacer al cliente, de modo que es importante estar en comunicación constante con el cliente, transmitirle el avance del proyecto, las dudas que puedan ir surgiendo o alguna sugerencia en el caso de que se crea que puede beneficiar el resultado final.

Los conocimientos técnicos adquiridos en la carrera me han sido de gran utilidad, aunque lo que más valoro de lo aprendido en la carrea para enfrentarme a un proyecto de estas características es la capacidad de adaptación al contexto y la capacidad de auto aprendizaje adquirida a lo largo de la carrera con las diferentes asignaturas.

4.3. Agradecimientos

En primer lugar agradezco a Hiberus el haberme ofrecido la posibilidad de realizar este proyecto como proyecto fin de carrera y por haberme ayudado en todo lo que estuviese en su mano.

Agradezco a Imaginarium por hacer este proyecto posible y por poner de su parte en las reuniones que se llevaban a cabo para mejorar el producto final.

También agradezco a mis dos tutores. Mi tutor en la universidad Eduardo Mena y mi tutor en la empresa Marcos Latorre por ayudarme con el proyecto y con la realización de esta memoria.

Agradezco a mis dos compañeros de trabajo Laura Anchuelo y Tenesor Guillén por ayudarme cuando tenía algún problema técnico.

Finalmente agradezco a mi familia y seres queridos por apoyarme cuando lo requería durante la realización del proyecto.

5. Bibliografía

Struts

[1] Chuck Cavaness, “Programming Jakarta Struts, 2nd Edition”, O’Reilly Media.

[2] Página oficial de documentación de Struts

http://struts.apache.org

Spring

[3] Craig Walls, “Spring in Action, Fourth Edition”, Anaya Multimedia

[4] API de Spring

https://spring.io/docs

Java

[5] API de Java 1.6

http://docs.oracle.com/javase/6/docs/api/

Html, Css y Javascript

[6] Página de referencia de tutoriales de distintas tecnologías

http://www.w3schools.com/

ANEXO A. ANÁLISIS DEL SISTEMA

1. Introducción

El propósito de este anexo es recoger el análisis del módulo de generación automática de ficheros de product feed de Hermes, a través del análisis de requisitos y el estudio de los casos de uso, que ayudarán a entender todas las funcionalidades que ofrece dicho módulo.

Este proyecto se desarrolla con el objetivo de cubrir la necesidad de una empresa de tener la capacidad de generar ficheros de product feed desde la administración de Hermes, permitiendo al cliente especificar todas las características requeridas para la generación de dichos ficheros.

Este documento está dividido en tres secciones, incluida esta primera introducción.

En la segunda sección se va a describir un planteamiento general sobre qué es el product feed y qué nos va a permitir el módulo generado así como el análisis de requisitos. En esta sección se van a analizar los distintos requisitos del módulo, tanto funcionales como no funcionales que se han obtenido a través de reuniones con el cliente, y que por lo tanto debe cumplir el sistema.

La tercera sección contiene el modelo de casos de uso. En esta sección se analizarán los distintos casos de uso, así como los actores y el escenario, de tal forma que se expliquen todas funcionalidades que ofrece el módulo a sus usuarios.

2. Modelo de requisitos

En esta sección se plantearán los requisitos que deberá cumplir el módulo generado al finalizar su desarrollo. Primero se definirá el planteamiento general del sistema y a continuación se describirán los distintos requisitos funcionales y no funcionales.

2.1. Planteamiento general

Un fichero de product feed, no es otra cosa que un fichero que contiene información de los productos que un comercio electrónico vende. Cuando el fichero está generado, se comparte con uno o varios canales para que los productos que contiene el fichero puedan mostrarse en este canal. Este concepto es innovador, práctico y cada vez está siendo más usado por los propietarios de un comercio electrónico.

La generación de los ficheros de product feed no es sencilla, ya que cada canal pide unas especificaciones distintas y muy concretas. Cuando se han analizado dichas especificaciones se tiene que crear un proceso específico que genere el fichero periódicamente con las características especificadas por el canal. De aquí surge la motivación de este módulo.

La meta de este módulo es la generación automática de ficheros de product feed sin necesidad de ningún conocimiento técnico. El cliente, a través de una interfaz gráfica, podrá especificar todas las características necesarias para generar un fichero de product feed, así como los aspectos relativos a la sincronización.

En el siguiente apartado, se van a analizar todos los requisitos que deberá cumplir el módulo al finalizar su desarrollo.

2.2. Requisitos funcionales

Los requisitos funcionales generados para la elaboración del proyecto han surgido de las necesidades del cliente. Dichas necesidades se han ido concretando a través de las reuniones efectuadas con el mismo.

Los requisitos funcionales del sistema son:

- La aplicación generará automáticamente los ficheros tras haberse configurado lo necesario para ello.
- El módulo estará dividido en cuatro sub-módulos. Tres con interfaz gráfica (Configuración del canal, configuración de gestión de feed y listado de feeds) y otro sin ella (histórico, sistema de backup y alarmas):

  1. Configuración del canal

  El sub-módulo permitirá la creación, modificación, eliminación y clonación de los distintos canales configurables para los que se generarán los feeds.

  - La creación/modificación de los canales permitirá:

    1- Asignar un nombre al canal.

    2- Asignar una cadena de texto común para todas las etiquetas del feed.

    3- Asignar una cadena de texto para denominar al atributo precio en el caso de que no exista precio rebajado.

    4- Asignar una cadena de texto para nombrar la etiqueta de comienzo de producto.

    5- Asignar una cadena de texto para nombrar la etiqueta de comienzo de feed.

    6- Gestionar los atributos del canal, pudiendo añadir o eliminar todos los atributos que se requieran para dicho canal.

    A su vez, para cada atributo añadido al canal, se podrá definir:

    1- El tipo de dato: texto o CDATA.

    2- Su obligatoriedad

    3- El orden en el que aparecerá en el fichero generado

4- Su atributo padre

5- Si está contenido en otro atributo

- La eliminación permitirá eliminar uno o más canales junto con todas sus características previamente configuradas.

- La clonación permitirá duplicar un canal junto con todas sus características previamente configuradas.

2. Configuración de la generación del feed

El sub-módulo estará compuesto de dos apartados: Atributos feed y Sincronización feed.

El apartado Atributos feed permitirá:

- Seleccionar el canal para el que se desee generar el feed.

- Seleccionar uno o más sites para los cuales se genere el feed.

- Seleccionar el formato de archivo que tendrá el feed (XML o CSV).

- Seleccionar para cada atributo creado para el canal, el campo de la base de datos que corresponderá a dicho atributo, además de la posibilidad de añadir a dicho valor un UTM.

- Seleccionar la forma en la que se consultarán los productos que aparecerán en el feed entre cuatro posibilidades:

- Consulta de productos.

- Categorías pmm

- Categorías ecommerce

- Todos los productos disponibles en el site

El apartado Sincronización feed permitirá:

- Asignar un nombre al feed

- Asignar la ruta de destino en la que se depositarán los ficheros generados para ese feed.

- Asignar un nombre de fichero para cada site.

- Asignar una frecuencia de generación de fichero para cada site.

3. Listado de feeds

El sub-módulo mostrará un listado con todos los feeds configurados en el sistema. Para cada feed mostrará una entrada en la lista por cada site que se esté generando un feed.

El listado mostrará la siguiente información para cada feed:

- Nombre del feed

- Nombre del canal

- Nombre del site

- Nombre del fichero resultante

- Ruta en la que se depositará

- La fecha de la última vez que se generó el fichero

- La fecha de la próxima vez que se generará el fichero

- Si se está generando en ese momento el fichero, una barra de progreso con el porcentaje del proceso de generación.

Se permitirá lanzar manualmente el proceso de generación del fichero.

Se permitirá activar/desactivar un feed.

4. Histórico, sistema de backup y alarmas

Este último sub-módulo no poseerá interfaz gráfica.

Cada vez que se genere un fichero, se guardará información acerca del proceso: el nombre del feed, la fecha y el motivo del fallo.

Además, se guardará una copia del fichero en la misma ruta dónde se haya generado dicho fichero en una carpeta denominada backup.

Si durante el proceso de generación de un fichero se produce un error, se informará vía correo electrónico con información detallada del proceso que falló e información del fallo ocurrido.

2.3. Requisitos no funcionales

Puesto que este sistema va a ser un módulo integrado en la aplicación Hermes, los requisitos no funcionales que atañen al rendimiento, disponibilidad, seguridad, accesibilidad, usabilidad... van a ser heredados por los que tiene Hermes.

2.3.1 Pruebas de aceptación

Es necesario establecer unas pruebas de aceptación previas a la implantación real del módulo en Hermes. Dichas pruebas tendrá como objetivo la validación del módulo por parte del cliente.

Podemos establecer las pruebas de aceptación en tres fases diferenciadas:

La primera fase constará de la creación de un canal nuevo o la modificación de un canal ya existente. Si el cliente es capaz de crear un nuevo cliente o modificar uno que ya existe daremos por validada esta prueba.

La segunda fase consistirá en la configuración de la generación del feed. En esta prueba se seleccionará un canal existente, se elegirán varios sites para los que se generará el feed, el formato de archivo y los atributos de la base de datos de Imaginarium que corresponderán a los atributos del feed. Esta prueba se verificará al finalizar la fase tres.

La tercera fase consistirá en definir una ruta accesible por el cliente para verificar los resultados. Se nombrarán los ficheros para los distintos sites seleccionados y se lanzará manualmente la sincronización. La prueba tendrá éxito si verificamos que se han generado correctamente los feeds en la ruta especificada, con los nombres especificados.

El objetivo final del product feed es que los productos de un comercio electrónico se vean disponibles en otros canales. La meta del uso de éste módulo es la misma, pero ciertos canales requieren de una comunicación previa por parte del comerciante con el canal para acordar condiciones. Una vez establecidas las condiciones, sí se garantiza que este módulo cumpla el objetivo final del product feed.

3. Modelo de casos de uso

Un modelo de casos de uso captura todos los actores, casos de uso y sus relaciones de un sistema. Los casos de uso representan los requisitos funcionales de un sistema. Los actores representan los conjuntos de roles que desempeñan los usuarios cuando interactúan con el sistema.

El modelo de casos de uso se describe mediante diagramas UML, especialmente diagramas de casos de uso, y descripciones textuales detalladas. Estos diagramas y descripciones muestran a los clientes, usuarios, revisores y a otros desarrolladores el conjunto de funcionalidades que desempeñará el sistema para cada actor.

3.1. Descripción general

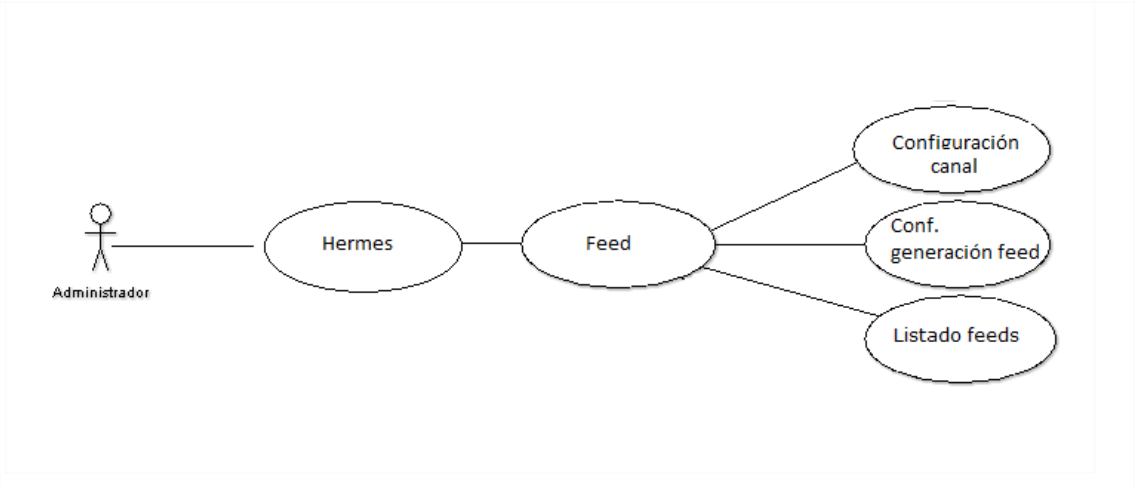


Figura 14. Casos de uso para el Administrador

El administrador accede a través de Hermes al módulo de Product Feed, Seleccionará una de las tres secciones disponibles del módulo para realizar el trabajo que corresponda. Las tres secciones que componen el módulo son la configuración del canal, la configuración de generación del feed y listado feeds.

3.2. Actores

Un actor representa un conjunto coherente de roles que juegan los usuarios de los casos de uso cuando interactúan con éstos. Cada uno de los actores del modelo de casos de uso posee los siguientes atributos:

Nombre – Nombre identificativo único del actor.

Tipo – Tipo de actor:

Principal – Actor que inicia un caso de uso para alcanzar unos objetivos.

Secundario – Actor que participa en un caso de uso.

Roles – Explicación del rol o roles del actor dentro del sistema.

Comentarios – Comentarios aclaratorios acerca del actor.

Versiones – Historial de cambios del actor. Cada registro de cambio incluye:

Número de versión, fecha y descripción de la modificación.

Para este módulo solo se contempla un único tipo de actor: el administrador. El motivo es que éste módulo está integrado en la aplicación Hermes, que es una aplicación de carácter administrativo que permite gestionar todo tipo de aspectos que pueda poseer un ecommerce, por lo que solamente administradores van a poder acceder y utilizar el módulo de product feed.

3.2.1 Administrador

Nombre – Administrador

Tipo – Principal

Roles – Actor que accede al módulo para utilizar la funcionalidad ofrecida por este. Podrá acceder a las tres secciones del módulo: configuración del canal, la configuración de generación del feed y la configuración de la sincronización del feed.

Comentarios –

Versiones –

Número de versión: 1.0

Fecha: 2014-11-05

Descripción de la modificación: Creación del actor

3.3. Casos de uso

A continuación se describen cada uno de los casos de uso identificados en el apartado 3.1 de este capítulo.

Un caso de uso es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable, de valor para un actor.

Cada uno de los casos de uso del modelo de casos de uso posee aquí los siguientes atributos:

Nombre – Nombre identificativo único del caso de uso.

Necesidad – Nivel de importancia del caso de uso:

Esencial – El sistema no es aceptable si no se implementa este caso de uso.

Deseable – Mejora el sistema producido pero no es un caso de uso esencial.

Opcional – Caso de uso propuesto por el proveedor como valor añadido.

Estado – Estado de aprobación del caso de uso:

Aprobado – El caso de uso ha sido aprobado por los interesados.

En estudio – El caso de uso está siendo estudiado por los interesados.

Rechazado – El caso de uso ha sido rechazado por los interesados.

Actor principal – Nombre del actor que inicia el caso de uso.

Objetivo – Metas de valor que persigue el actor con el caso de uso.

Precondiciones – Condiciones que se cumplen al inicio del caso de uso.

Postcondiciones – Condiciones que se cumplen al final del caso de uso.

Desencadenador – Acción o evento que inicia el caso de uso.

Flujo básico – Secuencia de acciones que constituyen el camino principal desde que se inicia el caso de uso hasta que se alcanzan los objetivos con éxito.

Flujos alternativos – Acciones que constituyen los caminos alternativos.

Requisitos – Requisitos no funcionales específicos del caso de uso.

Comentarios – Comentarios aclaratorios acerca del caso de uso.

Versiones – Historial de cambios del caso de uso. Cada registro de cambio incluye:

Número de versión

Fecha de modificación

Descripción de la modificación

3.3.1. Configuración canal

Nombre – Configuración canal

Necesidad - Esencial

Estado - Aprobado

Actor principal - Administrador

Objetivo – Gestionar los canales para los que se generarán feeds.

Precondiciones – El administrador ha accedido a la plataforma Hermes.

El administrador se ha autenficado con éxito en Hermes.

El administrador tiene permisos para acceder al módulo.

El administrador ha accedido a la sección Feed.

Postcondiciones – Se han guardado los cambios realizados sobre los canales.

Desencadenador – El adminsitrador accede a la sección Feed.

Flujo básico – 1. El administrador acceder a Hermes.

2. El administrador accede a Product Feed.

3. El administrador accede a configuración canal.

4. El administrador realiza cambios en los canales.

Flujos alternativos -

Requisitos -

Comentarios -

Versiones -

Número de versión: 1.0

Fecha: 2014-11-05

Descripción de la modificación: Creación del caso de uso

3.3.2. Configuración generación feed

Nombre – Configuración generación feed

Necesidad - Esencial

Estado - Aprobado

Actor principal - Administrador

Objetivo – Gestionar las propiedades de los feeds generados para el canal seleccionado.

Precondiciones – El administrador ha accedido a la plataforma Hermes.

El administrador se ha autenficado con éxito en Hermes.

El administrador tiene permisos para acceder al módulo.

El administrador ha accedido a la sección Product Feed.

Postcondiciones – Se han guardado los cambios realizados sobre las generaciones de feed.

Desencadenador – El adminsitrador accede a la sección Product Feed.

Flujo básico – 1. El administrador acceder a Hermes.

2. El administrador accede a Feed.

3. El administrador accede a configuración generación feed.

4. El administrador hace cambios en las generaciones de feed.

Flujos alternativos -

Requisitos -

Comentarios -

Versiones -

Número de versión: 1.0

Fecha: 2014-11-05

Descripción de la modificación: Creación del caso de uso

3.3.3. Configuración listado feed

Nombre – Configuración listado feed

Necesidad - Esencial

Estado - Aprobado

Actor principal - Administrador

Objetivo – Gestionar los feeds actuates del sistema.

Precondiciones – El administrador ha accedido a la plataforma Hermes.

El administrador se ha autenficado con éxito en Hermes.

El administrador tiene permisos para acceder al módulo.

El administrador ha accedido a la sección Product Feed.

Postcondiciones –

Desencadenador – El adminsitrador accede a la sección Product Feed.

Flujo básico – 1. El administrador acceder a Hermes.

2. El administrador accede a Feed.

3. El administrador accede a configuración generación feed.

4. El administrador hace cambios en las sincronizaciones de feed.

Flujos alternativos -

Requisitos -

Comentarios -

Versiones -

Número de versión: 1.0

Fecha: 2014-11-05

Descripción de la modificación: Creación del caso de uso

4. Diagramas de flujo

Los diagramas de flujos de datos (DFD) son una técnica de modelización, que nos muestra un sistema como una red de procesos conectados entre ellos por flujos y almacenamientos de datos.

Es un modelo que proporciona el punto de vista funcional de un sistema.

Se va a seguir la notación que se enseñó en la asignatura Ingeniería del Software 1 (Figura 15).

