

**DESARROLLO WEB CON DRUPAL  
PARA UNA TIENDA DE INSTRUMENTOS MÚSICALES**

**MASTER INTERNACIONAL EN SOFTWARE LIBRE**

**ADMINISTRACIÓN DE WEB Y DE COMERCIO ELECTRÓNICO  
EN ENTORNOS DE SOFTWARE LIBRE**

**AUTOR:  
Mikel Ruiz Echeverría**

**TUTOR:  
Francisco Javier Noguera Otero**

**12 DE JUNIO DE 2.011**

*Copyright (c) 2011 Mikel Ruiz. Se otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre de GNU, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin Secciones Invariantes ni **Textos de Cubierta Delantera ni Textos de Cubierta Trasera**..*

## RESUMEN:

El objeto del proyecto descrito en esta memoria es realizar una tienda virtual para la venta de instrumentos musicales utilizando la plataforma drupal.

Este proyecto surge de la necesidad real de una empresa que ofrece servicios de asesoramiento y venta de instrumentos musicales, de darse a conocer en Internet y de disponer de una aplicación con la que mantener actualizados los servicios que proporciona.

Mediante la adaptación de la plataforma drupal a las necesidades habidas, se ha conseguido un portal donde el cliente tiene acceso de manera directa a la información de la empresa y donde puede consultar ordenados por tipos, familias y fabricantes los productos disponibles en la tienda.

Para su consecución, han sido claves los módulos de vistas y taxonomía que proporciona la plataforma. El desconocimiento inicial de todas sus posibilidades por parte del autor, ha dilatado la implantación con respecto a la previsión inicial. Pero los objetivos están cubiertos y, una vez conocida la potencialidad de las herramientas, se proponen mejoras de cara a un futuro.

En cuanto al mantenimiento de los contenidos, drupal es cómodo de utilizar y no presenta dificultades a la hora de actualizar el contenido.

Visto el volumen de productos que maneja el cliente, es deseable adoptar mecanismos de carga masiva de productos haciendo uso nuevamente de los módulos prediseñados para drupal. En este caso nos referimos al módulo *node import*.

En cuanto al diseño, ha sido necesario incorporar páginas intermedias creadas expresamente en html para abandonar el diseño predefinido de la plataforma y enlazar menús con vistas de la forma requerida.

En el caso de las fichas de producto, se ha utilizado el diseño estándar de drupal, susceptible de mejora en futuras versiones.

Por lo demás, se ha obtenido como resultado un portal de fácil navegación, con costes de programación bajos (respecto a la alternativa de solución 1 estudiada) y con alto potencial de ampliación.

## ÍNDICE DE CONTENIDOS

|  |    |
|--|----|
| RESUMEN:.....  | 3  |
| 1. INTRODUCCIÓN.....   | 6  |
| 1.1 Objetivos del Proyecto.....                                    | 7  |
| 1.2 Fases.....   | 7  |
| 2. ESTUDIO DE VIABILIDAD.....                                      | 8  |
| 2.1 Objetivos y alcance del proyecto.....                          | 8  |
| 2.2 Definición de los requisitos del sistema.....                  | 9  |
| 2.2.1. Requisitos técnicos.....                                    | 9  |
| 2.2.2. Requisitos operativos.....                                  | 9  |
| 2.2.3. Requisitos legales.....                                     | 10 |
| 2.2.4. Requisitos económicos.....                                  | 10 |
| 2.3 Estudio de alternativas de solución.....                       | 10 |
| 2.3.1. Solución 1.....   | 11 |
| Lenguaje de los documentos web.....                                | 11 |
| Servidor Web y generación dinámica de contenido: Apache y PHP..... | 12 |
| Servidor de bases de datos: MySQL.....                             | 13 |
| 2.3.2. Solución 2.....   | 14 |
| Plataformas de software.....                                       | 15 |
| Evaluación las plataformas. Características necesarias.....        | 16 |
| Plataforma Drupal.....   | 17 |
| 2.3.3. Estudio de las alternativas de solución.....                | 19 |
| 2.3.4. Selección de la solución.....                               | 20 |
| 3. ANALISIS DEL SISTEMA DE INFORMACION.....                        | 20 |
| 3.1. Requisitos funcionales de la tienda.....                      | 21 |
| 3.1.1 Estructura de las páginas.....                               | 21 |
| Cabecera.....  | 21 |
| Menú cabecera.....   | 21 |
| Contenido.....   | 22 |
| Menú izquierdo.....  | 22 |
| 3.1.2. Catálogo.....   | 22 |
| 3.1.3. Fichas de los productos.....                                | 23 |
| 3.2. Requisitos no funcionales de la tienda.....                   | 23 |
| 3.2.1. Estandarización.....  | 23 |
| 3.2.2. Usabilidad.....   | 23 |
| 3.2.3. Modularidad.....  | 24 |
| 3.2.3. Rendimiento.....  | 24 |
| 3.3. Requisitos funcionales del módulo de mantenimiento.....       | 24 |
| 3.3.1. Productos.....  | 24 |
| 3.3.2. Agrupaciones de los productos.....                          | 24 |
| 3.3.3. Menús.....  | 25 |
| 3.3.4. Taxonomía.....  | 25 |
| 3.4. Requisitos no funcionales del módulo de mantenimiento.....    | 25 |

|  |    |
|--|----|
| 4. DISEÑO DEL SISTEMA DE INFORMACIÓN.....                      | 25 |
| 4.1. Modelo de casos de uso.....                               | 25 |
| 5. DESARROLLO DEL SISTEMA DE INFORMACIÓN.....                  | 30 |
| 5.1. Arquitectura de Drupal. Estructuración de contenido.....  | 30 |
| 5.1.1. Nodos.....  | 31 |
| 5.1.2. Módulos.....  | 32 |
| 5.1.3. Bloques y menús.....                                    | 32 |
| 5.1.4. Usuarios y permisos.....                                | 33 |
| 5.1.5. Plantillas.....   | 34 |
| 5.1.6. Taxonomía.....  | 34 |
| 5.2 Instalación del servidor.....                              | 34 |
| 5.2.1 Módulos.....   | 35 |
| 5.2.2 Roles y permisos.....                                    | 36 |
| 5.2.3 Configuración tienda Ubercart.....                       | 36 |
| 5.2.4 Taxonomía.....   | 36 |
| 5.3 Vistas.....  | 37 |
| 5.3.1 Vistas de instrumentos por familias.....                 | 38 |
| 5.3.2 Vista de instrumentos con argumento.....                 | 39 |
| 5.3.3 Vistas de accesorios por familias.....                   | 41 |
| 5.3.4 Vista de accesorios con argumento.....                   | 43 |
| 5.3.5 Vista de productos por marca.....                        | 44 |
| 5.3.6 Vistas de productos en oferta.....                       | 45 |
| 5.6 Estructura: páginas y menús.....                           | 46 |
| 5.6.1 Página de inicio.....                                    | 46 |
| 5.6.2 Páginas: Quienes somos, Situación, Contacto.....         | 46 |
| 5.6.3 Páginas: Instrumentos, Accesorios, Marcas y Ofertas..... | 48 |
| 5.6.4 Menús.....   | 49 |
| 5.7. Carga de productos.....                                   | 50 |
| 5.7. Plantillas.....   | 51 |
| 6. CONCLUSIONES.....   | 51 |

# 1. INTRODUCCIÓN

Vivimos en un mundo en el que las nuevas tecnologías están presentes en prácticamente todas las tareas cotidianas que realizamos a diario. De todas ellas, no podemos decir que las tecnologías informáticas sean las más importantes en nuestro día a día, pero por su parte, nadie puede negar que nos acercan y nos hacen más fáciles y cómodas muchas acciones.

Son infinitas las posibilidades que se nos ofrecen con tan solo un click de ratón. Cómodamente desde nuestro sofá y equipados únicamente con un ordenador conectado a internet por medio de la tecnología podemos realizar una instancia en la página web de nuestro ayuntamiento, realizar una transferencia a una cuenta bancaria en la otra parte del planeta o realizar una compra en cualquier lugar del mundo.

Sin duda, uno de los ámbitos que más ha avanzado en los últimos tiempos es el comercio electrónico. Gracias a ello, podemos buscar productos en las diferentes tiendas online de empresas y comercios de todo el mundo e incluso pueden ser los propios particulares quienes ponen a la venta sus productos en diversas páginas web de compra y venta.

El auge del comercio electrónico que vivimos en estos últimos tiempos ha hecho que sean muchas las empresas que disponen de una página web propia para ofrecer productos a los clientes a través de la red. En ocasiones la única funcionalidad es la de publicar los productos, de forma que los clientes podemos ver desde casa lo que el comerciante puede ofrecernos. En otras, la página web nos permite además realizar la compra online del producto, efectuar el pago del importe correspondiente y concretar los datos del envío del producto a nuestro domicilio.

Este hecho ha supuesto que cada vez sean más las empresas que ofrecen sus servicios únicamente a través de internet, sin estar físicamente ubicadas en una sede o local a disposición del cliente, simplemente operando y gestionando todas las tramitaciones de la empresa por medio de una página o portal web.

A pesar de todo ello, el desarrollo de una página o portal web de estas características, su gestión y posterior mantenimiento suponen un elevado coste que no todas las empresas están en condiciones de asumir.

Por ello, un elemento que se ha querido tener en cuenta para este desarrollo es la utilización en todo momento de elementos open source que aporten, además de otras funcionalidades, la ventaja de contar a coste cero con licencias para el servidor de aplicaciones, base de datos y sistema operativo.

En este contexto, surge la necesidad de Gune Music S.L. de desarrollar una aplicación que ofrezca su oferta de una manera comprensible y atractiva, permitiendo al mismo realizar su mantenimiento y actualización de forma rápida y cómoda.