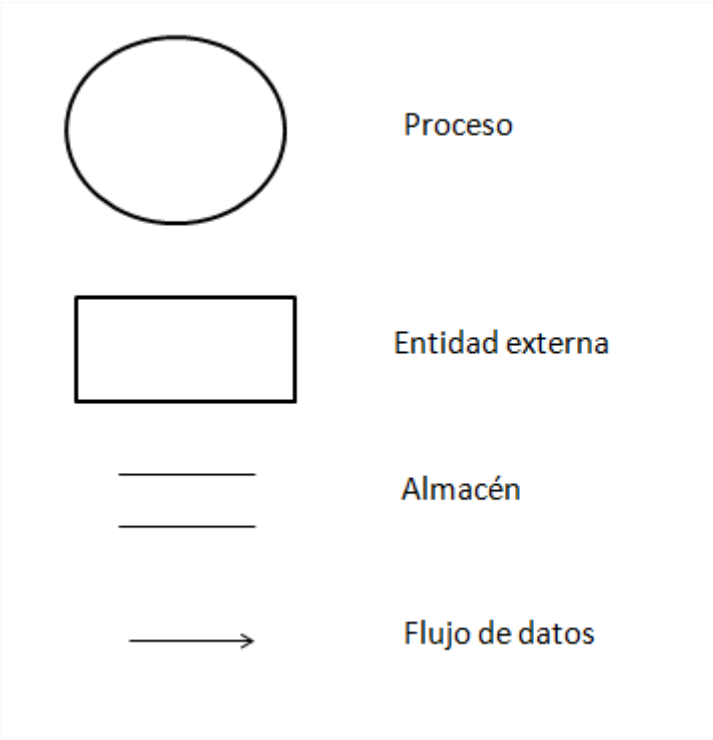


Figura 15. Leyenda elementos DFD

4.1. DFD Nivel 0

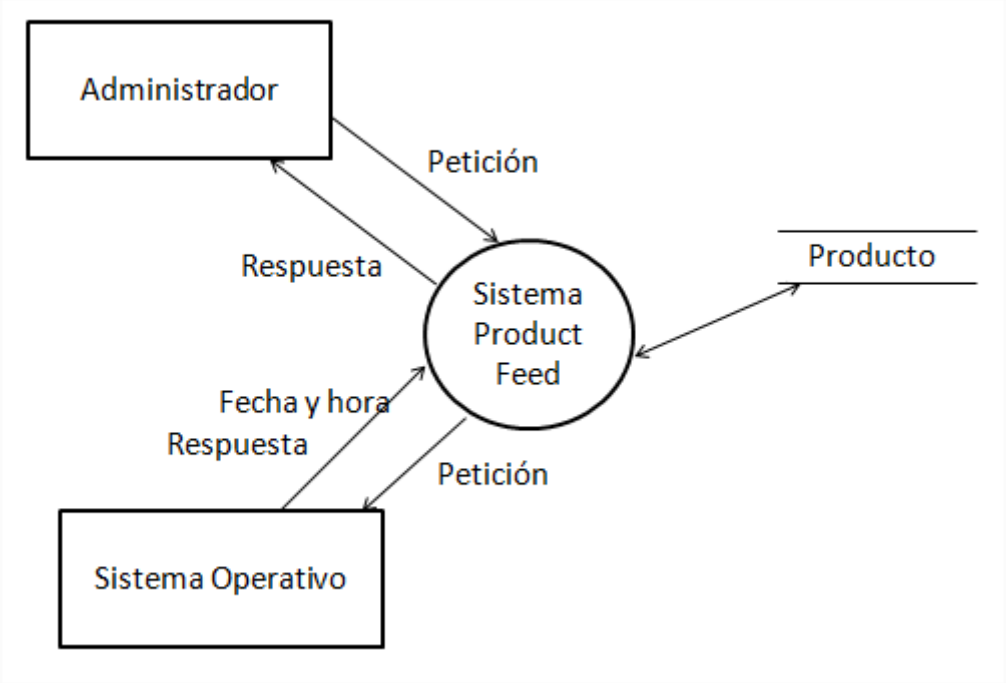


Figura 16. DFD Nivel 0

El diagrama de flujo de datos de nivel 0 sirve para hacer una representación muy genérica del sistema.

En este caso se puede observar como interactúan dos sitemas externos con el sistema de Product Feed. Estos dos sistemas son el Administrador que vaya a tratar con la aplicación y el Sistema Operativo de la máquina dónde esté implantada la aplicación web.

Además el sistema va a requerir de un almacén de datos de productos que corresponde a la información necesaria que precisará para generar los ficheros de product feed.

4.2. DFD Nivel 1

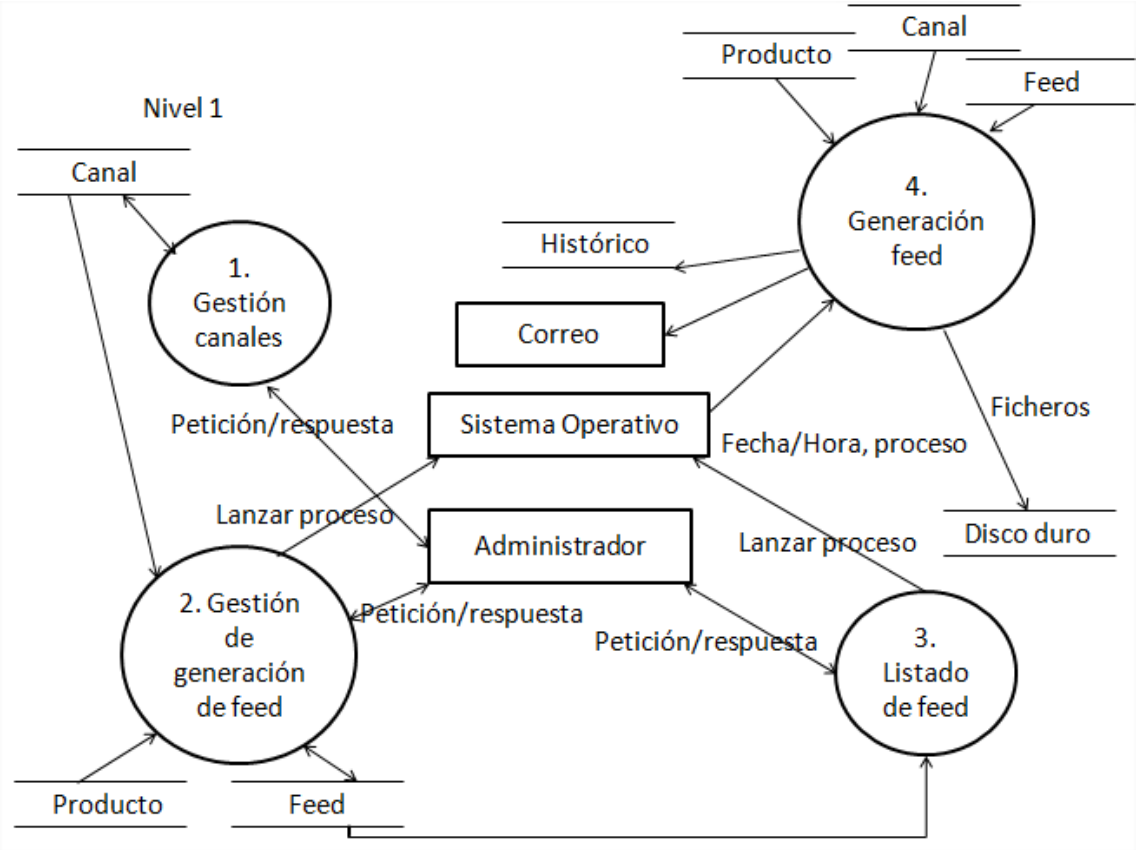


Figura 17. DFD Nivel 1

En el diagrama de flujo de datos de nivel 1 se ha dividido el sistema de product feed en cuatro procesos.

El primer proceso, Gestión canales, será el encargado de gestionar lo relacionado con los canales, es por eso que escribirá y leerá del almacén de canales cuando reciba peticiones del administrador y responderá tras manejar la operación pedida.

El segundo proceso, Gestión de generación de feed es el encargado de configurar la generación del feed. Necesita leer y escribir del almacén de feed, y leer de canal. También necesita leer del almacén de producto para mostrar las diferentes opciones de elegir que productos irán en el feed. Una vez haya atendido las peticiones del administrador y se haya configurado la generación de un feed por completo, informará sistema operativo de lanzar un proceso.

El tercer proceso, listado de feed informará al administrador del listado de feeds configurados actualmente. Además, como se ha descrito anteriormente, desde este módulo se puede lanzar manualmente la generación de un feed, por lo que deberá informar al sistema operativo de que se debe generar un proceso programado.

El cuarto proceso, Generación feed, es el encargado de generar los ficheros. Es lanzado por el sistema operativo y requiere para ello leer información de los almacenes feed, canal y producto y escribe en el disco duro del sistema. Además escribe en el almacén de histórico cada vez que genera

un fichero, y requiere de la hora del sistema cuando ha habido un error durante la generación y poder así enviar al correo la incidencia.

4.3. DFD Nivel 2

4.3.1. DFD Nivel 2 – Gestión canales

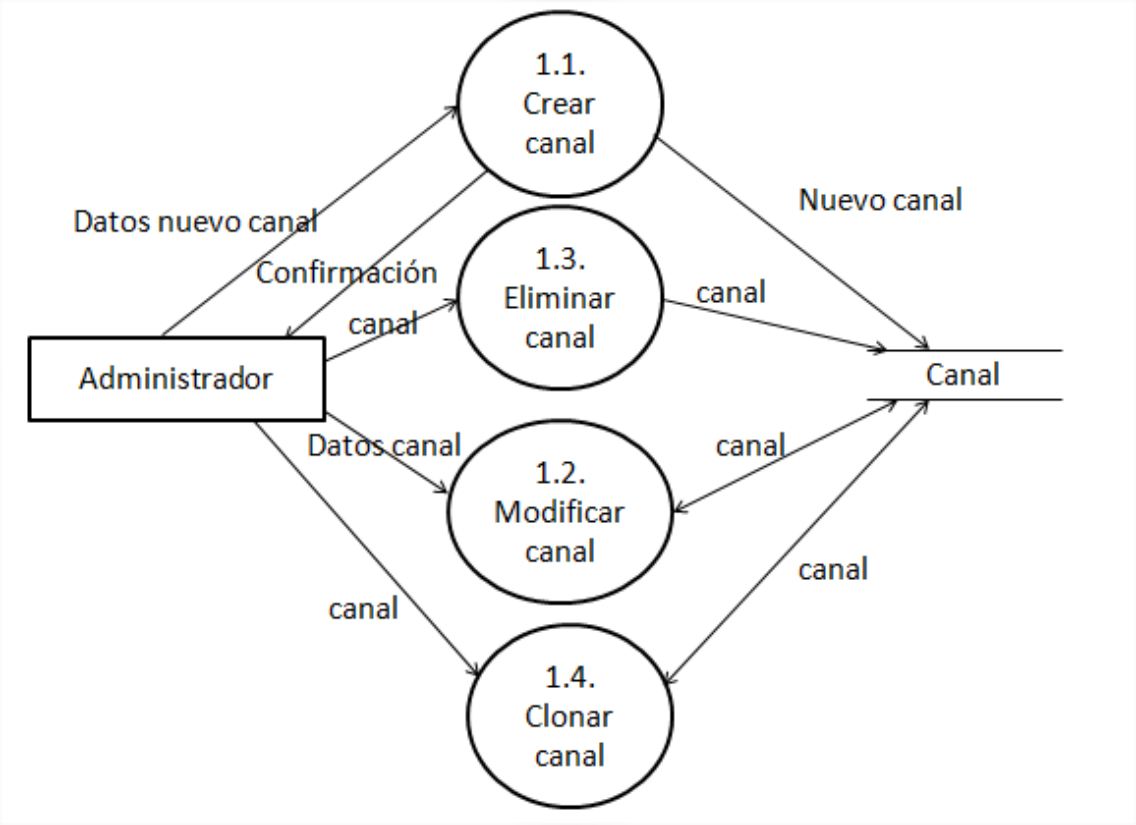


Figura 18. DFD Nivel 2 – gestión canales

En el DFD de nivel 2 – Gestión de canales se observan los procesos internos necesarios para la gestión de canales.

El proceso crear canal recibe datos del administrador para crear un nuevo canal y los escribe en el almacén Canal.

El proceso eliminar canal recibe que canal quiere el administrador borrar y lo escribe en el almacén.

El proceso modificar canal recibe del administrador datos de un canal modificado y consulta en el almacén ese canal que se quiere cambiar e inscribe los cambios.

El proceso clonar canal recibe del administrador el canal que se quiere clonar y lo duplica en el almacén.

4.3.2. DFD Nivel 2 – Gestión de generación del feed

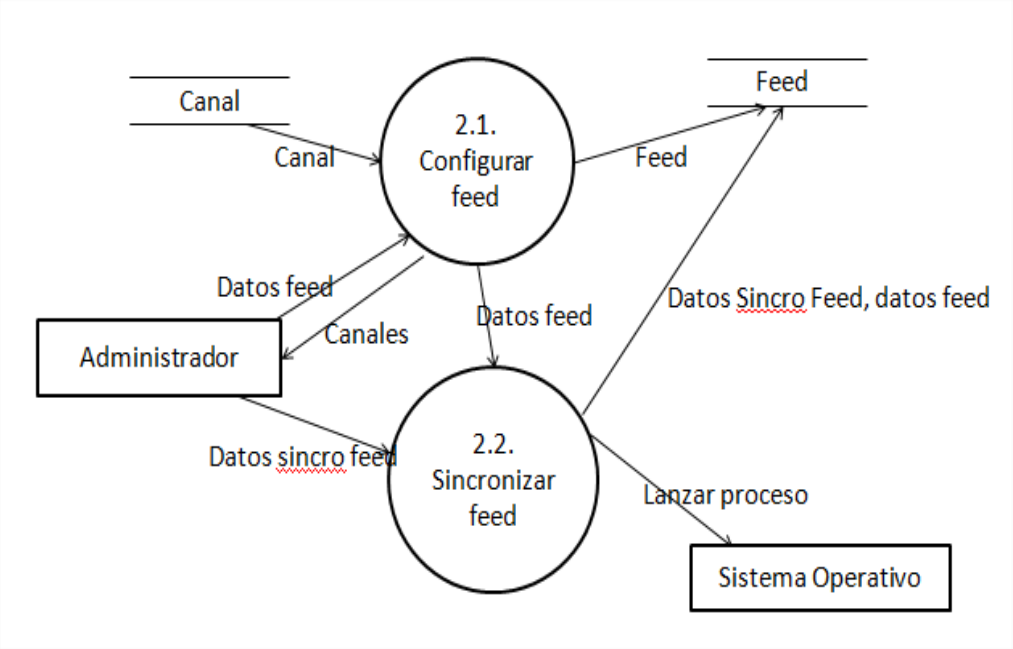


Figura 6. DFD Nivel 19 – gestión de generación del feed

El proceso configurar feed recibe la información del nuevo feed a crear del administrador y lo escribe en el almacén feed. Necesita información de canal para mostrar al administrador los canales disponibles.

El proceso sincronizar feed recibe nueva información de sincronización del administrador más la información del proceso anterior. Escribe la nueva información en el almacén feed y informa al sistema operativo de cuando tendrá que lanzar el proceso de generación del feed.

4.3.3. DFD Nivel 2 – Listado de feed

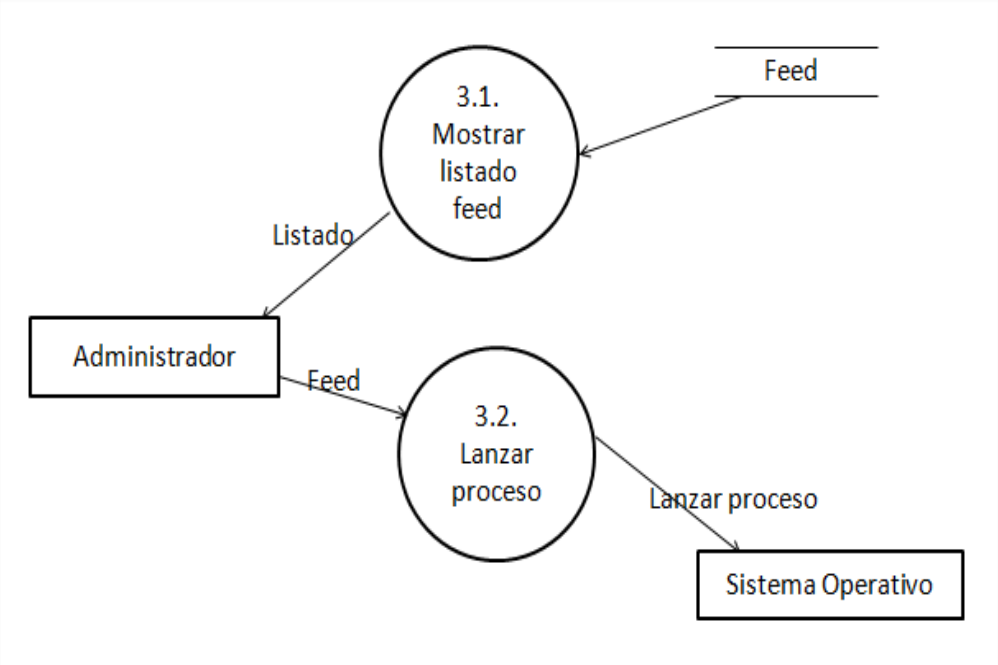


Figura 20. DFD Nivel 2 – Listado de feed

El proceso mostrar listado feed muestra al administrador el listado total de los feeds configurados en el sistema.

El proceso lanzar proceso recibe información del administrador de qué feed se debe lanzar e informa al sistema operativo para que lo programe.

4.3.4. DFD Nivel 2 – Generación feed

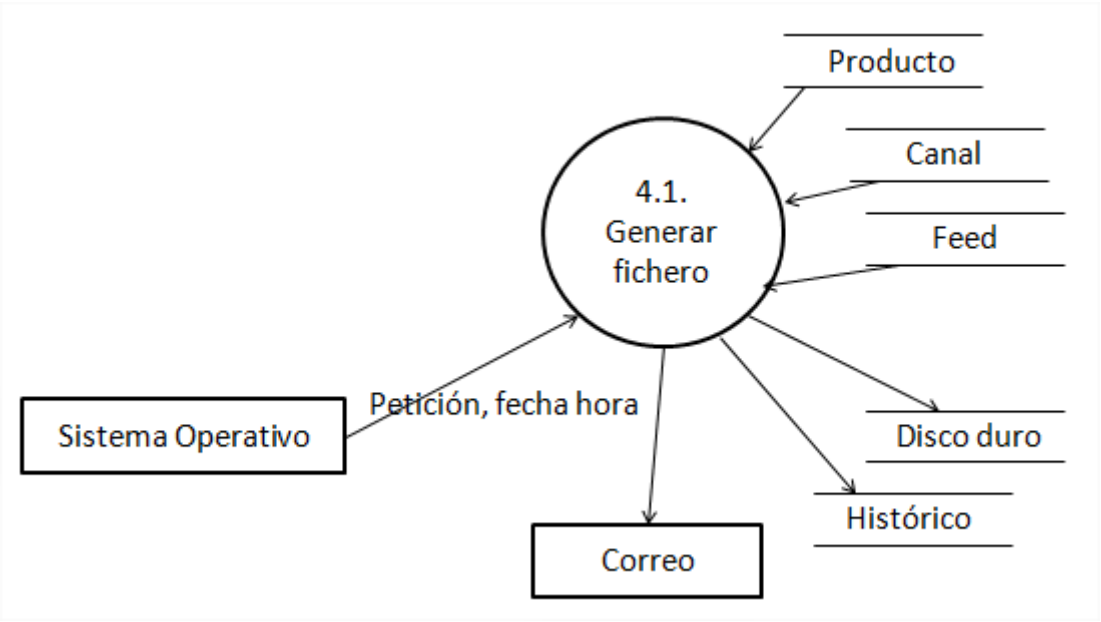


Figura 21. DFD Nivel 2 – Generación feed

El proceso generar fichero recibe la orden de ejecutarse del sistema operativo. Lee datos de los almacenes producto, canal y feed para saber que fichero tiene que generar y escribe el resultado en el disco duro y en el almacén histórico. Si ocurre algún error se comunica por correo electrónico.