## **1.1 Objetivos del Proyecto**

La empresa Gune Music S.L. se dedica a la distribución, venta y mantenimiento de equipamiento e instrumentos musicales ofreciendo un servicio profesional y experimentado a empresas, instituciones, particulares, orquestas y grupos musicales.

Con el fin de dar difusión a toda su oferta, Gune Music S.L. ha decidido desarrollar un sitio web corporativo con los siguientes objetivos:

-Presentar la información del tipo “presencial” (p. ej. “Quiénes somos” “Qué ofrecemos”) de forma visualmente clara y atractiva. Un sitio web para una empresa es un poderoso canal de comunicación con los clientes y el mercado potencial. Al mismo tiempo representa una parte fundamental de la imagen corporativa de la empresa, que ayuda a identificar la empresa con unos valores, en este caso, los propios de una tienda de calidad.

-Disfrutar de un sistema de gestión de contenidos que facilite su actualización y haciendo que ésta pueda ser llevada a cabo por personas no técnicas. No se pretende diseñar una web espectacular, sino una web bien estructurada y actualizada donde resulte sencillo encontrar información y donde se puedan gestionar y modificar los contenidos sin tener nociones de programación.

Podemos concluir, por tanto, que se trata de crear una web con contenido dinámico, fácilmente gestionado y que permita consultar de manera eficiente los productos de la empresa.

## **1.2 Fases**

Para llevar a cabo este proyecto basado en herramientas Web dentro de entornos de software libre, se ha completado un proceso que parte de la comprensión de la problemática y culmina con la implantación y mantenimiento de la solución elegida.

A lo largo del presente documento se recogen los puntos más destacados de cada una de las fases que se han seguido a lo largo de la ejecución del proyecto. Estas fases son las siguientes:

- Estudio de viabilidad: se estudiará en líneas generales qué problemas se desean resolver, qué soluciones posibles existen y cuál de ellas es la más adecuada.
- Análisis: se describirá detalladamente el sistema que se desea construir, qué requisitos debe cumplir y a qué usuarios debe satisfacer.
- Diseño: se realizará el planteamiento tecnológico de la solución.
- Desarrollo: se llevará a cabo la programación, integración, instalación, etc. de los diferentes subsistemas que compongan el proyecto.

## 2. ESTUDIO DE VIABILIDAD

### 2.1 *Objetivos y alcance del proyecto*

Como se ha señalado en la introducción, se pretende con este proyecto realizar mediante el uso de herramientas de software libre la implantación de un portal web que permita a la empresa Gune Music dar oferta a sus productos a través de Internet. Los objetivos son los siguientes:

-Presentar la información del tipo “presencial” (p. ej. “Quiénes somos” “Qué ofrecemos”) de forma visualmente clara y atractiva.

-Disfrutar de un sistema de gestión de contenidos que facilite su actualización y haciendo que ésta pueda ser llevada a cabo por personas no técnicas.

Desde el punto de vista económico, para que el proyecto sea viable deberá implicar el menor gasto posible, dado que el coste de este proyecto no estaba contemplado inicialmente en los presupuestos anuales de la empresa.

Desde el punto de vista legal, se pretende conseguir un dominio de albergue del sitio que sea comercialmente atractivo.

A nivel técnico, no se impone condiciones restrictivas por parte de la empresa, aunque sí es necesaria una solución que permita flexibilidad a la hora de incorporar modificaciones o nuevos módulos en un futuro.

GuneMusic S.L. desea tener el sistema en marcha lo más rápidamente posible, aunque sólo conocen los requisitos esenciales que debe cumplir.

Por eso es necesario cierto margen de flexibilidad durante el desarrollo de manera que adoptaremos una metodología ágil mediante la que generaremos una primera versión que cumpla con las características básicas, pero que tenga una buena base para ser mejorado y ampliado progresivamente.

Entonces, el alcance del proyecto consistirá en construir una versión de la tienda que satisfaga las necesidades actuales y cuyas funcionalidades puedan ser ampliadas en un futuro en función de los resultados y de la experiencia con el sistema.

En este proyecto intervienen varios departamentos de la empresa. Implicar activamente a sus miembros y haberles mantenido informados sobre los avances, ha permitido que el análisis de requisitos y la posterior validación de las soluciones, se hayan realizado de manera casi directa

Principalmente se han visto implicadas las siguientes unidades:

- *Marketing*: el diseño a nivel gráfico y de contenidos del sitio web es liderada por el equipo del departamento de marketing, de manera que la información de la empresa y sus productos se ofrezca con la claridad y sencillez deseadas.



- *Administración*: se abre un nuevo canal de comercialización de los servicios de la empresa y por tanto la actualización y carga de nuevos productos comportará labores administrativas de mantenimiento periódico del portal.

- *Administración*: el hecho de ofrecer la contratación de servicios mediante Internet puede suponer que aumente el número de clientes de otros puntos geográficos, ya que inicialmente la oferta comercial de la empresa en Internet suele ser más palpable para este público. Este hecho puede implicar que las personas que a día de hoy realizan las entregas a domicilio, puedan ver incrementado su volumen de trabajo o la distancia de los recorridos.

## **2.2 Definición de los requisitos del sistema**

Para realizar la toma de requisitos de forma eficiente, se han ido organizando reuniones con los usuarios, procurando documentar lo recabado para poder consultar en caso de dudas posteriores.

De este modo se han ido recopilando y definiendo todos los requisitos que debe cumplir el sistema. A continuación se han catalogado los requisitos recogidos estableciendo las prioridades de cumplimiento (la prioridad está indicada como un número entre 0 y 100, siendo 100 el prioritario).

### **2.2.1. Requisitos técnicos**

- (100) Arquitectura: el contenido del sitio web deberá poderse administrar mediante la utilización de cualquier navegador.
- (80) Arquitectura: el contenido del sitio web deberá estar almacenado en un sistema gestor de bases de datos relacionales, sobre el cual se puedan realizar futuras consultas no previstas en la actualidad.
- (80) Seguridad: el contenido del sitio web únicamente podrá ser modificado por aquellas personas autorizadas para ello.
- (80) Seguridad: se podrán realizar copias de seguridad por separado y conjuntamente del contenido del sitio web y de la forma en que éste se mostrará.
- (80) Normativas y/o estándares: el sitio web deberá cumplir con los estándares marcados por el World Wide Web Consortium (HTML, CSS, etc.).
- (80) Normativas y/o estándares: el sitio web deberá cumplir con las normas de accesibilidad marcadas por el World Wide Web Consortium (Web Accessibility Initiative).

### **2.2.2. Requisitos operativos**

- (100) Operativa: el sitio web deberá ser visualmente atractivo.
- (10) Operativa: el sitio web deberá poder ser consultado, manteniendo sus características visuales, a través de dispositivos diferentes a ordenadores personales que posean conexión a Internet y un navegador web, tales como televisores, PDA (*personal digital assistant*), etc.
- (80) Operativa: el sitio web deberá posibilitar la visualización de cualquier tipo de contenido multimedia (texto, gráficos, vídeos, etc.).
- (90) Operativa: el sitio web deberá tener una estructura clara, ordenando el contenido de éste en secciones y subsecciones que abarquen cualquier aspecto de los que se quieran comunicar.
- (100) Operativa: el sitio web deberá permitir la contratación de paquetes de horas a través del pago de éstas con tarjeta de crédito, generándose la correspondiente factura, pedido, etc.

- (100) Operativa: la gestión del contenido del sitio web deberá poder ser realizada por una persona no técnica, es decir, que no tenga conocimientos de HTML, JavaScript, etc., de forma fácil e intuitiva.
- (40) Administración: la administración del sitio web (consulta de estadísticas, mantenimiento de cachés, etc.) deberá poder realizarse a través de un navegador web.

### **2.2.3. Requisitos legales**

- (60) La licencia de uso del software de gestión de contenidos debe ser lo menos restrictiva posible.
- (60) La licencia de uso del sistema operativo del servidor web debe ser lo menos restrictiva posible.

### **2.2.4. Requisitos económicos**

- (80) En el caso de ser necesario un gasto en concepto de licencia de uso del software de gestión de contenidos, éste deberá ser lo más pequeño posible.
- (80) El gasto correspondiente al sistema operativo del servidor web debe ser lo más pequeño posible.

## ***2.3 Estudio de alternativas de solución***

Continuamos nuestro estudio con el fin de encontrar la solución que permita satisfacer las necesidades expuestas en el apartado anterior. Se deberán contemplar igualmente una serie de requisitos no funcionales como usabilidad, fiabilidad y robustez.

Teniendo claras dichas necesidades podemos pasar a estudiar cuál es el mejor modo de resolver el problema planteado.

En este momento es útil dejar apartados los detalles (por un momento) y ofrecer una visión de conjunto: interesa mostrar la estructura de la aplicación y cómo interactúan entre si sus partes.

Las tareas que debe realizar la aplicación pueden resumirse en los siguientes grupos:

· Visualización de información

El cliente pide a la aplicación que le proporciona diversa información: listado de productos, fabricantes, etc.

· Actualización de información

El cliente puede editar información ya existente o añadir nueva información.

· Persistencia de información

La información añadida por el cliente (o los cambios realizados sobre información existente) debe guardarse.

En resumen la aplicación permite editar, guardar y visualizar información. Lo que finalmente nos lleva a una estructura como la indicada en la siguiente imagen:



El cliente envía peticiones al servidor; el servidor procesa dichas peticiones recuperando información de la base de datos o almacenando datos en la base de datos según convenga.

La información fluye entre el servidor y el cliente (por ejemplo el servidor envía al cliente el producto solicitado), y entre el cliente y el servidor (el cliente envía al servidor modificaciones de precios o nuevos productos a dar de alta).

Asimismo la información fluye entre el servidor y la base de datos (el servidor almacena un nuevo producto suministrado por el usuario en la base de datos) y la base de datos y el servidor (la base de datos envía el producto solicitado al servidor).