ANEXO B. DISEÑO DEL SISTEMA

1. Introducción

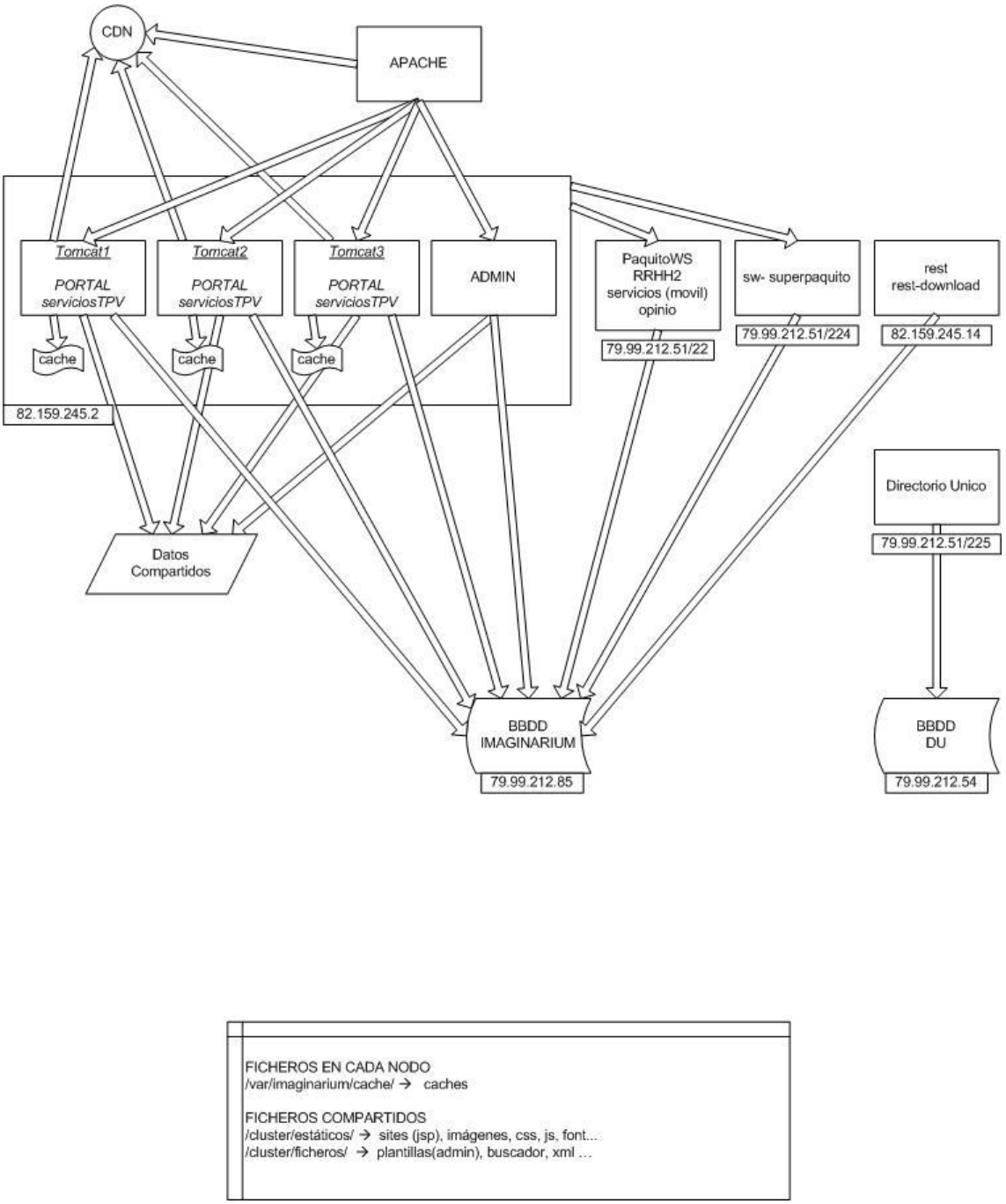
El propósito de este documento es detallar el diseño del sistema Product Feed partiendo del análisis realizado previamente. El diseñó representa una solución abstracta de la estructura que tendrá el sistema sin entrar en detalles de implementación que se tendrán en cuenta en fases posteriores al diseño.

En este documento está compuesto por tres apartados sin contar la introducción:

- El diseño de la arquitectura sobre la que se va a soportar el sistema.
- El diseño funcional del sistema representado a través de la herramienta gráfica Diagrama de estructura de cuadros de Constantine.
- El prototipado de las ventanas de la interfaz gráfica y la navegación a través de las mismas.

2. Diseño de la arquitectura física

La arquitectura en la que se va a desplegar el sistema viene totalmente condicionada por el sistema en el que está montado la aplicación Hermes, cosa lógica ya que es un módulo del mismo. Es por eso que en este aspecto no se ha tenido que barajar ninguna otra posibilidad.



FICHEROS EN CADA NODO
/var/imaginarium/cache/ → caches

FICHEROS COMPARTIDOS
/cluster/estáticos/ → sites (jsp), imágenes, css, js, font...
/cluster/ficheros/ → plantillas(admin), buscador, xml …

Figura 22. Arquitectura sistema Imaginarium

La aplicación Hermes reside en la máquina ADMIN, corriendo en un servidor web Tomcat. Las peticiones del cliente se lanzan contra la máquina que contiene el servidor Web APACHE que está configurado para redirigir las peticiones a las máquinas que correspondan.

En nuestro caso el Apache redirigirá todas nuestras peticiones hacía la máquina ADMIN ya que es la única que contiene la aplicación de Hermes. Para acceder al portal de Imaginarium en cambio, la aplicación del portal está en tres máquinas diferentes para balancear la carga, pero debido a que la administración no tiene mucha carga solo se precisó una máquina para ello.

Para liberar las máquinas con aplicaciones corriendo, la máquina que contiene el APACHE tiene acceso al módulo CDN, que no es más que un servidor de contenido estático (imágenes, ficheros javascript…).

A su vez, la máquina admin, a la vez que las máquinas que contiene el portal de Imaginarium acceden a una misma base de datos donde reside toda la información de los productos de Imaginarium entre otras cosas, y ahora además, la información de los canales, los feeds y todas las tablas creadas para el sistema Product Feed.

3. Diagrama de estructura de cuadros

El diagrama de estructura de cuadros (o DDE) es una herramienta útil para representar el diseño de un sistema. Sirve para obtener una estructura modular y los detalles de proceso del sistema partiendo de los "productos" obtenidos en la fase de análisis del sistema.

Se va a seguir la notación impartida en la asignatura Ingeniería del software para desarrollar los diagramas de estructura de cuadros.

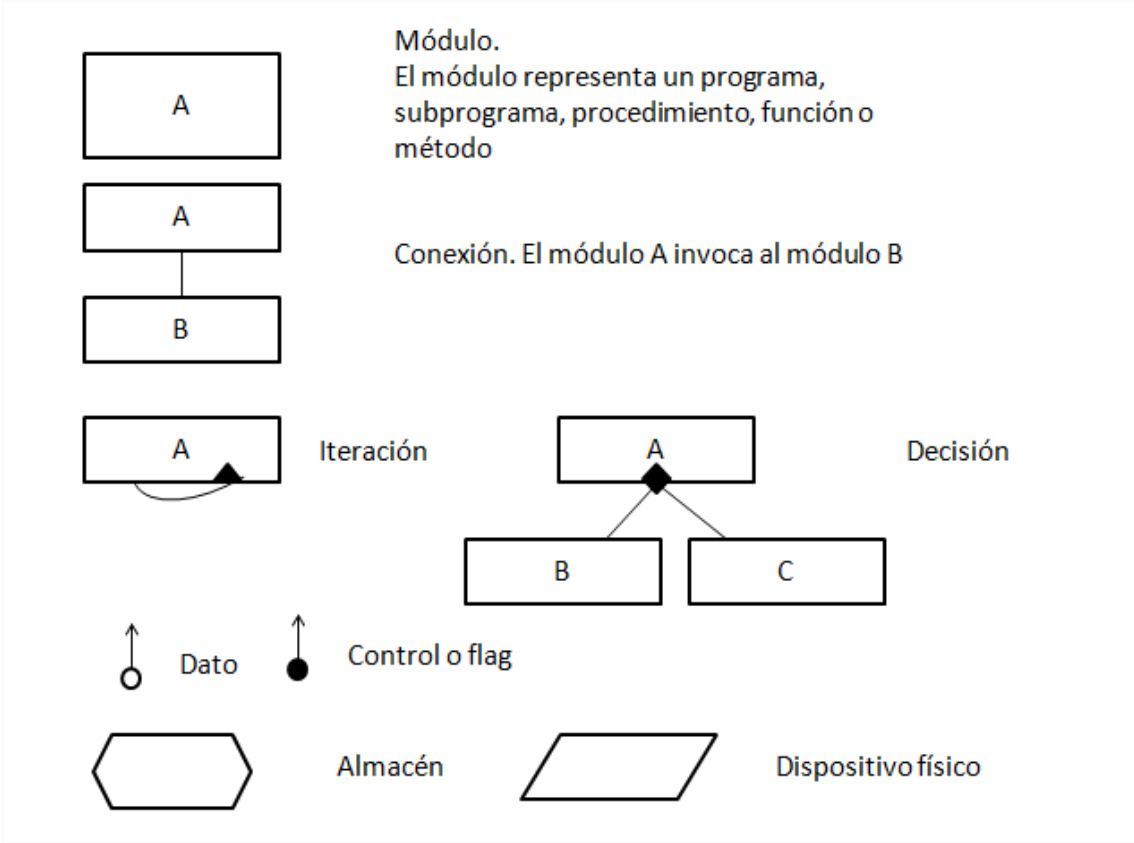


Figura 23. Leyenda DDE

3.1. DDE Gestión de canales

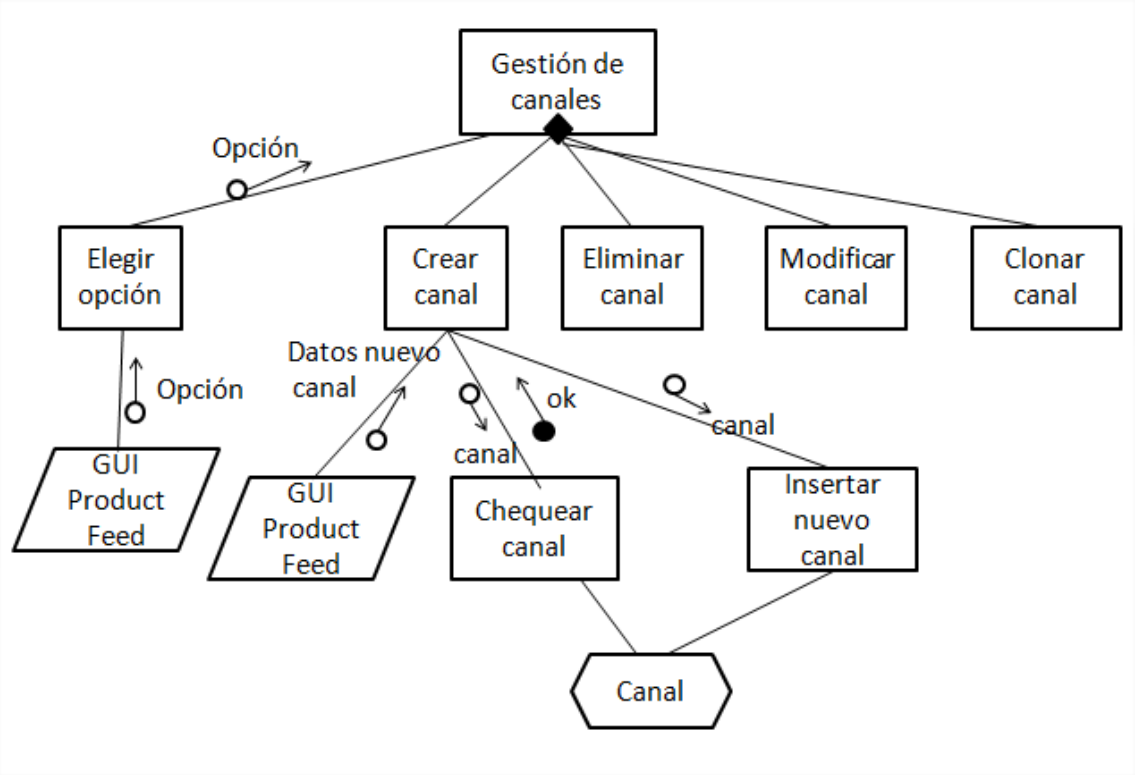


Figura 24. DDE gestión de canales

A través de la interfaz gráfica se seleccionará una opción entre crear, eliminar, modificar o clonar un canal. Cuando se selecciona crear canal se requiere de información acerca del nuevo canal (nombre, atributos…). Crear canal consulta en el almacén si ya existía un canal con el mismo nombre. Si no existía procede a insertar el nuevo canal en el almacén.

3.2. DDE Gestión de generación del feed

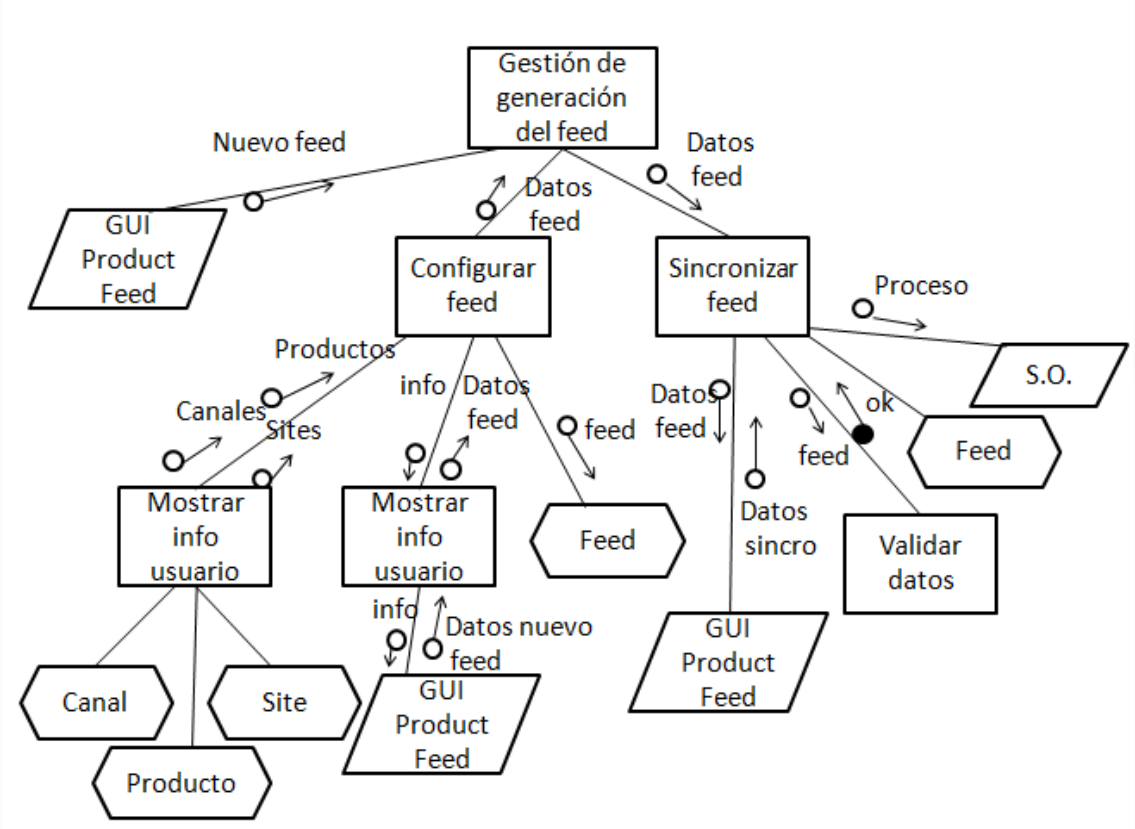


Figura 25. DDE gestión de generación de feed

A través de la interfaz gráfica el usuario accede a la pantalla de generación de feed. El módulo configurar feed requiere de canal, site y producto para mostrar la información a elegir al usuario. El usuario selecciona la información nueva acerca del nuevo feed a crear y lo escribe en el almacén feed y comunica al siguiente módulo: sincronizar feed.

El módulo sincronizar feed requiere de información de la frecuencia de generación del usuario. Comprueba que los datos son correctos y escribe la nueva información en el almacén feed. Además genera un nuevo proceso en el sistema operativo.

3.3. DDE Listado de feed

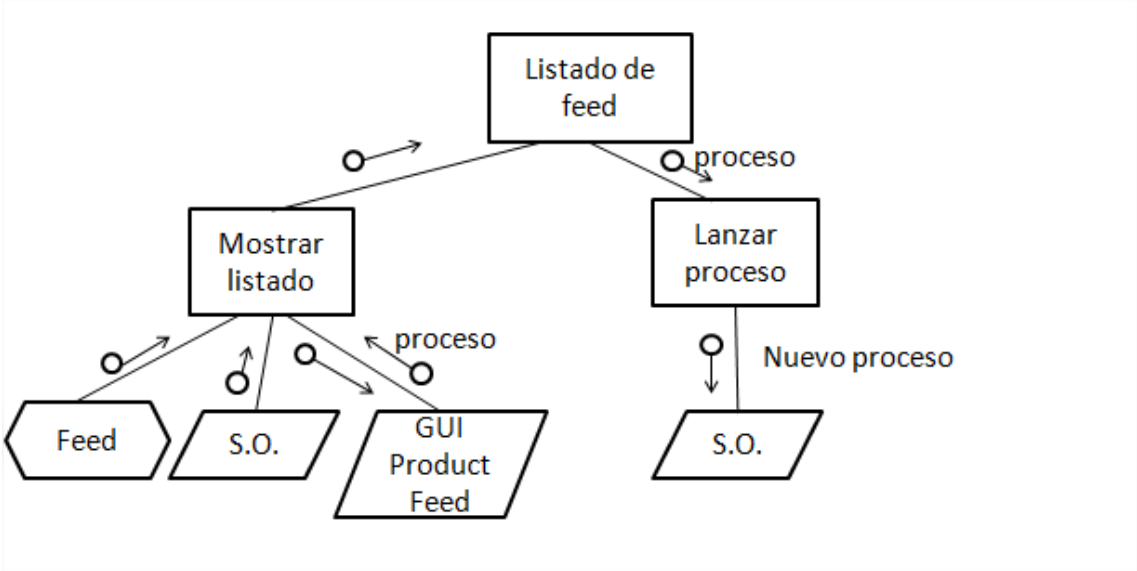


Figura 26. DDE listado de feed

El módulo listado de feed requiere de información del almacén feed para mostrar el listado de feeds. Además requiere saber de los procesos del sistema operativo que estén generando un feed en ese momento para mostrar la barra de progreso al usuario. El usuario a través de la interfaz gráfica puede lanzar un proceso manualmente. Este proceso se le tiene que comunizar al sistema operativo.

3.4. DDE Generación feed

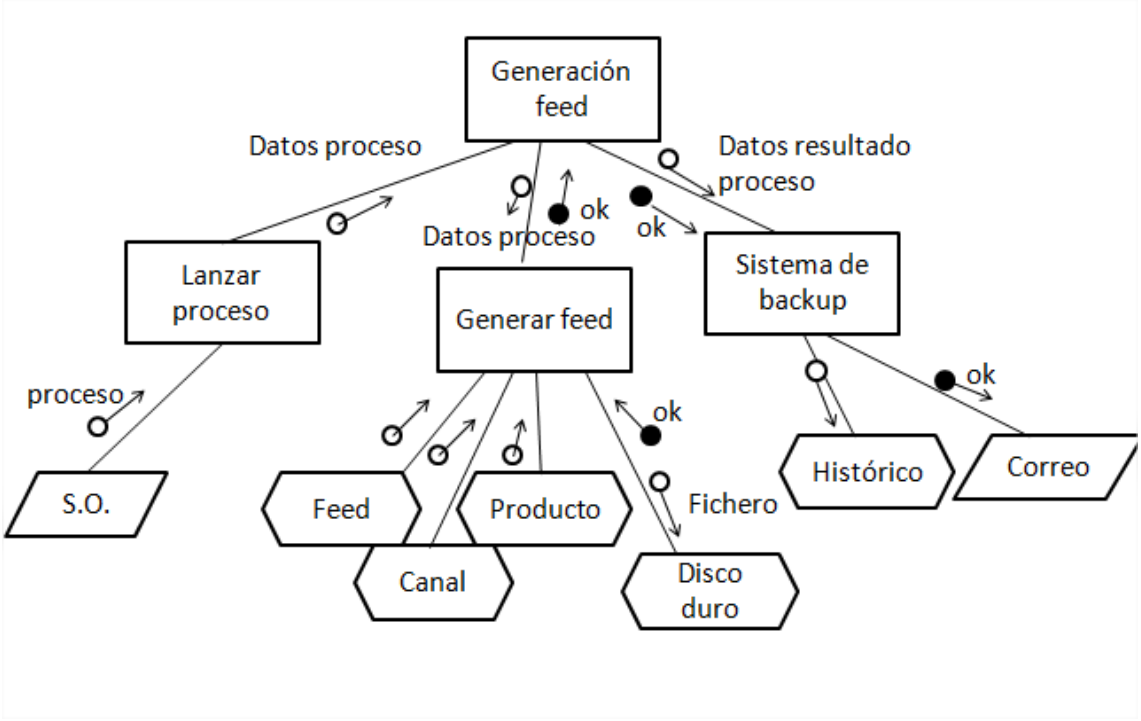


Figura 27. DDE generación feed

El módulo de generación feed se activa cuando un proceso del sistema operativo generado por los módulos previos se activa. Cuando se activa, se ejecuta el módulo generar feed que requiere información de los almacenes feed, canal y producto. Una vez ha generado el fichero lo escribe en el disco duro. Envía un flag al sistema de backup con el resultado de la generación del fichero. El sistema de backup escribe en el almacén histórico los datos del proceso y envía un mail en caso de que haya ocurrido un error.

4. Prototipado de ventanas y navegación

En este apartado se va a mostrar el prototipado de ventanas que se desarrolló previo a la implementación para diseñar la aplicación desde el punto de vista del usuario. Los prototipos se han hecho con un programa de *wireframing* llamado Balsamiq Mockups.

Figura 28. Listado de canales

En la figura 28 se muestra el prototipo de la ventana de listado de canales. En la parte de la izquierda aparece el menú de Hermes con el nuevo módulo Product Feed. Al listado de canales se accede a través del botón del menú “Canal”. En la figura 6 aparece el listado de canales junto con los botones “Crear”, “Modificar”, “Eliminar” y “Clonar”.