La aplicación descrita es una aplicación web estructurada en tres capas: navegador web, servidor web de contenido dinámico y base de datos. Para el desarrollo de la aplicación es necesario seleccionar la tecnología adecuada para cada nivel teniendo en cuenta que las tres capas deben comunicarse entre sí.

### **2.3.1. Solución 1**

La primera solución que hemos contemplado para el desarrollo del portal está basada en la utilización de PHP para la generación de páginas dinámicas que completen su contenido mediante consultas a la base de datos albergada en MySQL. Las páginas generadas serán servidas por Apache.

Se trata de tecnologías de desarrollo web libres. La unión de Apache Server para el servidor web, PHP para el lenguaje de programación y MySQL para el sistema de administración de base de datos proporcionan a la aplicación un conjunto de herramientas muy útiles para su desarrollo.

#### *Lenguaje de los documentos web*

El cliente final interactuará con la aplicación por medio de un navegador web: es decir va a navegar por un conjunto de páginas web. Una página web es un documento que contiene información estática (texto, imágenes, vídeo, sonido) junto a un conjunto de etiquetas que determinan el formato de dicha información.

Los lenguajes y tecnologías más usados en este nivel son: HTML, CSS y Javascript

#### *HTML*

La base de toda página web: son un conjunto de *tags* (etiquetas) que determinan el formato de la información (cursiva, negrita, salto de página, etc.).

#### *CSS*

Lenguaje usado para determinar la presentación de la información de una página web. El objetivo de dicho lenguaje es separar la presentación del contenido de una página web del propio contenido. Realiza tareas de las que antiguamente se encargaba el HTML con las siguientes ventajas: la información de presentación ya no esta repetida y dispersa entre varias páginas web sino descrita en un único lugar. Esto agiliza en gran medida el proceso de modificación del *look and feel* de un determinado sitio web.

#### *Javascript*

Es un lenguaje de *scripting* que se ejecuta en el cliente web (el navegador del cliente) y que permite añadir un toque de dinamismo a las páginas web estáticas descritas en este apartado. El Javascript permite a la aplicación responder a acciones del cliente tales como movimiento del ratón, clic del botón, etc.

Los lenguajes descritos anteriormente son la base de toda página web. Otra tecnología usada es Flash.

Flash es una plataforma que permite añadir animaciones y interactividad a la página web y que es bastante atractiva visualmente.

### *Servidor Web y generación dinámica de contenido: Apache y PHP*

Para implementar el proyecto va a ser necesario también un servidor web: una aplicación que genera las páginas web solicitadas por el cliente y que se las envíe a éste. Interesa además que la aplicación pueda generar las páginas web solicitadas por el cliente de un modo dinámico. Es necesario, por tanto, seleccionar un servidor web y un lenguaje para la generación de páginas web dinámicas soportado por el servidor.

*Apache Web Server*, es un servidor de páginas Web desarrollado por la Apache Software Foundation, organización formada por miles de voluntarios que colaboran para la creación de software de libre distribución.

Es uno de los servidores más utilizados en Internet ya que se trata de un servidor muy potente, flexible, rápido, eficiente y que siempre está adaptado a nuevos protocolos http. Apache se encuentra disponible para varias plataformas, desde Debian, hasta Windows XP y se le puede incrustar nuevos módulos que le permitirán ejecutar código Script como son JSP, PHP, etc.

#### Ventajas

- Modular

- Open source
- Multi-plataforma
- Extensible
- Bien documentado
- Gratuito

Otro popular servidor web es *ISS de Microsoft*, aunque diferencia de Apache es de código cerrado, no gratuito y sólo soporta la plataforma Windows.

Existen varios lenguajes que permiten generar contenido de un modo dinámico: PHP, Perl, Ruby, etc. Conviene destacar de entre ellos PHP porque es un lenguaje de "código abierto" interpretado, de alto nivel, embebido en páginas HTML y ejecutado en el servidor. Es usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Su interpretación y ejecución se da en el servidor web, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

*PHP* también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

#### Ventajas

- Alto rendimiento
- Bajo coste
- Interfaces para una gran cantidad de sistemas de base de datos
- Facilidad de aprendizaje y uso
- Portabilidad
- Detallada documentación y amplia comunidad de desarrolladores
- Acceso al código abierto
- Gran variedad de funciones integradas

#### Servidor de bases de datos: MySQL

Las bases de datos permiten almacenar y estructurar grandes volúmenes de información. Existen varios servidores de Bases de Datos fiables tanto gratuitas como de pago: IBM DB, MySQL, Oracle, etc.

MySQL es el sistema de administración de bases de datos (Database Management System, DBMS) más popular, desarrollado y proporcionado por MySQL AB. Es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

MySQL fue escrito en C y C++ y destaca por su gran adaptación a diferentes entornos de desarrollo, permitiendo su interacción con los lenguajes de programación más utilizados como PHP, Perl y Java y su integración en distintos sistemas operativos.

También es muy destacable, la condición de open source de MySQL, que hace que su utilización sea gratuita e incluso se pueda modificar con total libertad, pudiendo descargar su código fuente. Esto ha favorecido muy positivamente en su desarrollo y continuas actualizaciones, para hacer de MySQL una de las herramientas más utilizadas por los programadores orientados a Internet.

Según las cifras del fabricante, existirían más de seis millones de copias de MySQL funcionando en la actualidad, lo que supera la base instalada de cualquier otra herramienta de bases de datos.

Características MySQL:

- Velocidad. MySQL es rápido.
- Facilidad de uso. Es un sistema de base de datos de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes.
- Coste. Es gratuito.
- Capacidad de gestión de lenguajes de consulta. MySQL comprende SQL, el lenguaje elegido para todos los sistemas de bases de datos modernos.
- Capacidad. Pueden conectarse muchos clientes simultáneamente al servidor. Los clientes pueden utilizar varias bases de datos simultáneamente. Además, está disponible una amplia variedad de interfaces de programación para lenguajes como C, Perl, Java, PHP y Python.
- Conectividad y seguridad. MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar de Internet. Dispone de control de acceso.
- Portabilidad. MySQL se puede utilizar en una gran cantidad de sistemas Unix diferentes así como bajo Microsoft Windows.
- Distribución abierta. Puede obtener y modificar el código fuente de MySQL

### **2.2.1. Solución 2**

Llegados a este punto está claro que seleccionando un servidor web, lenguaje de programación y base de datos sería posible implementar la aplicación desde cero. Las ventajas de semejante aproximación es que es posible implementar el proyecto decidiendo hasta el más mínimo detalle cada uno de los detalles técnicos de implementación. Sin embargo semejante proyecto sería costoso de implementar (proyecto de larga duración), de mantener, modificar y menos escalable.

Otra opción es implementar una aplicación sobre una plataforma de software ya existente: dichas plataformas tienen implementadas funcionalidades tales como login, registro de usuarios, despliegue de menús, etc. Esto supone un significativo ahorro de

tiempo de cara a la implementación de la aplicación e incluso posterior mantenimiento (en forma de actualizaciones para corregir vulnerabilidades/errores etc.).

### Plataformas de software

Antes de proceder a evaluar las posibles plataformas existentes toca describir, aunque sea brevemente, que es una plataforma de software. En este proyecto se llama plataforma de software a un conjunto de herramientas de software integradas que se utilizan para crear aplicaciones informáticas.

Una plataforma aúna varias tecnologías existentes en un entorno cohesivo creando una base que servirá para el desarrollo de futuras aplicaciones.

Veamos a continuación un ejemplo: tomamos un lenguaje de programación ya existente (PHP por ejemplo), y añadamosle un conjunto de librerías software (código que realiza una tarea particular preparado para reutilizarse en cualquier aplicación que requiera la funcionalidad proporcionada por dicho código). Le añadiremos un lenguaje para bases de datos (SQL) y además usaremos dicho lenguaje para crear una estructura de datos donde guardar información, así como un conjunto de funciones que recuperan la información almacenada en esa estructura.

De momento estamos proporcionando al usuario una herramienta que le permite almacenar y recuperar información de un modo más rápido que si este hubiera tomado las tecnologías que ya existían (PHP y SQL) y se hubiera puesto a programar dicha funcionalidad desde cero.

La herramienta descrita consta de código prefabricado para realizar algunas funciones más una estructura para almacenar información.

Podríamos dar un paso más y preparar un entorno visual para el programador: podríamos incluir un conjunto de páginas web que permita introducir datos por forma de un formulario web y un conjunto de páginas web para visualizar dichos datos. Puestos a ello también podemos extender la funcionalidad de la herramienta añadiendo una interfaz gráfica prefabricada para el usuario.

Todas las interfaces descritas hasta ahora, por supuesto, deberían ser extensibles y modificables (por ejemplo, permitir añadir formularios a los proporcionados por defecto) ya que la herramienta gráfica proporcionada es de carácter genérico mientras que cada proyecto informático debe resolver un cúmulo de tareas específicas.

La herramienta descrita en el ejemplo anterior aglutina varios lenguajes distintos (SQL y PHP), sobre una estructura de datos con código preparado para dar solución a ciertas tareas (librerías PHP, funciones SQL para trabajar sobre la información almacenada), y un entorno gráfico (interfaces gráficas) para manipular/mostrar la información almacenada.

Todo este conjunto de herramientas funciona de un modo cohesivo: es decir no trabajamos con los distintos componentes de la herramienta sino con el conjunto. Además, es preciso tener en cuenta que la herramienta descrita no sólo proporciona un conjunto de software sino una arquitectura (estructuras de datos, flujos de relaciones entre los distintos componentes, etc.), una metodología de trabajo, y una serie de reglas para su expansión (todos los mecanismos que la herramienta incluya para adaptar una base genérica a un proyecto específico).

Por lo tanto es interesante evaluar qué plataformas de software ya existentes pueden usarse para implementar el proyecto y qué ventajas aportan sobre una aproximación “make from scratch”.