Para acceder a la ventana de creación/modificación de canales (Figura 7) se debe hacer click en el botón “Crear” o en el botón “Modificar”.

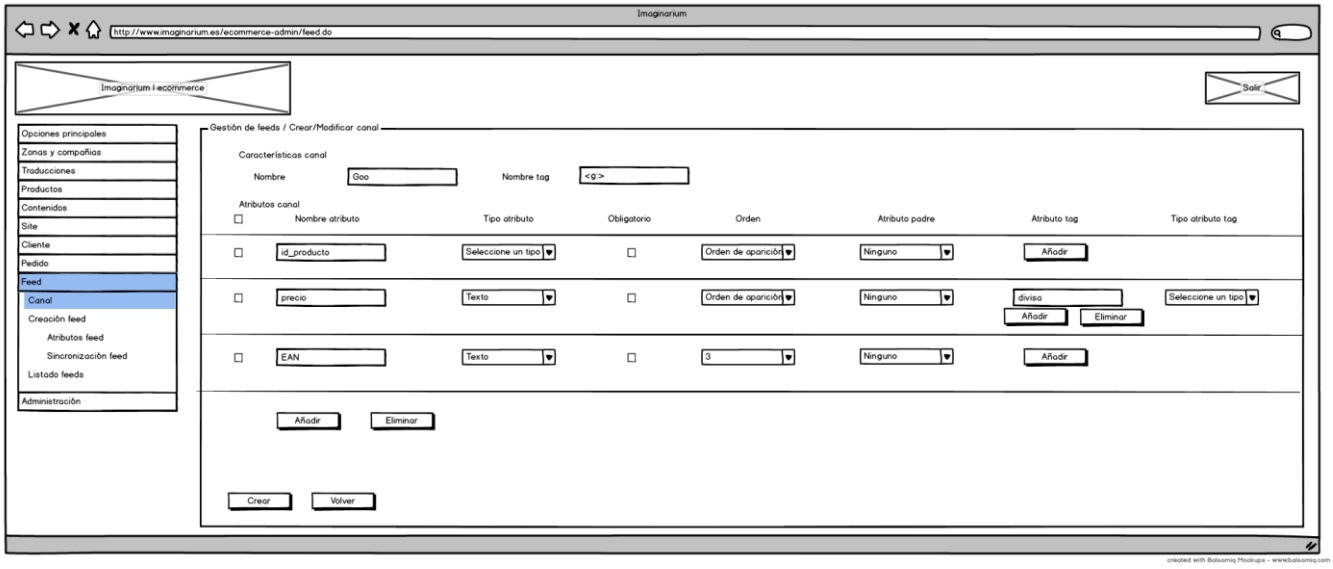


Figura 29. Creación/modificación de canales

En la figura 29 se configuran las características de cada canal, así como sus atributos. Para añadir más atributos al canal se debe pulsar el botón “Añadir” y para eliminar atributos el botón “Eliminar”. Pulsando el botón “Crear” guardaremos los cambios realizadas e iremos de vuelta a la pantalla de listado de canales (Figura 6).

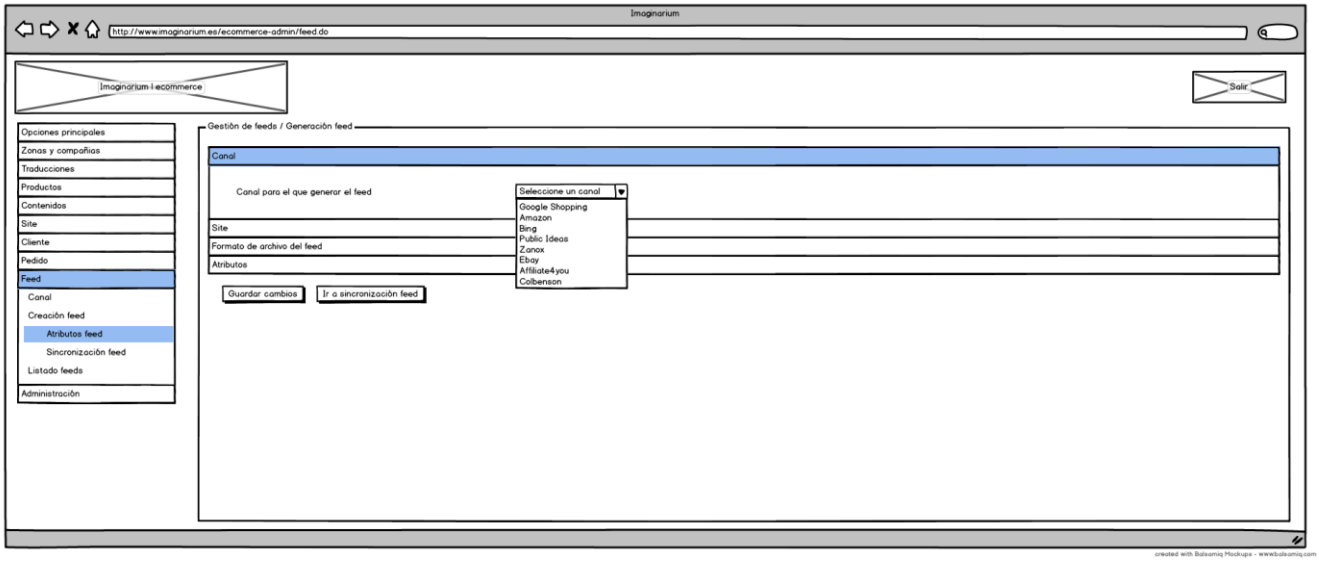


Figura 30. Generación de feed - Canal

Para acceder a la generación de feeds se debe pulsar el botón “Atributos Feed” del menú principal. Lo primero que debemos seleccionar para generar un feed es el canal para el que queremos que se genere como se muestra en la Figura 30.

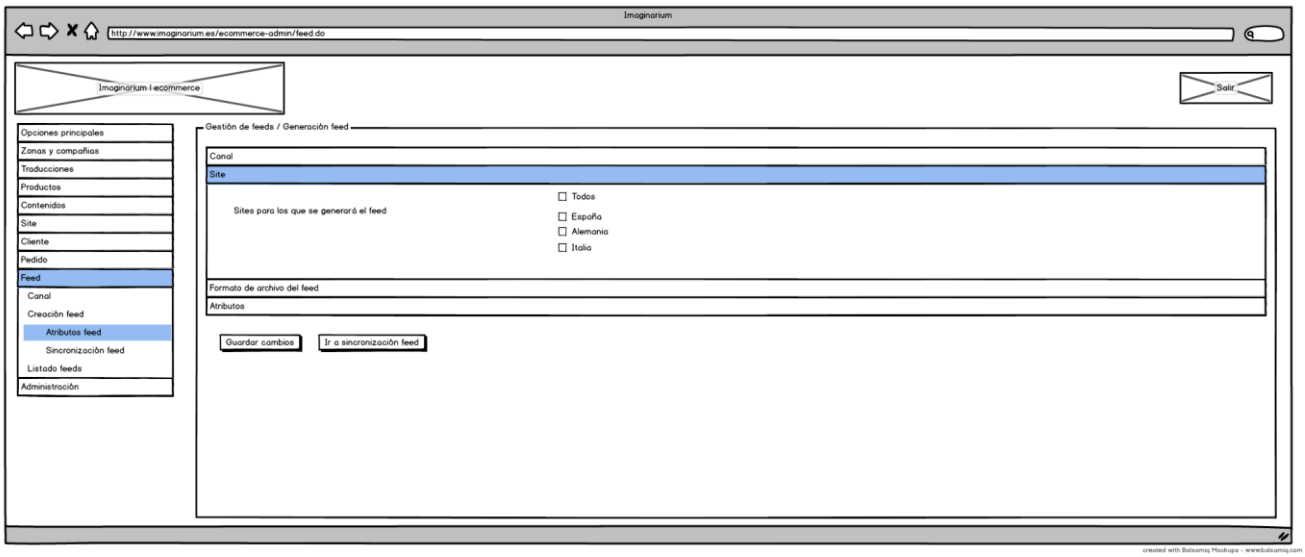


Figura 31. Generación de feed - Site

En la figura 31 podemos elegir marcando en el checkbox los sites para los que queremos que se genere el feed. Como mínimo un site debe ser seleccionado.

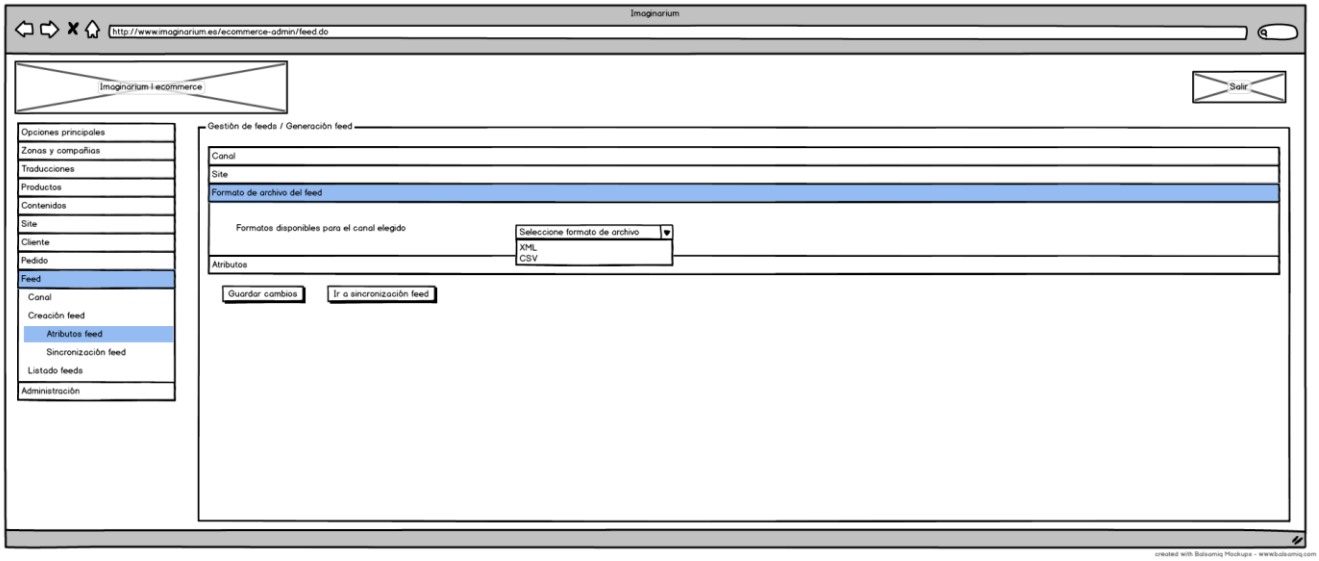


Figura 32. Generación de feed - Formato

En la figura 32 podemos elegir del select el formato de archivo que queremos que sea el fichero generado.

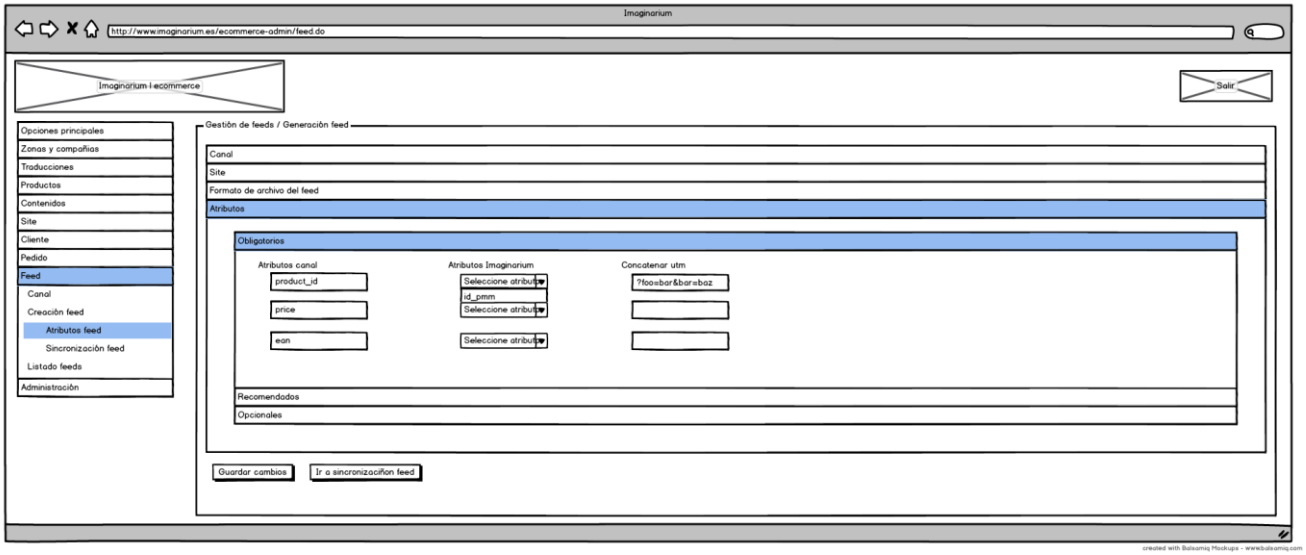


Figura 33. Generación de feed - Atributos

En la figura 33 aparecen los atributos del canal que hemos elegido. Para cada atributo definido en la sección de canal, debemos seleccionar que valor va a tener en el fichero, seleccionando un atributo interno de la base de datos de Imaginarium.

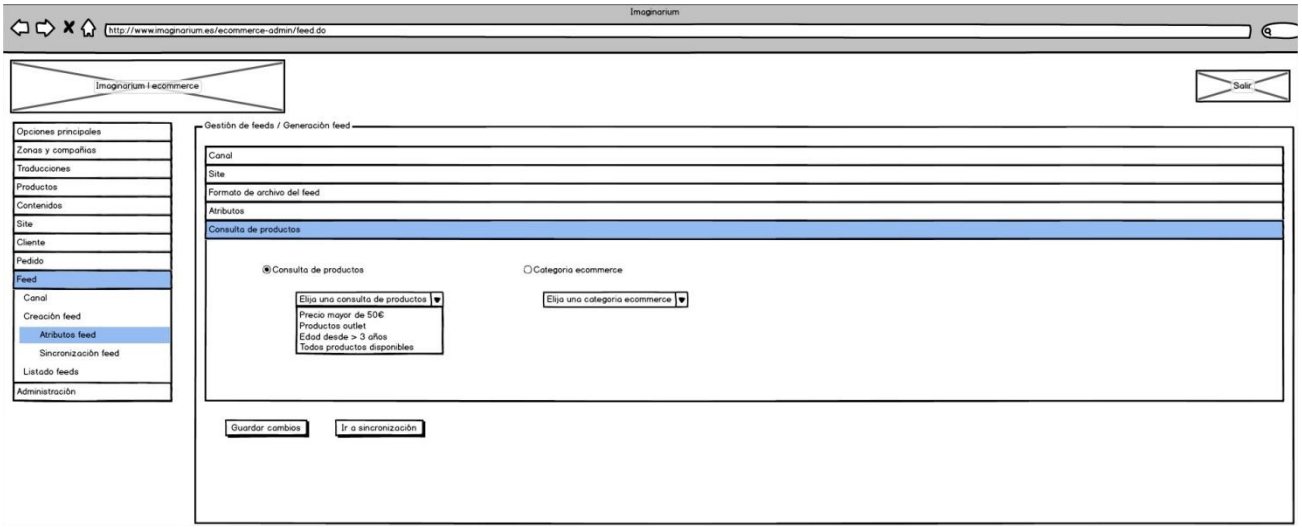


Figura 34. Generación de feed - Productos

En la figura 34 se elegirá que productos aparecerán en el feed. Se debe elegir entre sacar los productos a través de una consulta de productos definida en Hermes o a través de los productos contenidos en una categoría.

Una vez realizadas estas configuraciones si se pulsa el botón “Ir a sincronización” iremos a la siguiente pantalla (Figura 35):

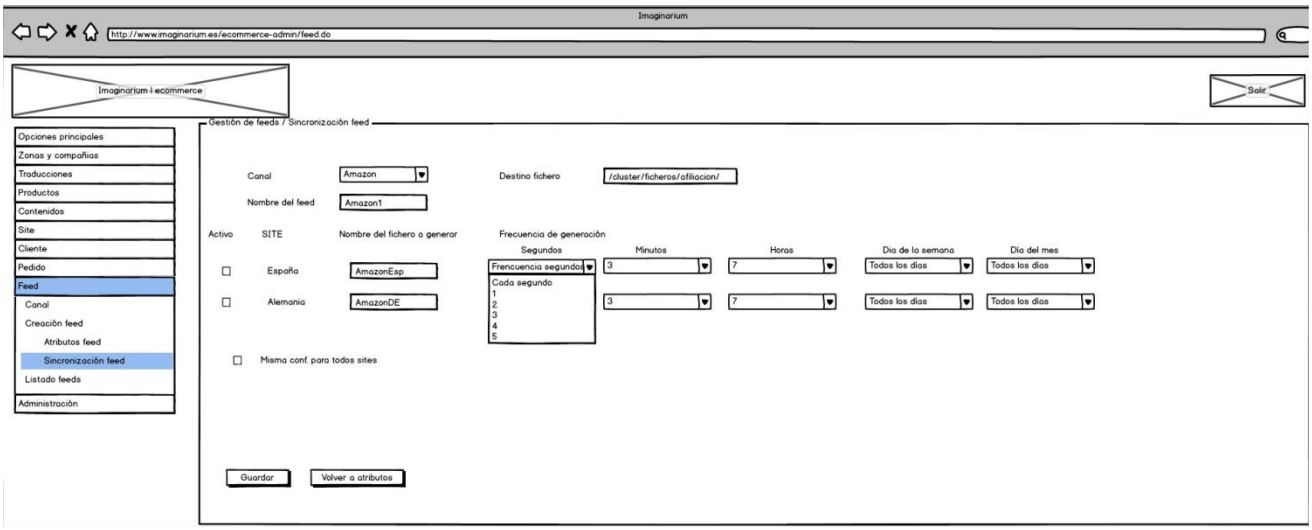


Figura 35. Sincronización feed

En la figura 35 nombrará el feed. Además se definirá la ruta dónde se depositarán los archivos generados para dicho feed.

Adicionalmente, para cada site seleccionado previamente, se nombrará el fichero resultante y se definirá la frecuencia de generación del archivo. Una vez terminado de configurar el feed, si pulsamos el botón guardar iremos a la siguiente pantalla (Figura 36):

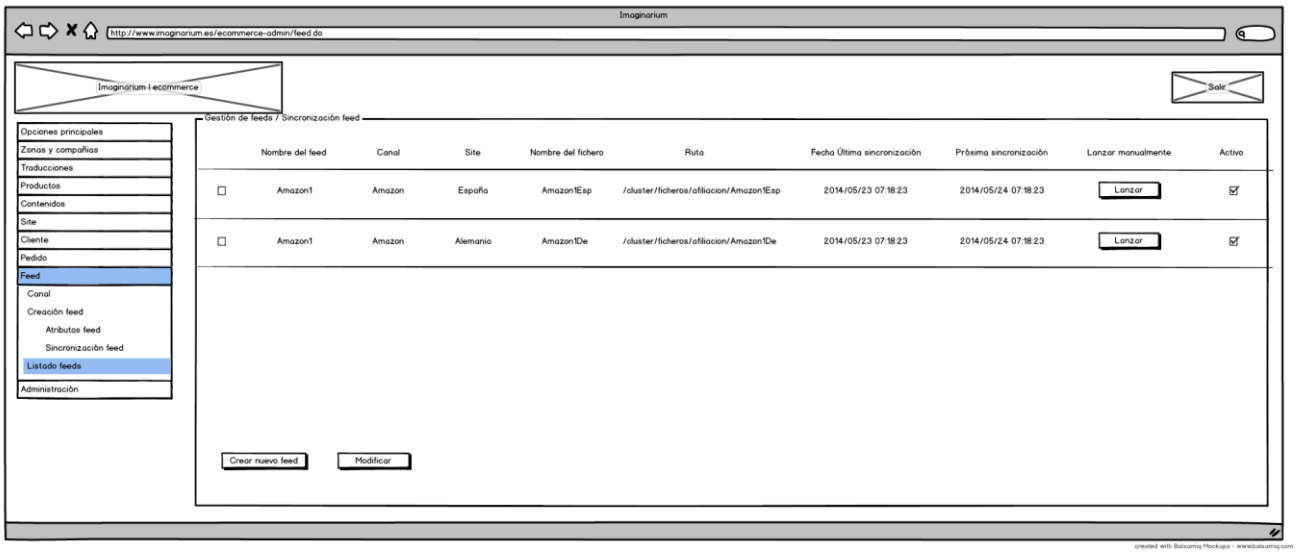


Figura 36. Listado feeds

En la figura 36 podemos observar el listado de los feeds que se tienen configurados. También se puede acceder a esta pantalla a través del menú principal, en el botón “Listado feeds”.

Desde esta pantalla podemos observar toda la información de cada feed. Además podemos activar o desactivar algún feed según convenga o

lanzar el proceso de generación de archivo manualmente con el botón “Lanzar”.

Si se pulsa un feed o el botón “Crear nuevo feed” nos llevará a la generación de feed.

ANEXO C. BASE DE DATOS

1. Introducción

En este anexo se va a describir la estructura a nivel de base de datos generada para sostener el sistema Product Feed.

No se ha precisado de inicializar una base de datos nueva ya que las tablas generadas necesitan tener relaciones con algunas de las tablas ya existentes en la base de datos que usa Imaginarium para hacer funcionar su página web.

El proceso ha sido costoso y ha requerido un estudio previo para familiarizarse y amoldarse con la estructura de base de datos ya existente.

EL sistema gestor de la base de datos es SQL Server. Para generar sentencias en la base de datos se ha utilizado el gestor HeidiSQL y la aplicación Product Feed ataca la base de datos a través de Hibernate. Se hablará más en detalle de esto en el Anexo D – Tecnologías utilizadas.

A continuación se describirá el modelo entidad relación generado.

2. Modelo entidad relación

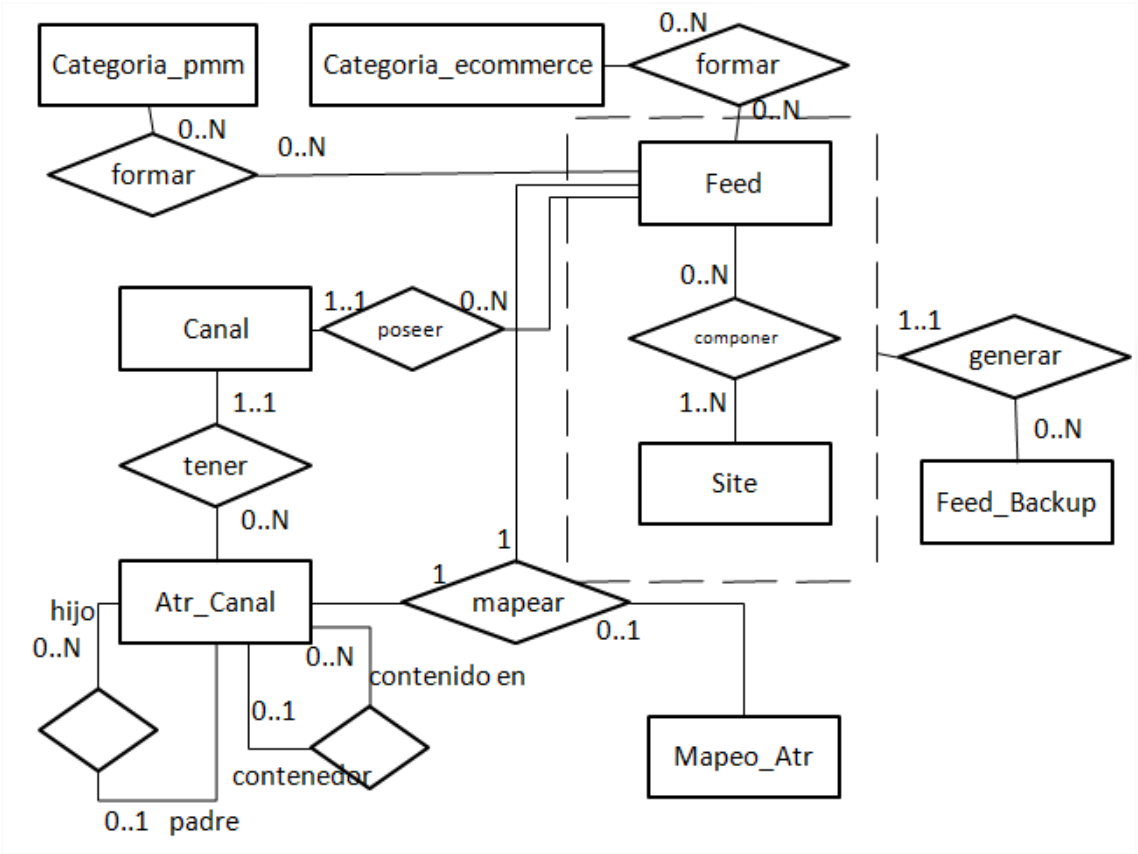


Figura 37. Modelo entirdad-relación Product Feed

Este es el esquema de las relaciones de los tipos de entidad que intervienen y sus cardinalidades. Los tipos de entidad Site, Categoria_pmm y Categoria_Ecommerce ya existían previamente en la base de datos de Imaginarium. A continuación se mostrarán los atributos de cada tipo de entidad y sus relaciones más en detalle.

Este es el tipo de entidad Canal (Figura 38). Aquí es dónde se almacenará cada canal. Los canales se identificarán por un número natural que se autogenerará cada vez que se inserte un canal (id_canal). Además, para cada canal se guardarán los distintos atributos ya mencionados en apartados anteriores.

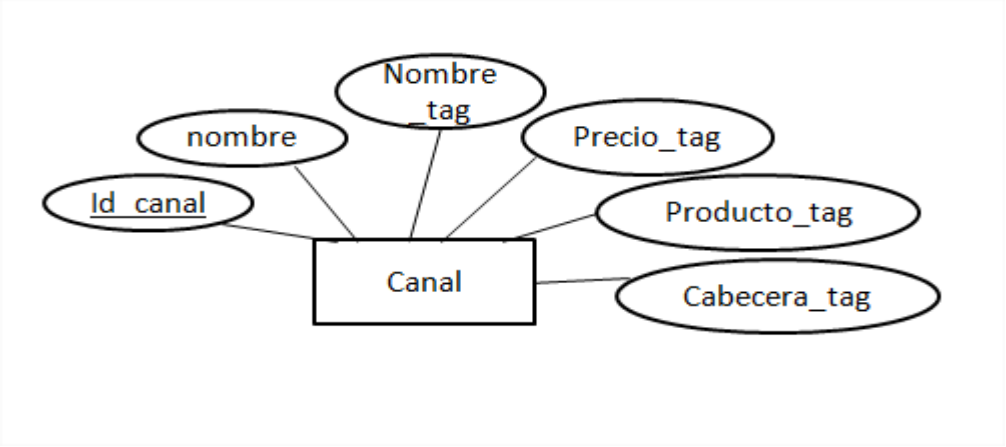


Figura 38. Tipo de entidad canal

Este es el tipo de entidad Atr_Canal (Figura 39). En él se almacenarán los atributos de cada canal. Se ha decidido crear un nuevo tipo de entidad en vez de mantenerlo como un atributo multivaluado de Canal ya que se necesitaba almacenar información adicional acerca de cada atributo del canal. La cardinalidad entre Canal y Atr_Canal es 1:N ya que cada canal puede tener N atributos pero cada Atr_Canal solo pertenece a un Canal. En un principio se pensó en si era viable que la relación fuese N:N pero posteriormente se descartó ya que si un Atr_Canal pertenecía a dos Canales, habría problema si se quisiese cambiar el orden de ese Atr_Canal para un Canal y no para el otro.

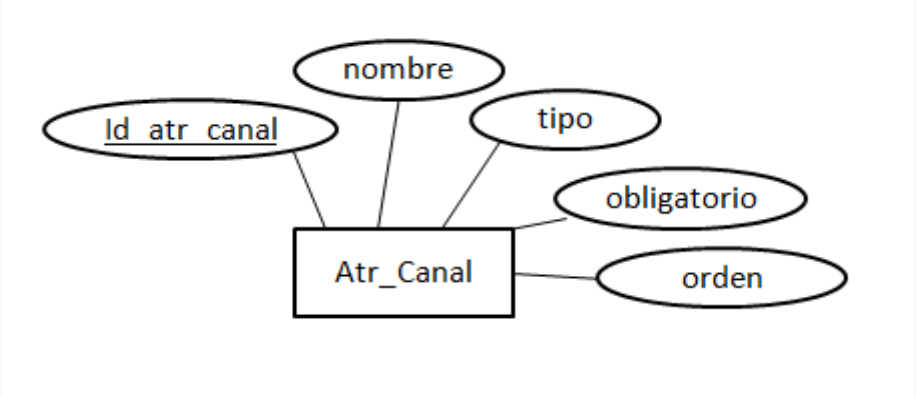


Figura 39. Tipo de entidad atr_canal

El tipo de entidad Atr_Canal (Figura 40) además posee dos relaciones consigo mismo. Una relación es para representar la relación ente hijos y padres y la otra entre atributos de canal contenidos en otros atributos. Ambas relaciones son 1:N ya que cada atributo canal solamente puede tener un atributo padre o estar contenido en otro atributo pero a su vez puede tener N atributos hijos y N atributos contenidos. Aquí debería especificarse una restricción que sea que si un atributo tiene un padre no puede estar contenido en otro atributo y viceversa.

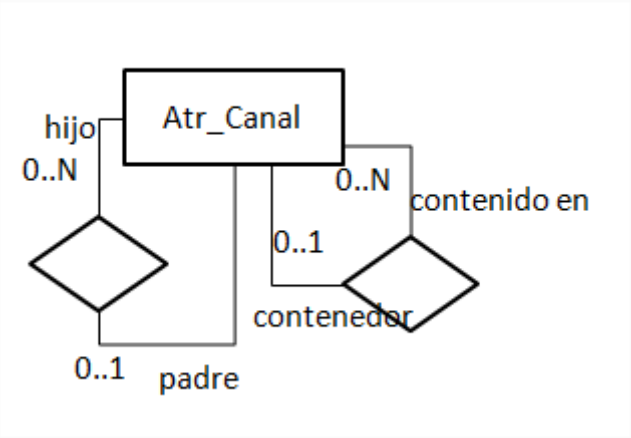


Figura 40. Relaciones Atr_Canal

Este es el tipo de entidad feed (Figura 41). Aquí es dónde se almacenará cada feed. Los feeds se identificarán por un número natural que se autogenerará cada vez que se inserte un feed (id_feed). Además, para cada feed se guardarán los distintos atributos ya mencionados en apartados anteriores.

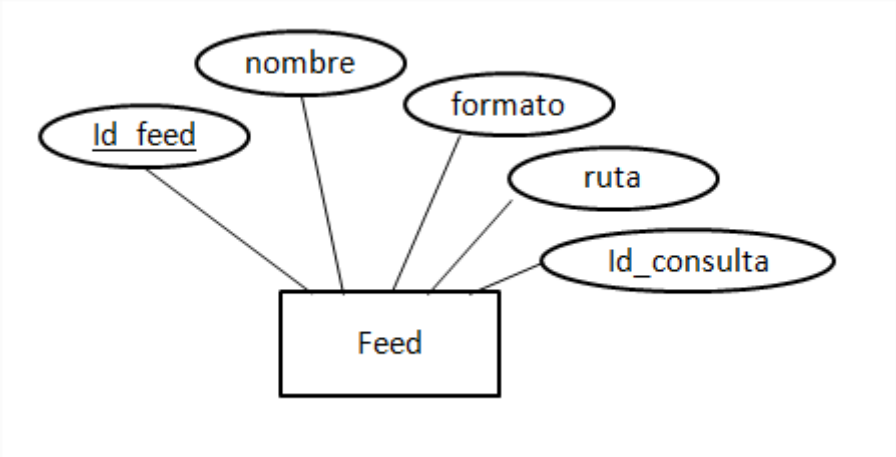


Figura 41. Tipo de entidad feed

El tipo de entidad Feed se relaciona con el tipo de entidad Site de Imaginarium con cardinalidad N:N (Figura 42), ya que cada Feed puede generarse para cada uno de los sites de Imaginarium, los cuales contienen información útil como por ejemplo el idioma en el que se debe generar el Feed. Además en la relación FeedSite generada se ha decidido guardar el nombre del archivo, el campo activo que actuará de flag para saber si se tiene que generar el feed para ese site en un determinado momento y un campo para la frecuencia, que almacenará una expresión cron para saber cuando tiene que generarse el feed.

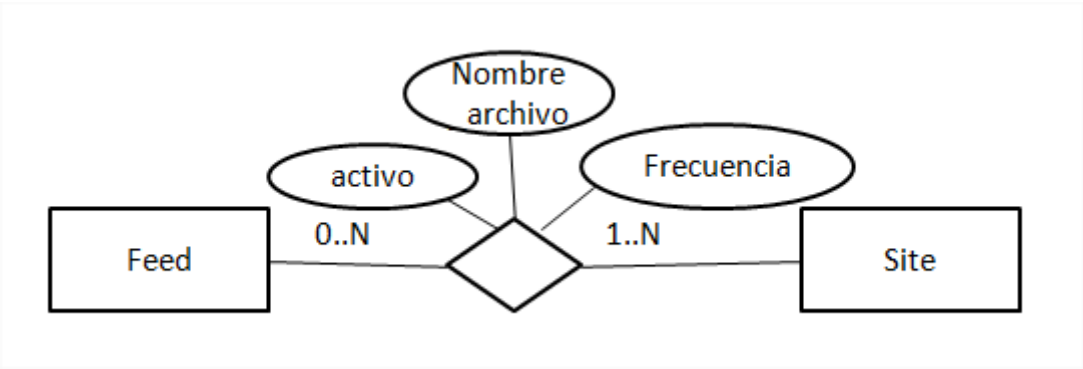


Figura 42. Relación feed-site

El tipo de entidad Mapeo_Atr (Figura 43) sirve para almacenar un mapeo sencillo de dónde se encuentran los atributos más comunes de los productos para la generación de Feed. Podría haberse omitido este mapeo y haberse relacionado directamente con todos los posibles atributos de productos de Imaginarium, pero se ha decidido hacer así por dos motivos. El primero es sencillez. Es más intuitivo tener que relacionarse solamente con una tabla que con muchas otras ya que los atributos están dispersos por las distintas tablas de la base de datos de Imaginarium. La segunda razón es por hacerlo genérico. De cara a un futuro, si se va a trasladar este módulo a otros proyectos que usan Hermes cambiaría radicalmente el esquema sin esta tabla.

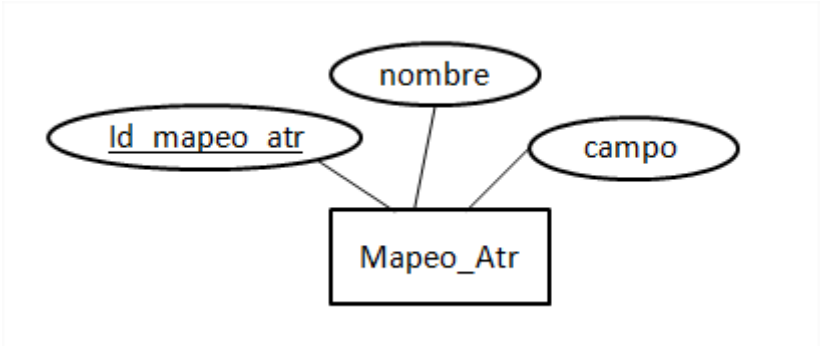


Figura 43. Tipo de entidad Mapeo_atr

La relación ternaria Atr_Canal-Feed-Mapeo_Atr (Figura 44) nace de la necesidad de almacenar para cada Feed los atributos de productos que se van

a generar en el fichero resultante. Para cada Feed se necesita almacenar qué atributo perteneciente a su canal asociado va emparejado con qué valor. Dicho valor se obtiene de la tabla Mapeo_Atr ya descrita anteriormente.

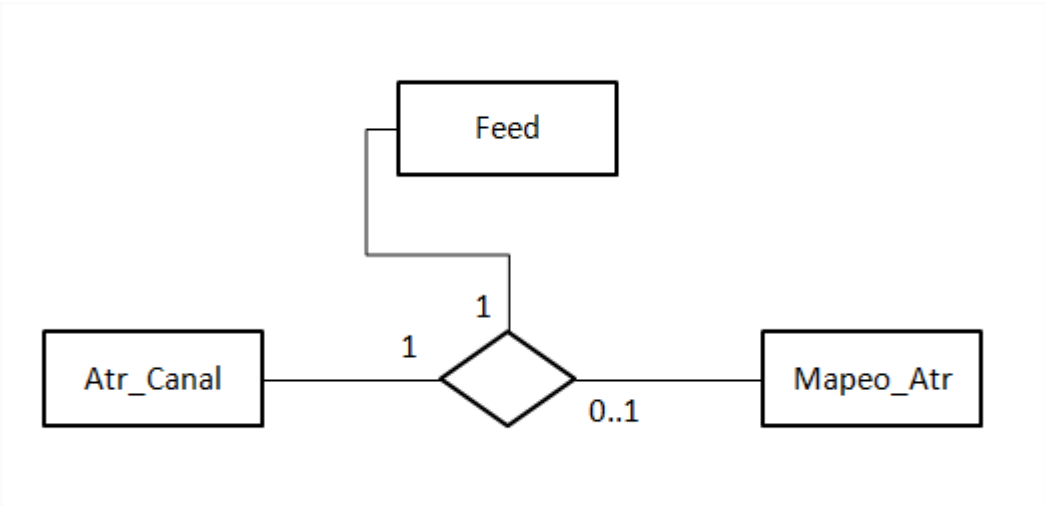


Figura 44. Relación ternaria Atr_Canal-Feed-Mapeo_Atr

Las relaciones de Feed con Categoria_pmm y Categoria_ecommerce se forman de la necesidad de tener la posibilidad de generar los Feeds con productos pertenecientes a categorías pmm o categorías ecommerce (Figura 45), que son dos tipos de categorías internas de Imaginarium. Las cardinalidades son N:N ya que para cada Feed pueden escogerse varías categorías pmm o varías categorías ecommerce y las categorías ecommerce o pmm pueden aparecer en distintos Feeds.

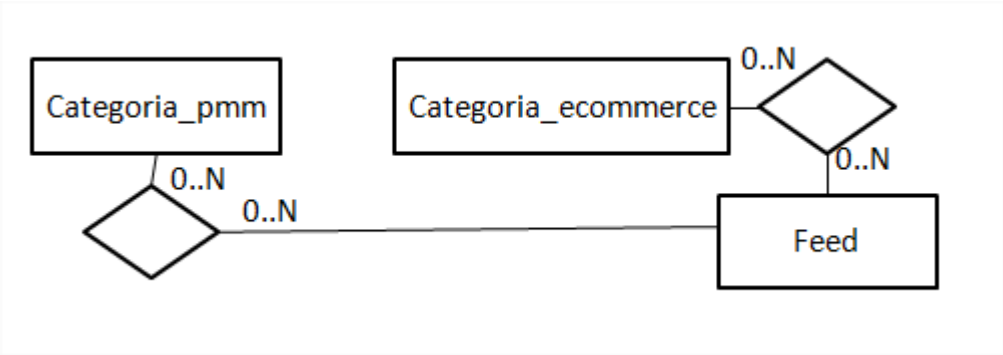


Figura 45. Tipo Relaciones Feed con Categoria_pmm y Categoria_ecommerce

Este es el tipo de entidad Feed_Backup (Figura 46). Aquí es dónde se almacenará la información relevante cada vez que se genere un feed feed. Los Feed_Backup se identificarán por un número natural que se autogenerará cada vez que se genere un feed. Además, para cada Feed_Backup se guardarán los distintos atributos ya mencionados en apartados anteriores.

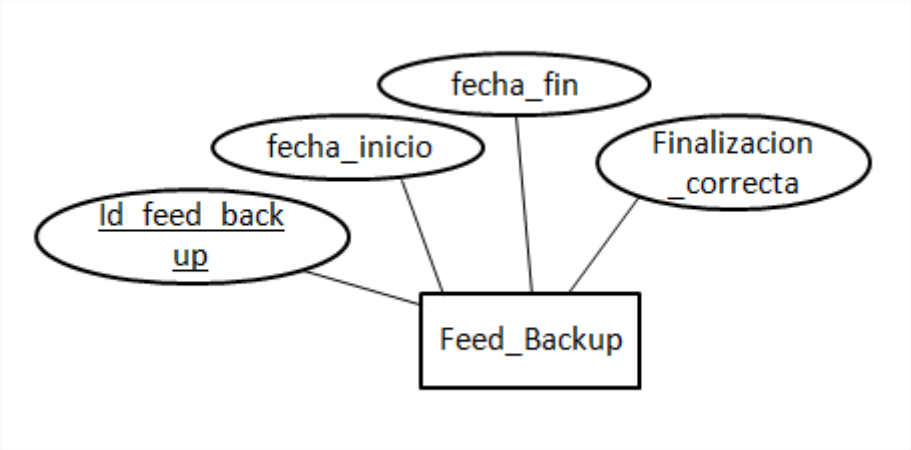


Figura 46. Tipo de entidad Feed_Backup

Una vez definido el modelo entidad relación se pasará al modelo relacinal en el siguiente apartado.