### Evaluación las plataformas. Características necesarias

Para ello veamos qué características son necesarias o deseables en la aplicación a implementar:

- Soporte para Firefox e Internet Explorer (requisito indicado en el apartado de definición de requisitos)
- Aplicación gratuita o con mínimo coste de licenciamiento (requisito incluido en la definición de requisitos)
- Interfaz amigable de cara al usuario final (requisito incluido en la definición de requisitos.)
- La plataforma debe ofrecer un soporte adecuado a la creación de interfaces de usuario.
- La plataforma escogida debe proporcionar soluciones a las necesidades mencionadas anteriormente, pero existen otras características que también es importante que la plataforma escogida cumpla:
- Soporte para funcionalidades comunes. Varias de las funcionalidades a implementar en el proyecto son comunes con cierto tipo de aplicaciones habituales y por ello se encuentran ya implementadas. Usar una implementación existente no sólo elimina tareas a realizar en el proyecto final sino que también supone una ventaja de cara al mantenimiento ( parches y actualizaciones proporcionados por el fabricante ).
- Flexibilidad en la ampliación y modificación Como hemos indicado, puede ocurrir que a las funcionalidades proporcionadas será necesario añadir las necesarios para satisfacer las necesidades del proyecto. Debe darse una integración entre el nuevo código creado y el código existente de modo que el conjunto funcione correctamente y su mantenimiento no se deteriore.
- Plataforma popular y establecida: una plataforma popular de uso extendido significa que existe un mayor número de personas familiarizadas con dicho plataforma. Encontrar desarrolladores para dicha aplicación es más fácil. Asimismo una plataforma ya establecida, pasada su etapa inicial, y de uso extendido ofrece mejores garantías de continuar existiendo y proveyendo servicio en los años posteriores a la implementación del proyecto.



- Documentación de calidad: una buena documentación reduce considerablemente el tiempo de implementación de una aplicación disminuyendo su coste. También repercute favorablemente en la calidad final de la misma ya que permite aprovechar al máximo las funcionalidades proporcionadas por la aplicación.
- Comunidad activa: una aplicación popular con una fuerte comunidad ya existente funciona como un complemento a una buena documentación con las mismas ventajas que esta conlleva.

### Plataforma Drupal

Existen varias plataformas que cumplen en mayor o menor medida los requisitos comentados anteriormente. Entre ellas se encuentran Wordpress, Joomla, Drupal, etc.

Drupal es más genérica que Wordpress (CMS usado para crear blogs) aportando más posibilidades en la creación de páginas web que no se ajusten al formato blog.

En lo que respecta a Joomla es un excelente CMS y es difícil establecer una clara superioridad de Drupal sobre Joomla o a la inversa. Joomla es ligeramente más sencillo de utilizar, mientras que en estos momentos Drupal ofrece una estructura menos rígida en la estructuración del contenido. De cara a una mayor versatilidad en futuras ampliaciones Drupal es más adecuado.

Por ello, aunque cualquiera de las mencionadas podría ser utilizada para implementar el proyecto, Drupal entendemos que presentaría las condiciones idóneas y que podíamos seleccionarla como plataforma de software sobre la que hacer el desarrollo.

Drupal es un sistema de gestión de contenido modular y multipropósito. Nos encontramos ante una solución PHP mantenida por una gran comunidad de usuarios. Existen disponibles grandes cantidades de módulos adicionales desarrollados por la comunidad y usuarios de Drupal.

De entre ellos, ubercart está orientado a la creación de un sitio web que funcione como tienda online de un comercio. Está desarrollado en lenguaje PHP y hace uso de una base de datos MySQL. El hecho de ser un módulo exclusivamente desarrollado para el gestor de contenidos Drupal hace que la integración sea perfecta, lo cual es una gran ventaja a tener en cuenta.

Una característica propia de Drupal en este aspecto es la existencia de dos roles por defecto, el rol Usuario anónimo y el rol Usuario autenticado. Estos dos roles no pueden ser borrados del sistema, ya que siempre tienen una utilidad. El rol Usuario anónimo se asocia a los usuarios que acceden al portal y no han iniciado sesión en el sistema con ninguna cuenta de usuario, por lo que sus permisos deberían ser reducidos. Por su parte, el rol Usuario autenticado se asocia a cualquier usuario que ha iniciado sesión

en el sistema con una cuenta de usuario y será utilizado para la actualización y mantenimiento del catálogo de productos.

Drupal es totalmente independiente de la plataforma sobre la que se ejecute. A pesar de que la mayoría de sistemas Drupal utilizan bases de datos MySQL, Drupal incluye soporte para el uso de otro tipo de bases de datos. A su vez, soporta Apache o Microsoft IIS como servidor web, soporta diferentes sistemas operativos como Windows, Linux, Mac OS X o Solaris y es soportado por cualquier navegador.

Una de las funcionalidades características de Drupal nos la ofrece su módulo Taxonomy, el cual nos permite crear y administrar una taxonomía con la cual podremos categorizar y clasificar todo el contenido que creamos con el gestor. Dentro de la taxonomía podremos crear diferentes vocabularios, dentro de los cuales podremos a su vez crear diferentes términos.

De esta forma, podremos crear un vocabulario con términos para clasificar según categorías los productos de la tienda online.

Drupal nos permite crear y configurar un catálogo con diferentes categorías, publicar productos en dicho catálogo, categorizar los productos o realizar el proceso de compra por medio de un carrito de compra y diversas plataformas de pago. Además, es posible crear nuevos atributos para los productos.

Por último, destacaremos otra importante característica del gestor de contenidos Drupal. Se trata del uso de plantillas con el fin de separar el contenido y la presentación, permitiendo así controlar o cambiar de forma sencilla el aspecto o apariencia del portal. Dichas plantillas están escritas en lenguaje HTML o PHP.

En cuanto a la tienda virtual proporcionada por Ubercart, podemos como administradores configurar multitud de opciones, ya sean relacionadas con los clientes, el carrito de compra, los pedidos, los informes de ventas, el stock de los productos, etc. En una fase inicial, no abordaremos la compra online.

Ubercart, por el hecho de ser un módulo de Drupal, hace que sea posible aprovechar todas las ventajas del propio gestor de contenidos. Es posible configurar la forma en que queremos que se visualicen los productos, clasificarlos por comerciante o proveedor o crear nuevas vistas a parte de las incluidas por el propio módulo Ubercart.

Podemos ampliar las funcionalidades de nuestro portal mediante la instalación de diversos módulos Drupal, lo cual hará sin duda más completo nuestro portal, a diferencia de soluciones especializadas y centradas en el comercio electrónico.

Todas esta característica mencionada hemos de saber aprovechar para dar solución a la necesidad de mantenimiento de una estructura ordenada de productos y marcas de instrumentos ofertados por Gune Music S.L.

Por todo lo visto podemos terminar diciendo que Drupal satisface los requisitos fijados para nuestra plataforma.

## 2.2.4. Estudio de las alternativas de solución

Una vez descritas ampliamente las dos alternativas de solución que más se ajustan a los objetivos que plantea el cliente, hemos de tomar la decisión de elegir entre basar el proyecto en un desarrollo propio o utilizar Drupal con Ubercart.

Se ha comprobado que las dos soluciones están soportadas por infraestructuras similares, basadas en ambos casos en Apache, MySQL y PHP.

En cuanto a las funcionalidades exigidas, la Solución 1 cumple pero exigirá la realización de los módulos de programación necesarios para ofrecer y recoger de la Base de Datos la información generada en los procesos de consulta por parte de los clientes o en los procesos de actualización y mantenimiento por parte de los clientes.

En este sentido, la Solución 2 también cumple aunque aprovechará los paquetes genéricos para ecommerce que la comunidad Drupal ya tiene desarrollados. El trabajo irá orientado a conseguir adaptar el funcionamiento de los módulos al funcionamiento requerido por la plataforma.

De la misma forma, el control de los accesos en el primer caso supondrá la implantación de un sistema de validación y gestión de permisos propio mientras que en el segundo se trata de aprovechar la característica de roles que la aplicación trae implementada.

El aprovechamiento de módulos genéricos creados para satisfacer necesidades globales podría presentar eventualmente la desventaja de no ser capaces de realizar la adaptación necesaria para completar un requisito muy particular.

Los requisitos que Gune Music S.L. ha ido planteando, son muy generales y pueden ser satisfechos completamente por las funcionalidades ofrecidas mediante la personalización de los módulos de drupal.

No obstante, en este sentido existe el riesgo de que en un futuro se diera la situación de que una necesidad no pueda solucionarse mediante la utilización de los módulos existentes.

Sin embargo, el impacto de este riesgo es menor ya que Drupal permite llevar incorporadas modificaciones a medida que pueden desarrollarse en php para dar salida a la necesidad no prevista o no solucionada por la comunidad.

Si bien se trata tanto en la Solución 1 como en la Solución 2 de sistemas libres, con costes nulos de licenciamiento, el costo de desarrollo es diferente para la Solución 1 que para la Solución 2.

Mientras las horas de programador se disparan con la propuesta de Solución 1, la necesidad de programación en la Solución 2 es mucho menor ya que con ella se va a hacer uso de paquetes adaptables en la práctica totalidad de las situaciones.

### **2.2.5. Selección de la solución**

Teniendo en cuenta la descripción general que tenemos del sistema, los requisitos que ha de cumplir necesariamente la solución, la descripción de las alternativas posibles y el estudio comparativo asociado a las mismas, vamos a considerar los siguientes factores con el fin de realizar la elección de la solución:

Requisitos planteados: las dos soluciones cubren en mayor o menor medida los requisitos básicos planteados a nivel técnico y funcional. En cuanto a los aspectos económicos, la Solución 2 es más ventajosa que la Solución 1 como hemos descrito en el estudio del apartado anterior.