3. Modelo relacional

En este apartado se describirá el modelo relacional obtenido a partir del modelo entidad relación descrito en el apartado anterior. En la figura 47 puede observarse la relación Canal.

```
Canal (
    id_canal : integer;
    nombre   : string;
    nombre_tag : string;
    precio_tag : string;
    producto_tag : string;
    cabecera_tag : string;
)
```

Figura 47. Relación Canal

El paso del tipo de entidad relación Atr_Canal aparte de los atributos que poseía, ha generado dos atributos nuevos: atributo_padre y atributo_contenido_en (Figura 48). Estos atributos se han generado ya que Atr_Canal tenía dos relacionaes consigo mismo de cardinalidad 1:N. Además, estos dos atributos serán claves ajenas de la misma relación Atr_Canal.

```
Atr_Canal (
    id_atr_canal : integer;
    nombre : string;
    tipo : string;
    obligatorio : boolean;
    orden : integer;
    atributo_padre : integer;
    atributo_contenido_en : integer;
    clave ajena atributo_padre
    referencia id_atr_canal (Atr_Canal);
    clave ajena atributo_contenido_en
    referencia id_atr_canal (Atr_Canal)
)
```

Figura 48. Relación Atr_Canal

En la relación Feed se ha generado un nuevo atributo id_canal, que referencia a la tabla Canal, ya que la cardinalidad entre estos dos tipos de entidad es de 1:N (figura 49).

```
Feed (
    id_feed : integer;
    nombre : string;
    formato : string;
    ruta : string;
    id_consulta : integer;
    id_canal : integer;
    clave ajena id_canal
    referencia id_canal (Canal)
)
```

Figura 49. Relación Feed

Debido a la cardinalidad de la relación entre los tipos de entidad Feed y Site N:N se ha generado una nueva relación que he llamado Feed_Site (Figura 50). Esta relación almacenará para cada Feed, los distintos Sites para los que se generará el fichero. Su clave primaria está compuesta por el identificar que representa a cada Feed (id_feed) y el que representa cada Site (id_site). Además estos dos identificadores son claves ajenas de Feed y Site respectivamente.

```
Feed_Site (
    id_feed : integer;
    id_site : integer;
    activo : boolean;
    nombre_archivo : string;
    frecuencia : string;
    id_feed clave ajena
    referencia id_feed (Feed);
    id_site clave ajena
    referencia id_site (Site);
)
```

Figura 50. Relación Feed_Site

El paso a la relación Mapeo_Atr es inmnediato (Figura 51).

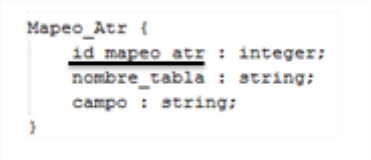


Figura 51. Relación Mapeo_Atr

De la relación ternaria entre los tipos de entidad Feed, Mapeo_Atr y Atr_Canal nace una nueva relación que he llamado Feed_AtrCanal_Mapeo (Figura 52). Dicha relación tiene como clave primaria compuesta los identificadores de las tres relaciones que la componen.

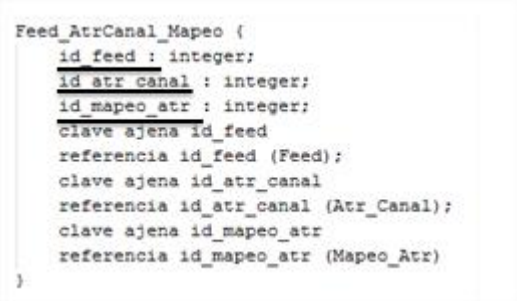


Figura 52. Relación Feed_AtrCanal_Mapeo

Se han generado dos relaciones nuevas de las relaciones de Feed con las tablas de Imaginarium Categoria_ecommerce y Categoria_pmm que he llamado Feed_Categoria_Ecommerce y Feed_Categoria_Pmm respectivamente (Figuras 53 y 54). Estas nuevas relaciones tienen como clave compuesta el identificador de Feed y el identificador de las categorías.

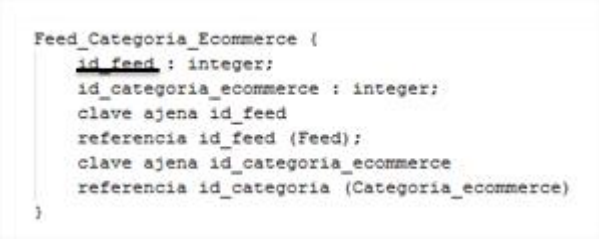


Figura 53. Relación Feed_Categoria_Ecommerce

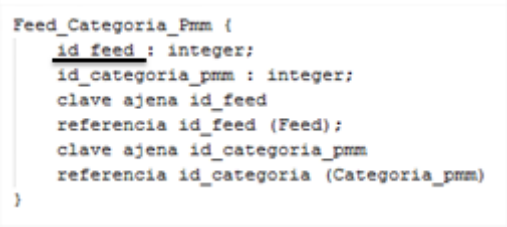


Figura 54. Relación Feed_Categoria_Pmm

Por último, la relación Feed_Backup tendrá como claves ajenas los identificadores de Feed y de Site para conocer de que feed es ese Backup (Figura 55).

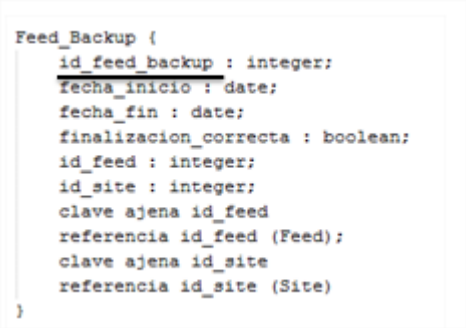


Figura 55. Relación Feed_Backup

4. Tablas SQL

Como se ha comentado previamente el sistema gestor de base de datos empelado ha sido SQL Server ya que la base de datos que se utiliza para albergar la administración de Hermes para Imaginarium era esta misma. EL paso del modelo relacional a las tablas SQL es inmediato:

```
CREATE TABLE CANAL (
	ID_CANAL numeric(18, 0) IDENTITY(1,1),
	NOMBRE nvarchar(250),
	NOMBRE_TAG nvarchar(250),
	PRECIO_TAG nvarchar(250),
	PRODUCTO_TAG nvarchar(250),
	CABECERA_TAG nvarchar(250)
)

ALTER TABLE CANAL ADD CONSTRAINT PK_ID_CANAL PRIMARY KEY CLUSTERED
(ID_CANAL ASC)

CREATE TABLE ATRIBUTO_CANAL (
	ID_ATRIBUTO_CANAL numeric(18, 0) IDENTITY(1,1),
	NOMBRE nvarchar(250),
	TIPO nvarchar(250),
	OBLIGATORIO bit,
	ORDEN numeric(4,0),
	ATRIBUTO_PADRE numeric(18,0),
	ID_CANAL NUMERIC(18,0),
	ATRIBUTO_CONTENIDO_EN numeric(18,0)
)

ALTER TABLE ATRIBUTO_CANAL ADD CONSTRAINT PK_ID_ATRIBUTO_CANAL PRIMARY
KEY CLUSTERED (ID_ATRIBUTO_CANAL ASC)

ALTER TABLE [dbo].[ATRIBUTO_CANAL]  WITH NOCHECK ADD  CONSTRAINT
[FK_ATRIBUTO_PADRE] FOREIGN KEY([ATRIBUTO_PADRE])
REFERENCES [dbo].[ATRIBUTO_CANAL] ([ID_ATRIBUTO_CANAL])
```

```
ALTER TABLE [dbo].[ATRIBUTO_CANAL]  WITH NOCHECK ADD  CONSTRAINT
[FK_ATRIBUTO_CONTENIDO_EN] FOREIGN KEY([ATRIBUTO_CONTENIDO_EN])
REFERENCES [dbo].[ATRIBUTO_CANAL] ([ID_ATRIBUTO_CANAL])

ALTER TABLE [dbo].[ATRIBUTO_CANAL]  WITH NOCHECK ADD  CONSTRAINT
[FK_ID_CANAL] FOREIGN KEY([ID_CANAL])
REFERENCES [dbo].[CANAL] ([ID_CANAL])


CREATE TABLE FEED (
       ID_FEED numeric(18, 0) IDENTITY(1,1),
       NOMBRE nvarchar(250),
       FORMATO nvarchar(250),
       RUTA nvarchar(250),
       ID_CONSULTA nvarchar(250),
       ID_CANAL numeric(18, 0)
)

ALTER TABLE FEED ADD CONSTRAINT PK_ID_FEED PRIMARY KEY CLUSTERED
(ID_FEED ASC)

ALTER TABLE [dbo].[FEED]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_ID_CANAL] FOREIGN KEY([ID_CANAL])
REFERENCES [dbo].[CANAL] ([ID_CANAL])


CREATE TABLE FEED_SITE (
       ID_FEED numeric(18, 0),
       ID_SITE NUMERIC,
       ACTIVO bit,
       NOMBRE_ARCHIVO nvarchar(250),
       FRECUENCIA nvarchar(250)
)

ALTER TABLE FEED_SITE ADD CONSTRAINT PK_ID_FEED_SITE PRIMARY KEY
CLUSTERED (ID_FEED,ID_SITE)

ALTER TABLE [dbo].[FEED_SITE]  WITH NOCHECK ADD  CONSTRAINT
[FK_ID_FEED] FOREIGN KEY([ID_FEED])
REFERENCES [dbo].[FEED] ([ID_FEED])

ALTER TABLE [dbo].[FEED_SITE]  WITH NOCHECK ADD  CONSTRAINT
[FK_ID_SITE] FOREIGN KEY([ID_SITE])
REFERENCES [dbo].[SITE] ([ID_SITE])



CREATE TABLE MAPEO_ATR (
       ID_MAPEO_ATR numeric(18, 0) IDENTITY(1,1),
       NOMBRE_TABLA nvarchar(250),
       CAMPO nvarchar(250)
)

ALTER TABLE MAPEO_ATR ADD CONSTRAINT PK_MAPEO_ATR PRIMARY KEY
CLUSTERED (ID_MAPEO_ATR ASC)
```

```
CREATE TABLE FEED_ATRCANAL_MAPEO (
       "ID_FEED" NUMERIC,
       "ID_ATRIBUTO_CANAL" NUMERIC,
       "ID_MAPEO_ATR" NUMERIC,
       UTM nvarchar(250)
)

ALTER TABLE FEED_ATRCANAL_MAPEO ADD CONSTRAINT PK_FEED_ATRCANAL_MAPEO
PRIMARY KEY CLUSTERED (ID_FEED,ID_ATRIBUTO_CANAL,ID_MAPEO_ATR )

ALTER TABLE [dbo].[FEED_ATRCANAL_MAPEO]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_ATRCANAL_MAPEO_ID_FEED] FOREIGN KEY([ID_FEED])
REFERENCES [dbo].[FEED] ([ID_FEED])

ALTER TABLE [dbo].[FEED_ATRCANAL_MAPEO]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_ATRCANAL_MAPEO_ID_ATRCANAL] FOREIGN KEY([ID_ATRIBUTO_CANAL])
REFERENCES [dbo].[ATRIBUTO_CANAL] ([ID_ATRIBUTO_CANAL])

ALTER TABLE [dbo].[FEED_ATRCANAL_MAPEO]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_ATRCANAL_MAPEO_ID_MAPEO_ATR] FOREIGN KEY([ID_MAPEO_ATR])
REFERENCES [dbo].[MAPEO_ATR] ([ID_MAPEO_ATR])

CREATE TABLE FEED_BACKUP (
       ID_FEED_BACKUP numeric(18, 0) IDENTITY(1,1),
       FECHA_INICIO nvarchar(250),
       FECHA_FIN nvarchar(250),
       FINALIZACION_CORRECTA bit
       ID_FEED numeric(18, 0),
       ID_SITE NUMERIC not null
)

ALTER TABLE FEED_BACKUP ADD CONSTRAINT PK_FEED_BACKUP PRIMARY KEY
CLUSTERED (ID_FEED_BACKUP ASC)

ALTER TABLE [dbo].[FEED_BACKUP]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_BACKUP_ID_FEED] FOREIGN KEY([ID_FEED])
REFERENCES [dbo].[FEED] ([ID_FEED])

ALTER TABLE [dbo].[FEED_BACKUP]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_BACKUP_ID_SITE] FOREIGN KEY([ID_SITE])
REFERENCES [dbo].[SITE] ([ID_SITE])

CREATE TABLE FEED_CATEGORIA_PMM (
       ID_FEED numeric(18, 0),
       ID_CATEGORIA_PMM NUMERIC
)

ALTER TABLE FEED_CATEGORIA_PMM ADD CONSTRAINT PK_FEED_CATEOGRIA_PMM
PRIMARY KEY CLUSTERED (ID_FEED,ID_CATEGORIA_PMM)

ALTER TABLE [dbo].[FEED_CATEGORIA_PMM]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_CATEOGRIA_PMM_ID_FEED] FOREIGN KEY([ID_FEED])
REFERENCES [dbo].[FEED] ([ID_FEED])

ALTER TABLE [dbo].[FEED_CATEGORIA_PMM]  WITH NOCHECK ADD  CONSTRAINT
[FK_FEED_CATEOGRIA_PMM_ID_CATEGORIA_PMM] FOREIGN
KEY([ID_CATEGORIA_PMM])
REFERENCES [dbo].[CATEGORIA_PMM] ([ID_CATEGORIA])
```

```
CREATE TABLE FEED_CATEGORIA_ECOMMERCE (
       ID_FEED numeric(18, 0),
       ID_CATEGORIA_ECOMMERCE NUMERIC
)

ALTER TABLE FEED_CATEGORIA_ECOMMERCE ADD CONSTRAINT
PK_FEED_CATEGORIA_ECOMMERCE PRIMARY KEY CLUSTERED
(ID_FEED,ID_CATEGORIA_ECOMMERCE)

ALTER TABLE [dbo].[FEED_CATEGORIA_ECOMMERCE]  WITH NOCHECK ADD
CONSTRAINT [FK_FEED_CATEGORIA_ECOMMERCE_ID_FEED] FOREIGN
KEY([ID_FEED])
REFERENCES [dbo].[FEED] ([ID_FEED])

ALTER TABLE [dbo].[FEED_CATEGORIA_ECOMMERCE]  WITH NOCHECK ADD
CONSTRAINT [FK_FEED_CATEGORIA_ECOMMERCE_ID_CATEGORIA_ECOMMERCE]
FOREIGN KEY([ID_CATEGORIA_ECOMMERCE])
REFERENCES [dbo].[CATEGORIA_ECOMMERCE] ([ID_CATEGORIA])
```

5. Diseño físico

Lo primero que he pensado a la hora de hacer el diseño físico de la base de datos ha sido, ¿cuáles son las operaciones más comunes? A la hora de hacer la configuración entera para generar un feed se realizan varios INSERT en todas las tablas empleadas pero el gran peso de las sentencias a la base de datos van a ser consultas de tipo SELECT, ya que para cada feed configurado se va a ejecutar periódicamente, mientras que como se ha comentado previamente configurar un feed solamente son unos pocos INSERT.

Una vez llegada a esta conclusión hay que observar que tablas van a ser las que más veces se van a consultar cada vez que se haga una generación de un fichero para estudiar si merece la pena crear algún índice.

La generación de un Feed comienza con una consulta a la tabla Feed para recoger sus atributos a través de su clave primaria. A continuación, realizar una consulta para sacar los productos que deben ir en el Feed. Estos productos están en tablas ajenas a las diseñadas para este sistema.

A continuación se realiza una consulta a la tabla Feed_Site para ver para que sites tiene que generar el fichero. Además se consulta también la tabla canal por su clave primaria obtenida cuando se consultó la tabla Feed. Se realiza otra consulta para obtener todos los atributos de un determinado canal a través del identificador del canal. Para cada atributo canal se consulta la tabla Feed_AtrCanal_Mapeo por el id del feed y el id del atributo del canal.

Finalmente se itera por todos los productos y se va consultando en tablas de existencia previa los valores de cada atributo. Estas tablas ya se estudiaron en su idea y estaban optimizadas en cuanto a su diseño físico.

De modo que las consultas más comunes según el análisis previo son a la tabla de atributos del canal. Además la consulta se realiza a través del identificador del canal, que no es clave primaria de la tabla, de modo que se ha creado un índice en la tabla AtributoCanal del atributo idCanal:

```
CREATE INDEX IDX_IDCANAL ON ARIBUTO_CANAL (IDCANAL)
```

El índice resulta de ayuda ya que aligera las consultas de tipo SELECT, que son las más comunes como se ha comentado. También retrasa los INSERTS o UPDATES ya que cada uno de ellos tiene que reorganizar los índices, pero el número de SELECTS es mucho mayor que de INSERTS y UPDATES.

ANEXO D. TECNOLOGÍAS UTILIZADAS

1. Introducción

Este anexo está dedicado a explicar las tecnologías y herramientas utilizadas para implementar el módulo Product Feed. La decisión de porqué se han elegido las tecnologías fundamentales en las que está desarrollada la aplicación viene marcada por la aplicación web Hermes, ya que el módulo desarrollado está incorporado en Hermes me he tenido que amoldar a lo que utiliza Hermes. Esto puede tener como ventaja el no tener que barajar otras posibilidades, pero como inconveniente el tener que aprender específicamente casi todas las tecnologías que utiliza y no poder analizar qué ventajas o inconvenientes pudiese tener utilizar otras herramientas.

Cada apartado del anexo explica una tecnología y lo que aporta al módulo.

2. Lenguajes de programación

A continuación se van a enumerar, describir y explicar para que se han usado los lenguajes de programación utilizados para realizar el módulo Product Feed.

2.1. Java

El lenguaje base para la aplicación web utilizado es Java. Esto ha sido una ventaja, ya que Java es un lenguaje intuitivo que ya conocía por lo que me ayudó a entender mejor la aplicación de Hermes y agilizar el proceso de implementación dónde se requería su uso.

Java es utilizado tanto en la parte de cliente como en la parte de servidor que componen una aplicación web. En la parte del servidor es usado para llevar toda la lógica de la aplicación. Mientras que en la parte del cliente también se ha usado java incrustado en los archivos html para mostrar al usuario la interfaz con la que interactuar. Los archivos html con java incrustado se denominan JSP.

2.2. HTML

HTML (HyperText Markup Language) ha sido el lenguaje de marcado utilizado para generar el contenido visual en la parte del cliente. HTML es un lenguaje muy común para este fin combinado con lenguajes de estilos para gestionar el aspecto visual de los elementos. El lenguaje usado para la gestión de los estilos de la página ha sido CSS

.

2.3. XML

XML (eXtensible Markup Language) es también otro lenguaje de marcado. Los ficheros xml han sido usados para portar la información de configuración de algunos módulos de la aplicación. Por ejemplo se ha usado para la configuración del FrameWork Struts del que se hablará más adelante.

2.4. JavaScript

JavaScript es un lenguaje potente que se utiliza dentro de los ficheros HTML. Su utilidad es gestionar el contenido de las páginas dinámicamente. Por ejemplo, si se pulsa un botón mostrar un texto. Con JavaScript se puede manipular totalmente cualquier elemento de una página web como se requiera. Además se ha utilizado alguna librería de JavaScript como JQuery ya que es algo más intuitiva y ofrece en ciertos casos más posibilidades.

También se ha utilizado AJAX (Ashynchronous JavaScript And Xml), que es una técnica de desarrollo web. Se ha utilizados para hacer peticiones al servidor y cargar contenido de una página sin necesidad de recargar la página entera, solamente la parte afectada.

2.5. SQL

SQL (Structured Query Language) ha sido el lenguaje utilizado para hacer consultas a la base de datos. El sistema gestor de base de datos que se ha utilizado ha sido SQL Server. Aunque se utilice SQL en varios sistemas gestor de base de datos la sintaxis de uno a otro varía a veces ligeramente.

Se ha utilizado un cliente con interfaz gráfica para realizar las consultas a la base de datos llamado HeidiSQL.

3. Frameworks

Un Framework o marco de trabajo es una herramienta que se utiliza para facilitar el desarrollo de aplicaciones o programas informáticos. Los siguientes Framworks utilizados ya se usaban para Hermes, de modo que he estudiado como funcionan y he aprendido a usarlos.

3.1. Hibernate

Hibernate es una implementación de JPA (Java Persistance API). Hibernate es usado para mapear objetos Java a elementos de la base de datos y facilita las consultas de los mismos. Por ejemplo, si se necesita consultar un feed: Se tiene un objeto Java Feed.java y una tabla en la base de datos FEED, Hibernate realiza un mapeo entre los dos elementos, y a la hora de consultar un feed te carga directamente los valores de la tabla en el objeto Feed.java.

3.2. Struts

Struts es un marco de trabajo que permite controlar el modelo MVC (Modelo Vista Controlador) de una aplicación web fácilmente bajo la plataforma Java EE. La funcionalidad de Struts es reducir el tiempo de desarrollo proporcionándote mecanismos que abstraen de los aspectos más complicados de la programación de una aplicación web.

3.3. Spring

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control para la plataforma Java. Spring ha sido usado para la inyección de dependencias a las distintas clases Java utilizadas. Spring tiene un marco dentro de la aplicación web dónde residen todas las clases Java que se quieran inicializadas en el lanzamiento de la aplicación para poder disponer de ellas sin necesidad de crear una instancia nueva de dicha clase cada vez que se quiera utilizar un método suyo.

4. Contenedores Web

Los contenedores web son los entornos de ejecución de los componentes web que hacen funcionar las aplicaciones web. En Hermes se utiliza Apache y Apache Tomcat.

4.1. Apache

Apache es el contenedor Web principal. Contra él van las peticiones de los cliente y es el encargado de decidir qué hacer. También es el encargado de servir los contenidos estáticos, que generalmente son de poco peso pero son muchos, para liberar de carga al Tomcat. Si se hace una petición que requiera del servidor la redirige al Tomcat.

4.2. Apache Tomcat

Es realmente dónde reside el servidor de la aplicación web. Es el entorno dónde se realizan las operaciones necesarias para responder las peticiones del cliente.

5. Entorno de desarrollo

El entorno de desarrollo utilizado ha sido Eclipse. Eclipse es de gran ayuda ya que compila automáticamente según vas desarrollando el código y muestra los errores de compilación y advertencias. También facilita una visión total de todos los paquetes del proyecto así como otros componentes como librerías externas o ficheros de configuración.

También se ha usado el pluggin Ant, que sirve para compilar la aplicación entera y permite depositar el archivo war generado dónde se quiera, que generalmente es en la carpeta webapps del tomcat para que despliegue la aplicación automáticamente.

ANEXO E. GESTIÓN DEL PROYECTO

1. Introducción

Este anexo está dedicado a describir todo lo relacionado con la gestión del proyecto. La gestión del proyecto ha ido ligada totalmente a la participación del cliente Imaginarium, ya que el proyecto nació por una propuesta suya y la solución debía ser conforme a sus necesidades. En este anexo se va a hablar sobre la planificación del proyecto, el control de versiones y las reuniones con el cliente.

2. Planificación del proyecto

En este apartado se va a describir la planificación inicial realizada como primer paso en la gestión del modulo de Product Feed, desglosando el tiempo total en distintos hitos. El proyecto está planificado para una sola persona durante aproximadamente 500 horas de trabajo.

La planificación se muestra en la figura 56:

Etapas del proyecto	Horas	Dic 15	Ene 15	Feb 15	Mar 15	Abr 15
Análisis	**70**					
Estudio previo	10					
Análisis de requisitos	15					
Análisis del sistema	35					
Análsis de la BD	10					
Diseño	**75**					
Diseño del sistema	40					
Diseño de la BD	20					
Prototipado de ventanas	15					
Implementación	**300**					
Implementación BD	30					
Maquetación UI	70					
Implementación módulo	200					
Pruebas	**60**					
Pruebas unitarias	40					
Pruebas aplicación conjunta	10					
Pruebas entorno de validación	10					
Despliegue	**20**					
Despliegue entorno pro	10					
Pruebas entorno pro	10					
Documentación	**40**					

Figuras 56 Planificación del proyecto

Considerando que cada jornada es de 8 horas se planificó el inicio del proyecto para el 10 de diciembre de 2014 y finalización aproximada del 27 de Marzo de 2015.

Aún sabiendo que planificar un proyecto de gran calibre es difícil, las horas reales invertidas en el proyecto se han aproximado mucho al plan inicial. Se comenzó el día estipulado pero se prolongó hasta el día 15 de abril de 2015 con un tiempo invertido total de alrededor de 600 horas. El plazo de entrega se vio afectado por incidencias urgentes que se me encomendaban en la empresa ajenas al proyecto y debían resolverse.

3. Control de versiones

En este apartado se va a describir como se ha llevado el control de versiones a lo largo del proyecto.

El desarrollo se ha realizado en un ordenador particular, pero el versionado se ha realizado almacenando los archivos del proyecto en un repositorio SVN. Esto se decidió siguiendo la línea de cómo se trabajaba en Hermes. El programa que se utilizaba era Tortoise SVN, un cliente de Apache Subversion para Windows.

A mitad del proyecto se cambió de responsable técnico del área en la empresa y se decidió cambiar de software de control de versiones a Git porque se pensó que se amoldaba más a las necesidades del equipo en general. De modo que se migró toda la información del repositorio SVN al repositorio Git incluidos los archivos de mi proyecto. El programa de escritorio para acceder al repositorio fue SourceTree. El cambio no supuso ningún obstáculo ya que Git gestiona mejor el concepto de “rama de desarrollo” de controlado de versiones, y ya que yo era el único trabajando en mi proyecto pude desarrollar el código de forma más independiente al resto del equipo.

4. Reuniones con el cliente

Las reuniones con el cliente han sido de gran utilidad, tanto para Imaginarium como para mí. No se tenía planeado con mucho tiempo de antelación cuando tendrían lugar sino que se celebraban si había alguna necesidad. Las reuniones solían ser de una hora de duración y tenían lugar en las oficinas de Imaginarium.

La primera reunión que se tuvo fue al principio del proyecto. En esta, expuse a Imaginarium los requisitos que había obtenido tras su propuesta. Les agradaron las ideas obtenidas y propusieron algunos pequeños cambios de cara a la funcionalidad final.

En las reuniones posteriores servían para mostrar el progreso del proyecto y para resolver dudas que iban surgiendo a lo largo del mismo. También servían como retroalimentación por parte de Imaginarium para ir puliendo los resultados que iban obteniéndose.

Finalmente, tuvo acontecimiento una última reunión cuando la implementación estaba terminada en el que se presentó el resultado final con una pequeña demo de todas sus funcionalidades en la que el cliente vio su producto final terminado y quedó satisfecho.

ANEXO F. MANUAL DE USUARIO

1. Introducción

Este anexo está dedicado a la descripción de la aplicación desde el punto de vista del usuario. Se explicará por secciones qué permite realizar la aplicación y como navegar a través de ella.

Se ha descompuesto el anexo en tres secciones, una por cada parte del módulo Producto Feed.

En la primera sección se describirá como gestionar los canales: crearlos, modificarlos, borrarlos y clonarlos, además de generar para cada canal sus propios atributos.

En la segunda sección se describirá como gestionar la generación de feeds. Se describirá como elegir el canal que se desee para ese feed, los sites para los que se generará, el formato del archivo resultante, así como el contenido que tendrá el feed y los productos que aparecerán en este. Además se describirá como establecer la frecuencia de generación de los ficheros y el nombrado de los mismos.

En la última sección se describirá como visualizar todos los feeds que se tienen configurados hasta el momento, como saber cuando será la próxima vez que se vayan a generar y como activarlos y desactivarlos.

2. Gestión de canales

Para acceder a la gestión de canales debemos hacer click en la opción “Canal” dentro de la sección “Product Feed” del menú de Hermes (Figura 57):

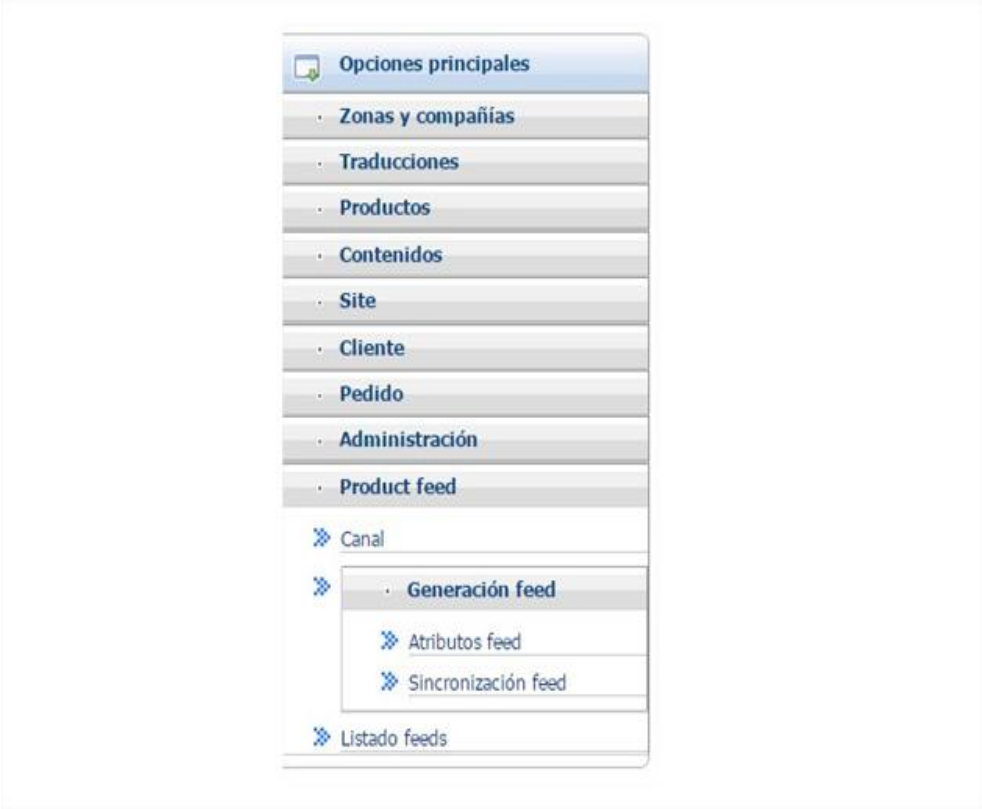


Figura 57. Menú Product Feed en Hermes

Una vez accedido a la sección de canales podremos visualizar la siguiente pantalla (Figura 58)

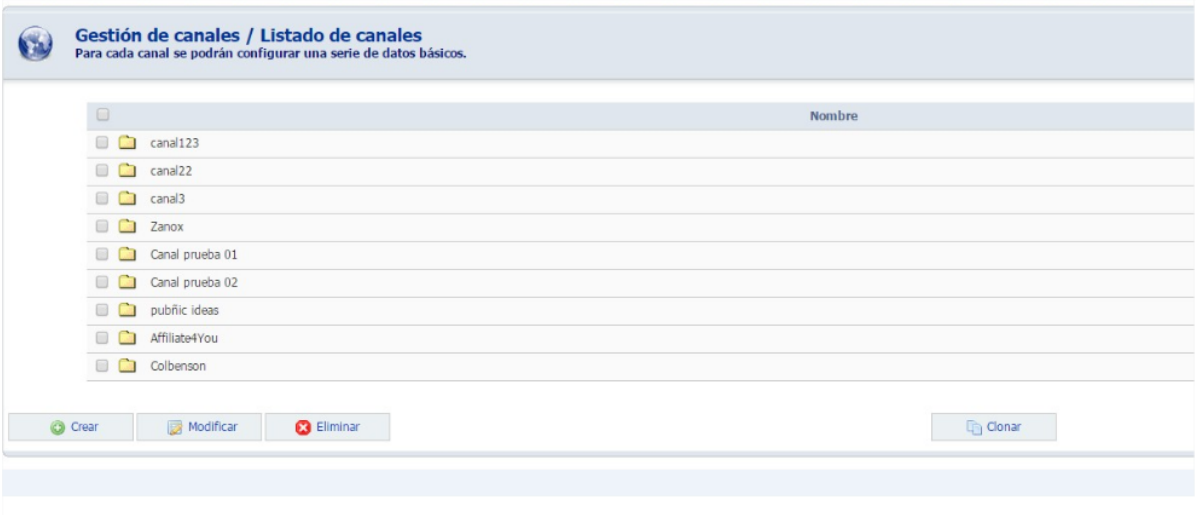


Figura 58. Pantalla gestión de canales

Desde esta pantalla podemos visualizar los canales que existen actualmente en el sistema.

Para borrar uno o varios canales debemos seleccionarlos y hacer click en el botón “Eliminar” seguidamente. Aparecerá un cuadro de diálogo preguntándonos si estamos seguros de que se quieren eliminar los canales seleccionados. Si aceptamos se borrarán los canales seleccionados.

Para clonar un canal debemos seleccionar un único canal y hacer click en el botón “Clonar”. Se clonará el canal seleccionado con el nombre “copia de “ más el nombre del canal clonado. Se habrá clonado tanto el canal como todos sus atributos y propiedades que tuviese configurados.

Para crear un nuevo canal debemos hacer click en el botón “Crear”. Esto nos llevará a la siguiente pantalla (Figura 59):

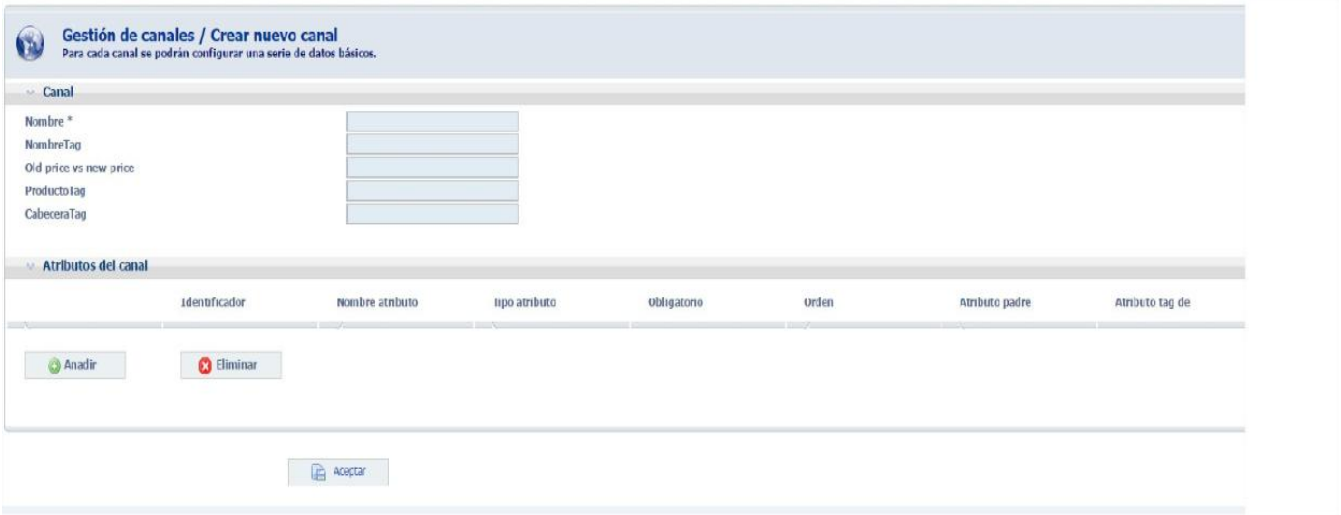


Figura 59. Pantalla creación de canales

Para crear un nuevo canal debemos rellenar los campos de la parte superior. Únicamente el nombre del canal es imprescindible y obligatorio.

El campo nombreTag es la cadena de texto que llevarán todos los tags de atributos en el fichero generado. Por defecto es vacía por lo que si no se rellena los tags llevarán exclusivamente el nombre que se le dé a cada atributo.

El campo Old price vs new price es algo muy concreto. Se debe rellenar en el caso de que se necesite para el feed el precio y el precio rebajado de los productos y en el caso de que se quiera cambiar el nombre del precio si no

existe precio rebajado para cada producto. Si esto ocurre el precio tendrá el nombre que indiquemos en este campo en vez de el nombre que le dimos originalmente.

·El campo productoTag es la cadena de texto que tendrá el tag de apertura de cada producto en el fichero.

El campo CabeceraTag es la cadena de texto que tendrá el tag de apertura del fichero.

Una vez rellenados estos campos, se debe hacer click en el botón “Aceptar” para crear el canal nuevo. Se llevará de nuevo al listado de canales (Figura 58).

Para modificar un canal se debe seleccionar un único canal y hacer click en el botón “modificar” o también se puede hacer acceder haciendo click en el nombre de un canal directamente. Se nos llevará a la siguiente pantalla (Figura 60):

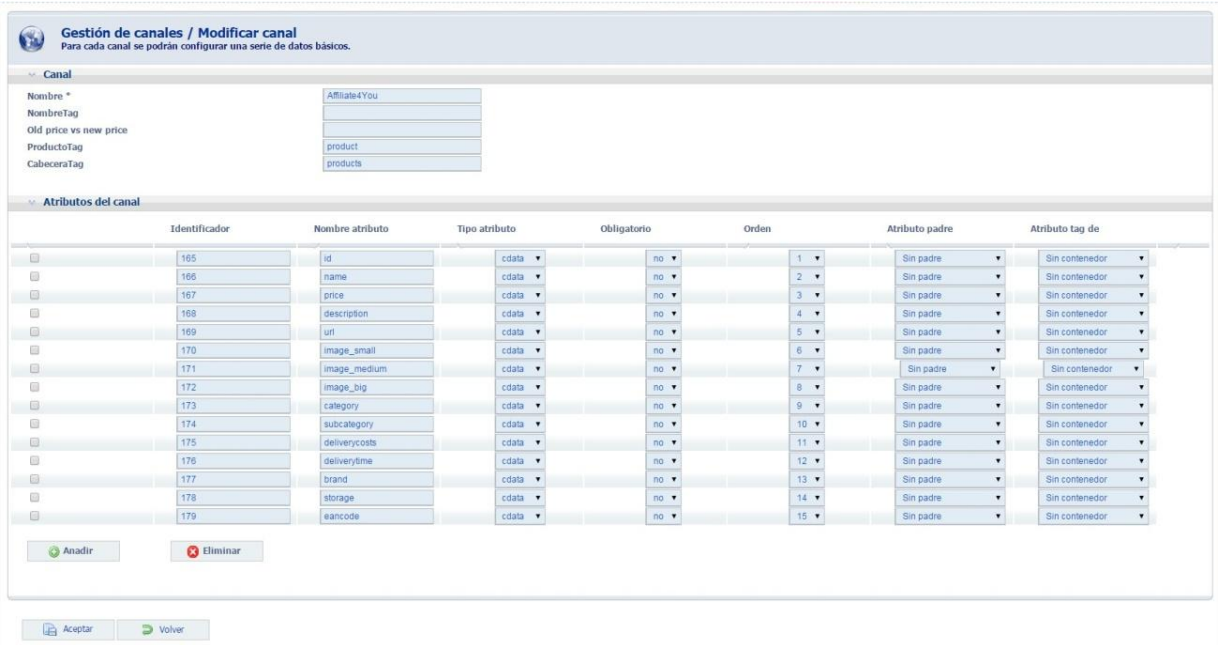


Figura 60. Pantalla modificación de canales

Con el botón “Añadir” se añadirá un nuevo atributo y con el botón “Eliminar” se eliminarán los atributos seleccionados.

Para cada atributo además, se deben elegir una serie de características como se muestra en la Figura 60.

- Nombre. Es el nombre que tendrá el atributo en el fichero generado. Por ejemplo, id-pmm, nombre, precio…
- Tipo de dato. Es común en ficheros xml introducir según qué tipo de datos entre la llave [CDATA]. Con esto se indica que se obtiene el valor del campo tal y como está entre las llaves. Esto se usa por el tema de los distintos tipos de codificación. Si no se quiere usar la llave [CDATA] se escogerá tipo de dato “Texto”.
- Obligatorio. En este campo se indica si el atributo deberá aparecer obligatoriamente en el fichero generado. Se explicará en la generación del feed la importancia de este campo.
- Orden. En este campo indicamos el orden en el que aparecerá el atributo en el fichero para cada producto. Cuando se selecciona para un atributo el campo atributo padre o atributo contenido en , el campo orden se desactiva ya que no tiene sentido que un atributo que esté contenido en otro tenga un orden de aparición.
- Atributo padre. Puede darse el caso de que algunos canales requieran atributos anidados.

```
<product>
    <id_pmm> 17728 </id_pmm>
    <precio>
        <valor> 9.95 </valor>
        <divisa> EUR </divisa>
    </precio>
</product>
```

Figura 61. Atributos anidados

Para configurar este tipo de estructuras se utiliza el Atributo padre. Se crearía un atributo llamado precio, y otros dos atributos: valor y divisa. Estos dos atributos tendría en el atributo padre el atributo precio.