Análisis Coste/Beneficio: los beneficios potenciales que generaría la implementación de cada solución son similares en términos generales. En cuanto a los costes, como ya hemos indicado, la Solución 2 ofrece ventaja respecto a la Solución 1.

Riesgos: Si bien la Solución 1 no presenta riesgos significativos (obviando la dilatación temporal que pudiera derivarse de su implantación), la adopción de la Solución 2 presenta el riesgo que supondría no disponer o no poder adaptar en su justa medida un módulo concreto para satisfacer un hipotético requisito muy particular. Como hemos visto, en el estudio previo el impacto de este riesgo se ve minimizado con la posibilidad de desarrollar a medida puntos concretos para incorporar al sistema drupal.

Se decide por tanto que la Solución 2 es más adecuada que la Solución 1.

## **3. ANALISIS DEL SISTEMA DE INFORMACION**

Una exhaustiva toma de requisitos es clave para definir y delimitar de forma precisa el funcionamiento del sistema, formalizando un contrato con el cliente a través del cual pueda validar que realmente se satisfacen sus necesidades. Además, el hecho de tener esta fase bien elaborada permitirá luego desarrollar el modelo de casos de uso de manera casi directa.

Para realizar la toma de requisitos de forma eficiente, se han ido organizando reuniones con el cliente y sus colaboradores, sobre las cuales se redactan actas que se envían a todos los asistentes para obtener su conformidad. Con ello se pretende dejar constancia de lo que se ha comentado durante la reunión, y luego tener argumentos a la hora de justificar las decisiones tomadas así como de disponer de documentos de consulta en caso de dudas posteriores.

De este modo se han ido recopilando y definiendo todos los requisitos que debe cumplir el sistema, obteniendo en todo momento una visión global que permita aclarar imprecisiones o posibles inconsistencias.

Diferenciaremos dos tipos de requisitos:

*Requisitos funcionales:* Definen el comportamiento específico del software, es decir, qué debe hacer exactamente la aplicación.

*Requisitos no funcionales:* Complementan los anteriores describiendo cualidades y restricciones de las operaciones sistema, centrándose en características de diseño e implementación.

Primero empezaremos por especificar qué requisitos tendrá la tienda y posteriormente pasaremos a definir qué requisitos deberá cumplir el módulo de mantenimiento.

### **3.1. Requisitos funcionales de la tienda**

#### **3.1.1 Estructura de las páginas**

Todas las páginas tendrán una estructura común que contendrá:

##### Cabecera

Parte reservada para el logo del sitio web. Adicionalmente Drupal permite añadir un menú de navegación secundario en la cabecera que en ocasiones resulta interesante aprovechar, pues nos permite ver donde esta situada la pantalla actual dentro del sistema de navegación.

En nuestro caso el cliente ha pedido deshabilitar esta opción por motivos estéticos.

##### Menú cabecera

Se ha considerado interesante dividir los elementos del menú de cabecera en categorías de modo que los elementos relacionados entre sí se presenten en un mismo grupo. Con esta división se pretende que el usuario final pueda visualizar la máxima información de un sólo vistazo. Dado que un usuario visita el sitio para obtener información y no para perder tiempo buscándola, es interesante hacer el proceso de navegación tan rápido como sea posible y el acceso a la información deseada tan directo como sea posible.

*¿Quiénes somos?:* Consistirá en una presentación de la empresa que podremos encontrar en la primera pestaña empezando por la izquierda sobre el menú de la cabecera.

*Instrumentos:* En la pestaña 2 se mostrarán los instrumentos siguiendo una clasificación por familias (viento, cuerda, percusión, tecla, escolar, regionales y varios).

*Accesorios:* Desde la pestaña 3 estarán accesibles accesorios (fundas, atriles, afinadores, etc.) también clasificados según la familia a la que pertenece el instrumento al que complementan (viento, cuerda, percusión, tecla, varios, cornetas y parada, mobiliario e informática).

*Marca:* En la pestaña 4 aparecerán agrupados los productos (instrumentos y accesorios de cualquier familia) por la marca comercial de su fabricante.

*Oferta:* A continuación encontraremos aquellos productos que presentan algún tipo de oferta.

*Ubicación:* Contendrá un mapa con la ubicación de la tienda y será la segunda por la derecha.

*Contacto:* Contendrá los datos de contacto, así como el horario de apertura y podremos acceder a través de la primera pestaña (por la derecha) del menú de cabecera.

### Contenido

La parte central de la página web queda reservada para visualizar el contenido: en este espacio se muestran los formularios para aquellos casos en los que el usuario debe introducir o modificar información (alta de usuarios, por ejemplo); es el espacio usado, también, para mostrar al usuario el resultado de sus consultas (por ejemplo si un usuario pide ver la lista de obras de un autor, dicho listado se muestra en este espacio).

Aunque para la navegación entre las diversas pantallas hemos reservado el menú de cabecera de la aplicación, la parte de contenido también podrá incorporar elementos de navegación para mejorar la claridad (por ejemplo al visualizar en pantalla los datos del usuario es natural incluir una opción que permita editar dichos datos desde la misma pantalla sin necesidad de obligar al usuario a regresar a los menús laterales).

### Menú izquierdo

*Administración:* (enlace al panel de administración que permite al usuario administrador gestionar el sitio web), alta de productos, *login*, nuevas categorías, modificación de precios y actualización de ofertas.

## **3.1.2. Catálogo**

Se accederá al catálogo virtual a través de las pestañas 2, 3 y 4 del menú de cabecera. Ahí estará recogida la información de todos los artículos que se venden en la tienda: **instrumentos** siguiendo una clasificación por familias (viento, cuerda, percusión, etc.), **accesorios** (fundas, atriles, elementos de limpieza, etc.) también clasificados según la familia a la que pertenece el instrumento al que complementan.

En la pestaña 4 aparecerán agrupados los productos (instrumentos y accesorios de cualquier familia) por la **marca** comercial de su fabricante.

A continuación encontraremos aquellos productos que presentan algún tipo de **oferta**.

Las páginas de familias y de agrupaciones por marca podrán tener una imagen y una descripción previamente al listado de sus productos.

Los productos de una familia se mostrarán mediante vistas que agrupan por la denominación de instrumento o accesorio y por marca.

Los productos de un fabricante se mostrarán mediante vistas que agrupan por denominación ofreciendo listados que mostrarán la descripción y el precio de los productos.

Los productos dispondrán de una ficha propia a la que se accederá mediante link a ella desde la vista y en la que se mostrarán todos los detalles.

Los precios de los productos se mostrarán con IVA incluido.

Los listados con muchos productos aparecerán paginados, aunque se debe poder ver todos los elementos en la misma página.

### **3.1.3. Fichas de los productos**

Se trata de páginas que mostrarán información detallada sobre un producto concreto: nombre, descripción, fabricante, imagen y precio.

## **3.2. Requisitos no funcionales de la tienda**

### **3.2.1. Estandarización**

La web deberá de visualizarse y tener el mismo comportamiento en los exploradores más importantes: Internet Explorer, Mozilla Firefox, Google Chrome, Safari y Opera.

Deberá visualizarse correctamente en una resolución de pantalla de 800x600 píxeles o superior.

Las páginas deberán cumplir con los estándares CSS y XHTML.

Las páginas deberán estar adaptadas y ser accesibles para invidentes y robots de los buscadores

### **3.2.2. Usabilidad**

La navegación en la web será intuitiva y amigable, siendo fácil de usar incluso para personas no habituales en internet.

Todas las páginas tendrán una estructura común para que el visitante se sienta siempre orientado durante su navegación

La web será atractiva y agradable visualmente, acorde a las indicaciones dadas.

### **3.2.3. Modularidad**

Debe ser un sistema escalable, pudiendo añadir fácilmente nuevas funcionalidades (mejoras en el catálogo, secciones, registro de usuarios, etc.)

Debe ser reusable, pudiendo aprovechar en otras áreas las funcionalidades implementadas.

### **3.2.3. Rendimiento**

Las páginas deben ser ágiles y tener un tiempo de carga adecuado.

## **3.3. *Requisitos funcionales del módulo de mantenimiento***

Sólo podrán acceder al módulo de mantenimiento los usuarios identificados. Las funcionalidades que debe cumplir el módulo de mantenimiento las podemos organizar conceptualmente en:

### **3.3.1. Productos**

Los productos disponen de nombre, descripción, datos de catálogo, tipo de producto, fabricante, fotografía precio y si está o no en oferta.

La información básica los productos se cargará de forma masiva desde ficheros csv haciendo uso del módulo node\_import.

Se podrán gestionar las imágenes de los productos.

La descripción deberá soportar formato HTML.

Se podrán activar y desactivar productos para que aparezcan o no en la web.

Se podrá definir un orden para los productos.

### **3.3.2. Agrupaciones de los productos**

Hay establecida una primera agrupación de los productos en función su tipo. Puede tratarse de instrumentos o de accesorios.

A su vez estas agrupaciones se dividen en familias (viento, percusión, cuerda, etc).

También se agrupan los productos en base a la marca comercial de su fabricante.

Se podrá gestionar el Alta/Baja/Modificación de las agrupaciones.

Se podrá definir si la agrupación aparecerá en la web como punto de menú, como página o bien permanecerá oculta.



### **3.3.3. Menús**

Se definirá el orden en que se quiere presentar los apartados del menú de cabecera. Se permitirá agregar nuevos elementos del menú y borrar o modificar las entradas existentes.

### **3.3.4. Taxonomía**

Se establecerá un vocabulario GuneMusic para productos, con terminos para Tipo de Producto (Instrumento o accesorio), Familia de pertenencia (Viento, Cuerda, Percusión, ...), Fabricante, (Adams, Bach,...), Nombre del elemento (Trompeta, Atril, Funda,...), Oferta (Si, No) que permitan clasificar a los productos según los requisitos especificados.

Paralelamente, se crearán vocabularios Instrumentos\_Viento, Instrumentos\_Cuerda, etc para ayudar a la creación de las vistas intermedias.

## **3.4. Requisitos no funcionales del módulo de mantenimiento**

El sistema será accedido a través del navegador web del cliente: Microsoft Internet Explorer, versión 6 o posterior.

Ha de tener una navegación ágil e intuitiva.

Debe de tener un rendimiento aceptable, sin necesidad que sea óptimo.

Las contraseñas no se almacenarán directamente en base de datos, sino que se guardará su *Hash* para validar credenciales.

Se realizarán frecuentemente copias de seguridad de la Base de Datos.

## **4. DISEÑO DEL SISTEMA DE INFORMACIÓN**

Una vez analizados los requisitos del proyecto la siguiente etapa es el diseño del sistema, donde se describirá qué hará y cómo funcionará el sistema sin llegar a definir todavía cómo implementarlo

El objetivo de la fase de diseño es determinar cómo va ser construido el sistema a partir de los requisitos y el modelo obtenidos durante la especificación, aplicando patrones de diseño sin entrar a nivel de tecnología.

### **4.1. Modelo de casos de uso**

El modelo de casos de uso sirve para definir a los actores o usuarios del sistema y la forma en que interactuarán con él, desde el punto de vista del usuario final.

Los casos de uso son representaciones de uno o más escenarios que describen cómo será la interacción entre el sistema y el usuario para alcanzar un único objetivo o tarea del negocio.

Los casos de uso han sido desglosados en función del actor con quien interactúa, mostrando un diagrama de casos de uso del actor y la definición o contrato de cada uno de ellos. En el caso del actor *administrador*, también se han clasificado según la entidad principal con la que relacione.

Con el fin de simplificar el modelo de casos de uso, en los casos básicos de gestión de entidades (alta, baja, consulta y modificación) que tengan un comportamiento común no se entrará en un elevado nivel de detalle, de modo que se tratarán sus datos de forma global sin especificar todos sus atributos.

**Caso de uso**  
**Requisitos asociados**  
**Actor iniciador**  
**Descripción**

**Consultar menú**  
Tienda – Menú corporativo  
Visitante  
Se consulta el contenido de una sección corporativa (¿quiénes somos, ubicación o contacto)

**Precondición**  
**Postcondición**

Se muestra el contenido de la sección por pantalla

**Curso normal**

1. El usuario quiere acceder a una sección.
2. El sistema muestra las secciones existentes.
3. El usuario elige la sección deseada.
4. El sistema muestra el contenido de la sección.

**Curso alternativo**

**Caso de uso**  
**Requisitos asociados**  
**Actor iniciador**  
**Descripción**

**Consultar agrupación de productos**  
Tienda – Catálogo virtual  
Visitante  
Se consultan las agrupaciones asociadas a instrumentos, accesorios o marcas.  
La categoría está activa.  
Se muestran las familias de la agrupación seleccionada

**Precondición**  
**Postcondición**

**Curso normal**

1. El usuario quiere consultar una agrupación.
2. El sistema muestra desde menú las agrupaciones existentes

- (instrumentos, accesorios y marcas).
3. El usuario elige la categoría deseada.
  4. El sistema muestra las familias de la categoría.

**Curso alternativo**

**5. Extensión Consultar familia**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso</b>          | <b>Consultar familia</b>   |
| <b>Requisitos asociados</b> | Tienda – Catálogo virtual  |
| <b>Actor iniciador</b>      | Visitante  |
| <b>Descripción</b>          | Se consulta el contenido de una familia (viento, cuerda, percusión, etc) |
| <b>Precondición</b>         | La familia está activa.  |
| <b>Postcondición</b>        | Se muestran la información y los productos de la familia seleccionada    |

**Curso normal**

1. El usuario quiere consultar una familia.
2. El sistema muestra las familias disponibles.
3. El usuario selecciona la familia deseada.
4. El sistema muestra la información de la familia.

**Curso alternativo**

**5. Extensión Consultar producto**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso</b>          | <b>Consultar producto</b>                                   |
| <b>Requisitos asociados</b> | Tienda – Fichas de productos                                |
| <b>Actor iniciador</b>      | Visitante   |
| <b>Descripción</b>          | Se consulta la información de un producto                   |
| <b>Precondición</b>         | El producto está activo, no es obsoleto y dispone de ficha. |
| <b>Postcondición</b>        | Se muestra la información del producto seleccionado         |

**Curso normal**

1. El usuario quiere consultar un producto.
2. El sistema muestra los productos disponibles.
3. El usuario selecciona el producto deseado.
4. El sistema muestra la información del producto.

**Curso alternativo**

**Caso de uso**

**Iniciar sesión**

|                             |   |
|-----------------------------|---|
| <b>Requisitos asociados</b> | Tienda – Registro de usuarios                                     |
| <b>Actor iniciador</b>      | Cliente   |
| <b>Descripción</b>          | El usuario inicia sesión en el sistema                            |
| <b>Precondición</b>         |   |
| <b>Postcondición</b>        | El visitante es identificado como cliente registrado de la tienda |

**Curso normal**

1. El usuario quiere identificarse en el sistema.
2. El sistema solicita el email del usuario y la contraseña.
3. El usuario introduce su email y contraseña.
4. El sistema valida los datos.
5. El sistema reconoce el usuario y recupera su información.

**Curso alternativo**

2. Si no existe la cuenta o la contraseña es incorrecta, se muestra un mensaje de error y se vuelve al punto 3.

|                             |  |
|-----------------------------|--|
| <b>Caso de uso</b>          | <b>Cerrar sesión</b>   |
| <b>Requisitos asociados</b> | Tienda – Registro de usuarios  |
| <b>Actor iniciador</b>      | Cliente  |
| <b>Descripción</b>          | El cliente cierra sesión en el sistema                                   |
| <b>Precondición</b>         | El cliente ha iniciado sesión  |
| <b>Postcondición</b>        | El usuario deja de ser identificado como cliente registrado de la tienda |

**Curso normal**

1. El usuario quiere cerrar sesión en el sistema
2. El sistema pide al usuario que confirme la acción.
3. El usuario confirma.
4. El sistema cierra la sesión del cliente.

**Curso alternativo**

3. Si el usuario no confirma la acción, finaliza el caso de uso.

|                             |   |
|-----------------------------|---|
| <b>Caso de uso</b>          | <b>Consulta Productos</b>                         |
| <b>Requisitos asociados</b> | Admin – Productos                                 |
| <b>Actor iniciador</b>      | Administrador                                     |
| <b>Descripción</b>          | Consulta de los datos de un producto              |
| <b>Precondición</b>         | El administrador ha iniciado sesión               |
| <b>Postcondición</b>        | El producto existe en el sistema                  |
|                             | Los datos del producto son mostrados por pantalla |
| <b>Curso normal</b>         |   |

1. El administrador quiere consultar los datos de un producto.
2. El sistema muestra todos los productos existentes.
3. El administrador selecciona el producto deseado.
4. El sistema muestra los datos del producto.

**Curso alternativo**

**Caso de uso**  
**Requisitos asociados**  
**Actor iniciador**  
**Descripción**

**Modificación Producto**  
Admin – Productos  
Administrador  
Modificación de los datos de un producto  
El administrador ha iniciado sesión  
El producto existe en el sistema  
Los datos del producto quedan actualizados en el sistema

**Precondición**

**Postcondición**

**Curso normal**

1. El administrador quiere modificar los datos de un producto.
2. El administrador modifica los datos (Nombre, Descripción, Precio, Términos del vocabulario, Imagen, Fabricante).
3. El sistema valida los nuevos datos.
4. El sistema registra los cambios introducidos en el producto.

**Curso alternativo**

4. Si alguno de los datos introducidos no es válido, se muestra un mensaje de error y se vuelve al punto 3.

**Caso de uso**  
**Requisitos asociados**  
**Actor iniciador**  
**Descripción**

**Agrupar producto**  
Admin – Agrupaciones  
Administrador  
Inserción de un producto dentro de una agrupación  
El administrador ha iniciado sesión  
El producto no pertenece a la agrupación  
Se registra la inserción del producto en la agrupación

**Precondición**

**Postcondición**

**Curso normal**

1. El administrador quiere añadir un producto en una agrupación.
2. El sistema muestra los productos que todavía no pertenecen a la agrupación.
3. El administrador selecciona el producto y le cambia el término de la categoría.
4. El sistema añade el producto en la nueva categoría y registra los cambios.

**Curso alternativo**

## 5. DESARROLLO DEL SISTEMA DE INFORMACIÓN

En este apartado de la memoria del proyecto explicaremos todo el proceso de desarrollo llevado a cabo para la creación del portal, correspondiente al prototipo para la aplicación planteada, analizada y diseñada para Gune Music.

Primeramente explicaremos como se realizó la instalación del gestor de contenidos Drupal en el equipo donde se ha desarrollado y probado el proyecto. Este equipo ha hecho funciones servidor, así como de cliente, a través de la interfaz *localhost*.