- Atributo contenido en. Puede darse el caso de que algunos canales requieran atributos contenidos en otros atribuos.

<product>

<id_pmm> 17728 </id_pmm>

<precio divisa = “EUR””> 9.95</precio>

</product>

Figura 62. Atributos contenidos en

Para configurar este tipo de estructuras se utiliza el Atributo contenido en . Se crearía un atributo llamado precio y otro atributo llamado divisa. El atributo divisa tendría como atributo contenido en, el atributo precio.

Una vez configurado todo lo relevante al canal, ya tenemos preparado el esqueleto que tendrán todos los feeds generados para ese canal.

Ahora queda indicarle con qué datos se va a generar cada feed. Esto se configura en la gestión de generación de feed.

3. Gestión de generación del feed

Para acceder a la gestión de generación del feed se debe pulsar el botón “Atributos feed” del menú principal del módulo Product Feed (Figura 57).

Una vez se accede, se muestra la pantalla de atributos feed (Figura 63).

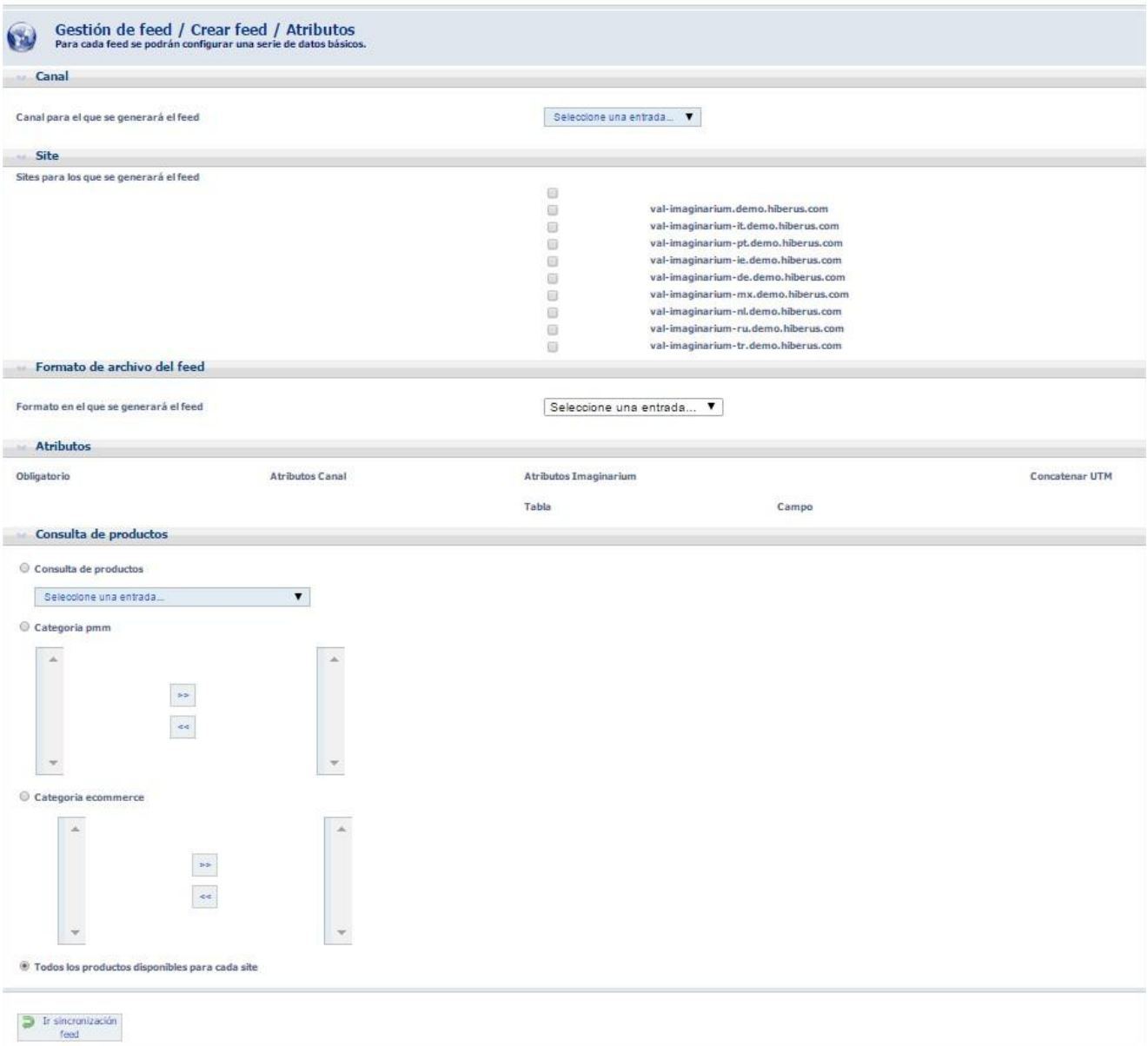


Figura 63. Atributos feed

Lo primero que se debe seleccionar es el canal para el que se generará el feed. Si se hace click en la pestaña con la etiqueta canal se desplegará una lista con todos los canales que se hayan configurado hasta el momento. (Figura 64)

Figura 64. Atributos feed - canal

A continuación, se muestra un listado de sites (Figura 65). Un site equivale a un sitio web.. Cada site tiene muchas diferencias con las demás, pero las más notorias y las que más afectan a la generación del site son dos:

- El idioma**.** Cada site lleva relacionado un idioma en el que se muestran los contenidos. Es por eso que el idioma de la información que obtendremos en el feed depende directamente del site que elijamos.
- Los productos**.** Cada site tiene asociado un determinado conjunto de productos. Además, para sites distintos puede que se venda el mismo producto pero con diferentes características, como por ejemplo el precio. Es por eso que los productos que obtengamos en el feed y sus atributos dependerán del site para el que se genere el feed.

Figura 65. Atributos feed – site

Es posible seleccionar más de un site, pero como mínimo se debe seleccionar uno. Se generará un fichero diferente para cada site seleccionado.

Lo tercero que se debe elegir es el formato del fichero o ficheros resultantes (Figura 66). Lo más común entre los canales para los que ya se generan feeds actualmente es formato xml, aunque en algunos casos se generan en formato csv. En el caso de que se precisara de un formato nuevo sería necesario desarrollar cómo generar el nuevo tipo de fichero.

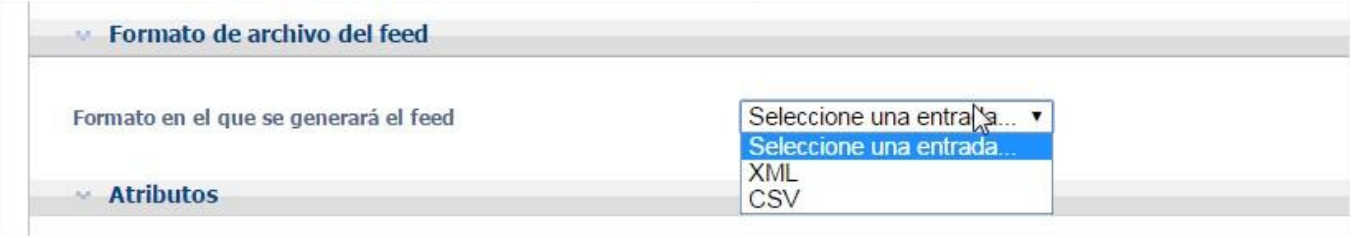


Figura 66. Atributos feed – formato fichero

Hasta ahora se ha descrito como construir el fichero generado, pero no se había configurado nada acerca de los datos reales que se mostrarán en el fichero. En esta cuarta parte de la pantalla (Figura 67) se elige para cada atributo que hemos definido en la estructura del canal, a que valor se va a corresponder de los existentes en la base de datos que utiliza el portal web.

Veámoslo con un ejemplo para clarificarlo. Para cada atributo debemos elegir con que dato de la base de datos se corresponderá. Por ejemplo, el atributo de canal “atr1” se corresponderá con el campo “precio_rebajado” de la tabla “precio_producto_zona_precio”. Esto quiere decir que para cada producto generará la etiqueta < atr1> (valor del campo “precio_rebajado” de la tabla “precio_producto_zona_precio”) </ atr1>.

La columna de obligatorio es importante. Si un atributo es obligatorio aparecerá en el fichero generado independientemente del valor que elijamos. Es decir, en este el caso que elijamos un campo de la base de datos que para algunos productos no tenga valor, en el caso de que sea obligatorio se mostrará la etiqueta sin valor y en el caso de que no sea obligatorio no se mostrará nada para esos productos que no se disponga información.

Figura 67. Atributos feed – mapeo atributos canal – atributos feed

También se dispone de la columna UTM. Esta columna es opcional y su utilidad es concatenar una cadena de texto a cada valor de atributo para un atributo determinado. Es común necesitar concatenar ciertos valores sobre todo en URLs. Por ejemplo, la url de un producto de Imaginarium puede ser www.imaginarium.es/58713 y para según qué afiliaciones se requiere que las urls contengan un atributo extra denominado utm, con lo cuál se pide que la etiqueta que contenga el valor de la url para todos los productos sea del estilo: www.imaginarium.es/58713?utm=public_ideas.... Para configurar esto se erigiría el campo “url_real” de la tabla “url_real_nueva” y en el apartado UTM se insertaría la cadena “?utm=public_ideas...”.

Lo último que podemos configurar en esta pantalla es lo más importante, los productos que aparecerán en el feed.

Se dispone de cuatro opciones (figura 68):

- Consulta de productos. Se pueden seleccionar un conjunto de atributos a través de una consulta de atributos personalizada. Esto es una utilidad que ya incorporaba Hermes con otras utilidades, de modo que se no va a indagar mucho en ello. Simplemente, si se elige esta opción aparecerán en el feed los productos del resultado de la consulta de productos seleccionada.
- Categorías ecommerce. Las categorías ecommerce son aquellas categorías que aparecen en la web de Imaginarium. Por lo que cada categoría ecommerce tiene una serie de atributos asociados a la misma.
- Categorias pmm. Las categorías pmm son las categorías que usa Imaginarium internamente, las que utilizan en su ERP. No son más que otra forma de agrupar productos. Se ha habilitado la posibilidad de elegir los dos tipos de categoría ya que según qué tipo de empleado de Imaginarium utilice la herramienta puede estar más familiarizado con las categorías pmm o con las categorías ecommerce.

- Todos los productos. Esta opción mostrará en el feed la información de todos los productos que estén disponibles para la venta en cada site.

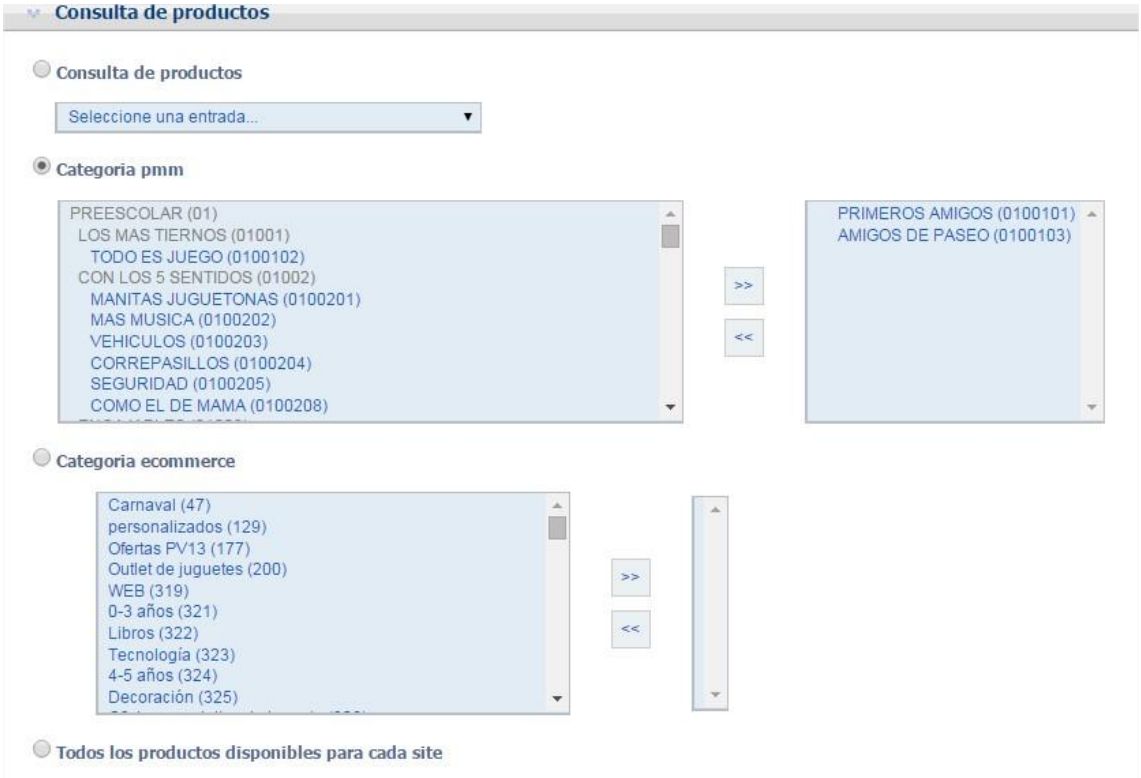


Figura 68. Atributos feed – consulta de productos

Una vez realizadas todas estas configuraciones con el botón “Ir a sincronización” en la parte inferior de la pantalla (Figura 8) nos llevará a la configuración de lo relacionado con la frecuencia de generación del feed (Figura 69).

. Figura 69. Sincronización feed

Lo primero que se pide en esta pantalla es el nombre del feed. Este nombre es algo intuitivo para el usuario, no tiene ningún impacto en el resultado del fichero o ficheros generados (Figura 70).

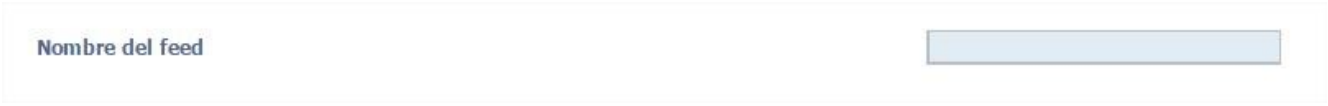


Figura 70. Sincronización feed – nombre feed

Lo segundo que se nos pide es el destino de los ficheros que se generarán (Figura 71). Debe ser una ruta absoluta válida dentro de la máquina en la que está corriendo la aplicación. Si no existe la carpeta de destino se creará automáticamente al generar el primer feed.

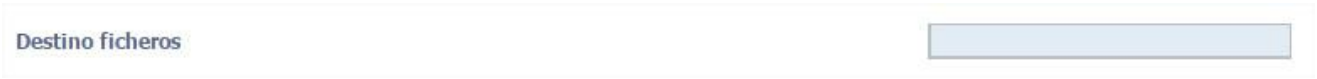


Figura 71. Sincronización feed – Destino feed

En la parte inferior de la pantalla (Figura 72) se muestrán los sites que hemos elegido en la pantalla previa. Para cada site se elegirá el nombre que tendrá el fichero resultante y la frecuencia con la que se generará el mismo.

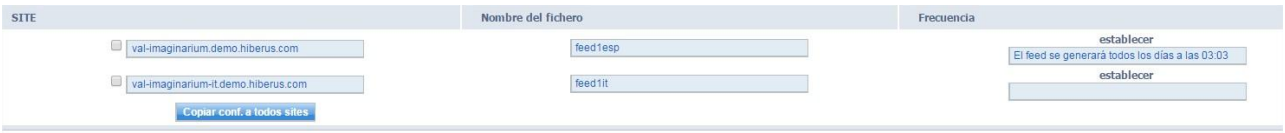


Figura 72. Sincronización feed – Sites

Si se pulsa el botón establecer se abrirá una ventana que nos permitirá establecer la frecuencia con la que se quiera generar el fichero (Figura 73). Lo más habitual es elegir una frecuencia diaria ya que en Imaginarium se hace una sincronización de los productos de su ERP a la base de datos de la web diariamente.

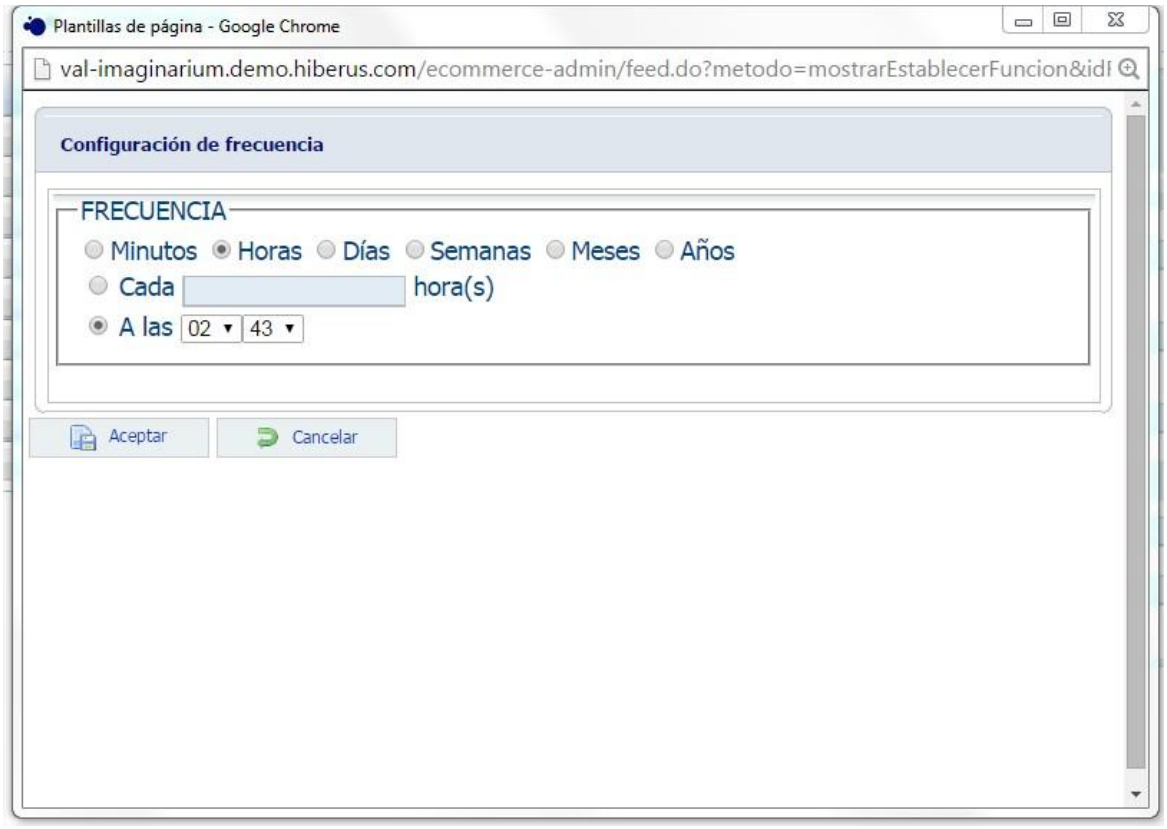


Figura 73. Sincronización feed – Frecuencia

Una vez configurada la frecuencia si se pulsa el botón “Guardar” nos llevará al listado de feeds.

4. Listado de feeds

Figura 74. Listado de feeds

Al listado de feeds se puede acceder de dos formas: tras configurar la generación de un feed como se acaba de comentar, o desde el menú principal del sistema Product Feed.

En esta pantalla se pueden observar todos los feeds que tenemos configurados (Figura 74). Para un mismo feed es posible tener configurados varios sites, por lo que aparecerá una línea por cada site ya que pueden tener frecuencias de generación diferentes. En esta pantalla se nos muestra información acerca de los procesos de generación de cada feed: si están activos, la fecha de la última vez que se generó y la fecha de la próxima vez que se generará. Cuando se está generando un feed, se muestra una barra de progreso que indica el porcentaje del curso del proceso (Figura 75).

Figura 75. Listado de feeds - progreso