El entorno de desarrollo que se ha utilizado con el gestor de contenidos Drupal es basado en Apache, MySQL, PHP:

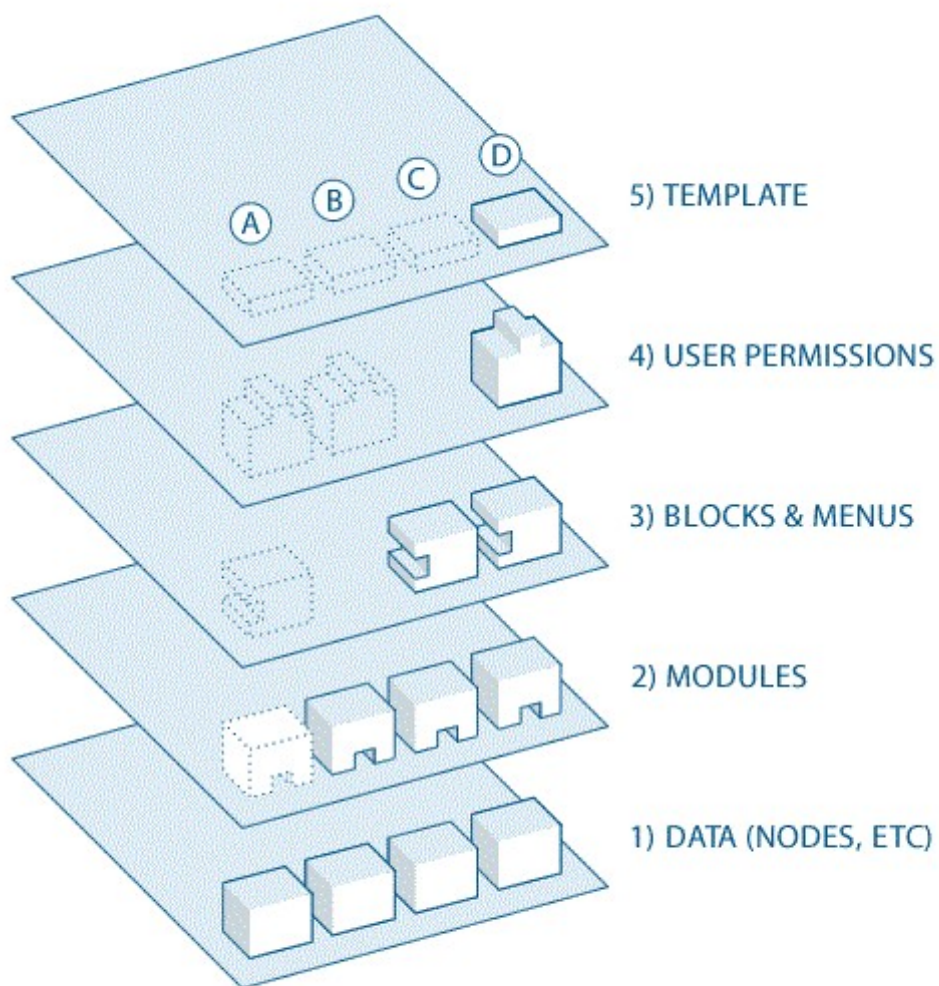
Apache Web Server v2.2.14  
PHP Script Language v5.3.1  
MySQL Database v5.1.41  
phpMyAdmin Database Manager v3.2.4

Los datos de configuración del servidor de aplicaciones y del servidor de base de datos son los siguientes:

Nombre del servidor: localhost  
e-mail del administrador: [mruizecheverria@gmail.com](mailto:mruizecheverria@gmail.com)  
Puerto HTTP de Apache: 80  
Base de datos: pfm\_basic  
Nombre de usuario del servidor MySQL: root  
Contraseña de administrador del servidor MySQL:

### **5.1. Arquitectura de Drupal. Estructuración de contenido**

Drupal estructura el contenido en una serie de elementos básicos: nodos , módulos, bloques y menús, permisos de usuario y plantillas.



<http://www.drupal.org>

### 5.1.1. Nodos

Los nodos son los elementos básicos de Drupal, en los que la información es almacenada. Cuanto más crece un sitio web, mayor es el número de nodos que lo componen. Podemos definir un nodo como todo contenido publicado dentro de un sitio Drupal, ya sea una página, una imagen, un producto, etc.

Por otra parte, existen diferentes tipos de nodo, por lo que varios nodos serán de tipo página, otros de tipo producto, otros de tipo video, etc.

Como se puede comprobar, a menudo las urls de los sitios web Drupal siguen el patrón [http://nom\\_servidor/node/num\\_nodo](http://nom_servidor/node/num_nodo), donde num\_nodo es el número de nodo asignado al contenido que estamos visitando.

### 5.1.2. Módulos

Los módulos operan sobre los nodos y le dan funcionalidad a Drupal permitiendo incrementar su funcionalidad o adaptarlas a la necesidad de cada sitio. Drupal separa, tanto a nivel de programación como a nivel de utilización, las distintas funcionalidades en diferentes módulos. De esta manera, cada módulo contiene por separado el código correspondiente a la acción o funcionalidad asociada.

El núcleo de Drupal contiene una serie de módulos que proporcionan al sistema las funcionalidades básicas para la configuración, gestión y creación de contenido. Además, en la página web de la comunidad de Drupal podemos encontrar una gran cantidad de módulos diseñados para añadir nuevas funciones y características a nuestro portal. La ventaja de estos módulos reside en que en muchas ocasiones satisfacen las funcionalidades que requerimos a nuestro portal y han sido probados por la propia comunidad, con lo que la probabilidad de que estos módulos contengan bugs y fallos se reduce sensiblemente.

Los nuevos módulos que vayamos instalando añadirán nuevos permisos al sistema, los cuales deberemos configurar para adaptar el portal al funcionamiento deseado. Para instalar nuevos módulos al gestor, basta con extraerlos en la carpeta *RUTA\_BASE/sites/all/modules*.

Posteriormente, en la página *Administrar -> Construcción del sitio -> Módulos*, podemos activar o desactivar todos los módulos disponibles, ya sean del núcleo del gestor o añadidos por nosotros.

Esta es otra de las ventajas más notables de los gestores de contenidos, ya que la integración de los módulos con el gestor, su instalación y activación es realmente sencilla y rápida.

### 5.1.3. Bloques y menús

Sobre la capa anterior está la que constituyen bloques y menús. Es a través de estos que podemos estructurar y organizar el contenido. A través de ellos podemos acceder o mostrar la información generada por los módulos a partir de la información almacenada en los nodos.

Drupal nos permite diferentes formas de crear contenido e información para nuestro portal. Varias de ellas se basan en la gestión de páginas, menús y bloques. A través de ellos podemos configurar el contenido que deseamos incluir, en qué lugar lo queremos ubicar y de qué manera se puede acceder a él.

Las páginas son la manera más cómoda, directa y práctica de incluir información en nuestro portal. En ellas podemos indicar un título para la página y un texto o código HTML donde escribiremos toda la información que deseamos mostrar.



El contenido de la página puede ser controlado, ya que Drupal nos permite elegir varios modos de interpretación de código HTML por razones de seguridad. Así pues, podemos elegir entre la interpretación de etiquetas básicas o interpretación de todas las etiquetas de HTML. Además, si activamos el módulo PHP Filter, podemos incluir código PHP directamente como contenido de la página.

Una forma que nos proporciona Drupal para controlar las navegaciones y los compartimentos se basa en los menús. Podemos crear diversos bloques de menús, donde cada menú es un enlace a una ruta asociada. Drupal nos ofrece tres bloques de menús:

- Menús primarios
- Menús secundarios
- Menús de navegación

Cada menú puede ser creado por diferentes causas. El propio gestor de contenidos proporciona varios menús para la gestión y visualización básica del portal. Así mismo, algunos de los módulos que vayamos añadiendo al gestor, generarán automáticamente nuevos menús. Por último, podemos crear nosotros mismos los menús que queramos y enlazarlos con la ruta correspondiente.

Los bloques son el último modo de gestionar el contenido en Drupal que vamos a comentar. Los bloques son contenedores que bien pueden mostrar información estática, contenido dinámico, un formulario de datos, etc. Al igual que los menús, el propio núcleo del gestor incluye una serie de bloques funcionales que podemos incluir en nuestro portal. Así mismo, los nuevos módulos que instalemos aportarán nuevos bloques.

Además, al igual que en el caso de las páginas, Drupal nos permite crear un bloque a partir un texto, código HTML o código PHP, según la interpretación que elijamos.

#### **5.1.4. Usuarios y permisos**

La siguiente capa es la de control de usuarios y permisos. Drupal dispone de un registro de usuarios y roles que permiten especificar qué tareas pueden realizar de las que ofrecen los módulos instalados en el sitio así como el contenido que se puede crear, al que se puede acceder o el que puede visualizar cada usuario o tipo de usuario (rol). Esto es lo mismo que decir que esta capa controla las operaciones (visualización, edición, creación, etc.) que un usuario puede realizar sobre elementos de las capas inferiores.

Una de las principales funcionalidades que nos aportan los gestores de contenidos es la gestión de permisos de usuarios. Esta es una de las principales ventajas, ya que nos permite configurar los permisos y privilegios de forma cómoda y práctica. Al igual que otros gestores, Drupal basa su gestión de permisos en roles, usuarios y permisos.

Así pues, existen diversos permisos relacionados con las acciones o funciones del gestor. Por defecto, existe una serie de permisos propios de los módulos del núcleo de Drupal. Así mismo, los nuevos módulos instalados crean generalmente nuevos permisos que debemos controlar.

Por otra parte, existen diferentes roles en el sistema. Varios de ellos son creados por defecto, mientras que si queremos podemos añadir nuevos roles. El núcleo de Drupal contiene inicialmente los roles Usuario anónimo y Usuario autenticado.

El primero de ellos es utilizado para configurar los permisos de los usuarios que acceden al portal sin haberse autenticado, mientras que el segundo rol es utilizado para configurar los permisos de cualquier usuario autenticado en el portal. Es importante reseñar que cualquier rol adicional creado por el administrador heredará los permisos asignados al rol Usuario autenticado.

Cada usuario puede estar asignado a uno o varios roles, mientras que nosotros, como administradores del sistema, podemos elegir qué permisos se asignan a cada rol. De esta manera, al asignar un rol a un usuario, le estamos asignando todos los permisos concedidos a dicho rol.

### **5.1.5. Plantillas**

La última capa, es la capa de las plantillas o templates es la que establece la apariencia o estilo de la información a mostrar al usuario. Esta separación entre información y aspecto permite cambiar la apariencia de un sitio web sin modificar el contenido.

### **5.1.6. Taxonomía**

Una de las características más potentes del gestor de contenidos Drupal es la taxonomía. Con ella podemos organizar el contenido de una forma muy eficiente. La taxonomía está basada en vocabularios, cada uno de los cuales puede contener términos.

Dentro de la taxonomía podemos crear diferentes vocabularios, referidos a distintas categorías. Dentro de cada vocabulario podemos crear diversos términos en relación a su categoría. Además, podemos crear relaciones entre términos.

De esta manera podemos clasificar si lo queremos el contenido creado según una categoría. Así, posteriormente podremos emplear esta clasificación para facilitar la búsqueda o el filtrado de nodos.

## **5.2 Instalación del servidor**

Primero, hemos instalado el gestor de contenidos Drupal. Para ello, hemos descargado gratuitamente el archivo correspondiente desde la página web [www.drupal.org](http://www.drupal.org).

Posteriormente hemos extraído el archivo dentro de la carpeta de de aplicaciones del servidor apache. En nuestro caso, la carpeta creada como resultado de la extracción tiene como nombre pfm\_basic.

Acto seguido, en un navegador, hemos introducido la dirección [http://localhost/pfm\\_basic/install](http://localhost/pfm_basic/install). De esta manera iniciamos el asistente instalador de Drupal, mediante el que instalamos y configuramos de forma sencilla el gestor de contenidos, indicando los datos correspondientes a la base de datos a utilizar.

Los principales datos a resaltar durante la instalación de Drupal son los siguientes:

Nombre del sitio web: Gune Music  
e-mail del sitio: [mruizecheverria@gmail.com](mailto:mruizecheverria@gmail.com)  
datos de la cuenta de administrador:  
nombre: drupal  
contraseña: jtarueta  
e-mail: [mruizecheverria@gmail.com](mailto:mruizecheverria@gmail.com)

A partir de aquí, todos los pasos y acciones descritas se realizan a través del navegador web, accediendo a la dirección [http://localhost/pfm\\_basic](http://localhost/pfm_basic), a través de la cuenta de administrador, la cual corresponde a un superusuario con permisos para realizar cualquier acción posible.

## 5.2.1 Módulos

Para empezar, hemos descargado todos los módulos que a raíz del estudio se han considerado necesarios para el desarrollo del portal. Los módulos descargados deben extraerse en la ubicación `RUTA_BASE_DRUPAL/sites/all/modules`. A continuación debemos activar los módulos en el menú *Administrar->Construcción del sitio->Módulos*, haciendo click en cada uno de los módulos que queremos activar.

A continuación comentamos cada uno de los principales y más importantes módulos activados para nuestro proyecto:

Módulo locale (localización): permite tener drupal en varios idiomas y elegir el idioma predefinido.

Menu per Role: permite restringir el acceso a menús según el rol del usuario.

Módulo node import: permite importar contenido desde ficheros de texto.

Módulo Taxonomy csv import/export: permite importar y exportar los vocabularios desde ficheros de texto.

Ubercart: proporciona una completa tienda virtual para la compra y venta de productos por medio de un catálogo, un carrito de compra, etc.

Módulo Views: permite crear vistas personalizadas diseñando el contenido a mostrar y la manera en que se muestra.

Módulo SWF Tools para incorporar una animación en la página de inicio (o en la que sea)

Custom search: permite personalizar y modificar la configuración y los parámetros del módulo de búsqueda básico incluido en el núcleo de Drupal.

### 5.2.2 Roles y permisos

Muchos de estos módulos añaden nuevos permisos que como administradores debemos controlar. En ocasiones nos interesa que los usuarios puedan realizar estas acciones, pero es importante que los permisos de administración y configuración de los módulos estén únicamente asignados al usuario administrador del sistema.

En nuestro caso y en esta primera fase, la empresa quiere que todos los usuarios sean visitantes, a excepción de los empleados de la tienda, que actuarán como administradores.

Como por defecto Drupal dispone de dos roles, el usuario anónimo y el usuario autenticado, hemos aprovechado esta circunstancia de manera que el rol usuario anónimo mantiene los permisos básicos de un usuario no autenticado.

El resto de permisos, los cuales consisten en la configuración y administración del portal y de los diferentes módulos, son únicamente poseídos por el usuario administrador.

### 5.2.3 Configuración tienda Ubercart

Para configurar la tienda Ubercart debemos navegar como administradores a *Administrar->Store administration->Configuration*. A partir de ahí, podemos seleccionar cada uno de los ámbitos de la tienda virtual y configurar cada parámetro.

En nuestro caso, la compra online no está contemplada en el desarrollo de este proyecto, por lo que las opciones de pay-pal, carrito de compra, etc no están habilitadas.

Accedemos a configurar los atributos básicos para la tienda:

Nombre de tienda: Gune Music

Propietario de tienda: Gune Music S.L.

Dirección de correo electrónico: [mruizecheverria@gmail.com](mailto:mruizecheverria@gmail.com)

Formato de datos (moneda, precio, peso, longitud, fecha)

### 5.2.4 Taxonomía

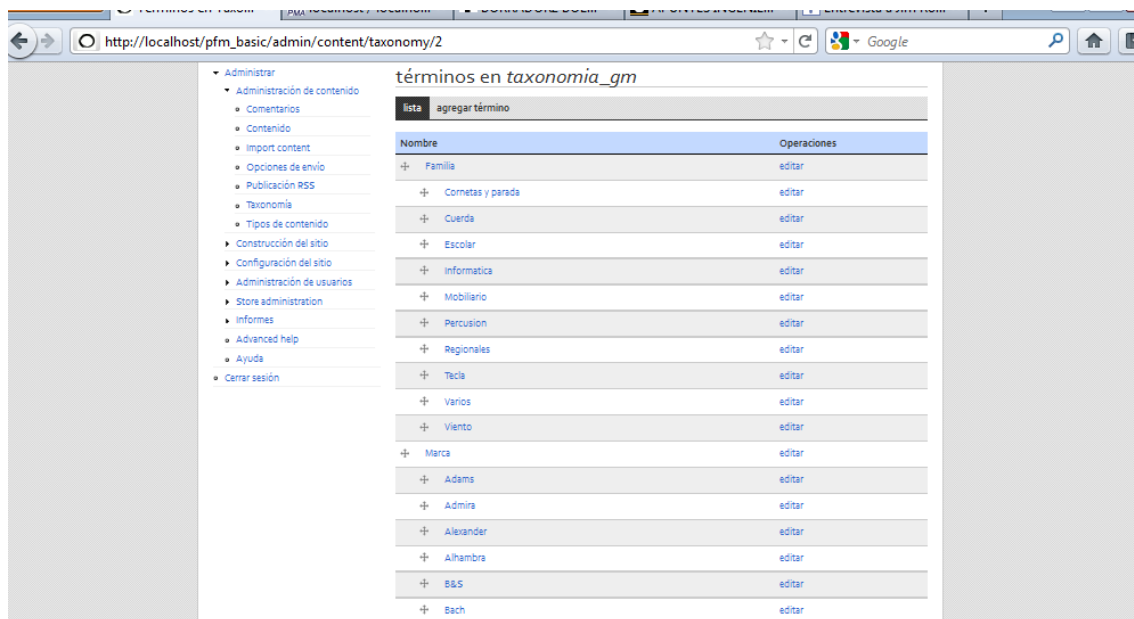
La taxonomía nos permite categorizar o clasificar los nodos o el contenido que hemos creado. En nuestro caso utilizaremos la taxonomía para clasificar los productos por categorías.

La activación del módulo de comercio Ubercart crea automáticamente un vocabulario *Catalog*, con el fin de que lo usemos para crear un catálogo de productos y facilitarnos posteriormente la visualización. Nosotros hemos definido el vocabulario *Taxonomia\_GM* con los términos que queremos contemplar en la clasificación por

vistas. Hemos utilizado el módulo Taxonomy CSV import/export para generar los terminos.

El vocabulario debe estar asociado a un tipo de nodo, de forma que al crear un nodo de dicho tipo podamos clasificarlo según los términos contenidos en el vocabulario. En nuestro caso, el vocabulario Taxonomia\_GM está asociado al tipo de nodo producto.

Tal y como se especificó en el análisis, las categorías serían creadas por el administrador y, en ningún caso, podrán ser creadas por los usuarios del portal. A continuación citamos cada una de las categorías creadas en el vocabulario Taxonomia\_GM.



Nos hemos basado en Taxonomía también para obtener los elementos de cada familia de instrumentos a través de vistas. De esta manera se han importado los vocabularios Instrumentos de viento, Instrumentos de cuerda, Accesorios viento, etc...

### 5.3 Vistas

Uno de los módulos usados en la construcción del portal web es el módulo VIEWS. Como se ha comentado, este módulo nos permite hacer una selección del contenido a mostrar y nos permite a su vez configurar la manera en que se va a ser mostrado dicho contenido.

VIEWS nos permite, mediante su interfaz gráfica, recuperar contenido de la base datos, seleccionando los campos a mostrar y los filtros de búsqueda a aplicar. Así mismo, nos permite elegir el formato de visualización deseado para mostrar los datos recuperados.

Para crear una búsqueda, debemos indicar la siguiente información para que la vista realice la consulta sobre la base de datos:

Fields: campos que deseamos seleccionar de un nodo, usuario, vocabulario, etc.

Filters: filtros que queremos aplicar a la búsqueda.

Arguments: la vista nos permite recibir argumentos para añadir al filtrado de contenido.

Sort criteria: criterio de ordenación de los resultados obtenidos tras la búsqueda.

En cuanto a la visualización del contenido recuperado, la vista nos permite configurar los siguientes aspectos:

Estilo para visualizar el contenido: en una tabla, en una lista, etc.

Por cada campo seleccionado, podemos editar la salida y redireccionar dicho campo enlazándolo a su propio nodo o a otra página.

Restringir la visualización de la vista en función de un rol.

Por último, la interfaz gráfica nos permite elegir el contenedor para visualizar la vista. Así pues, podemos, entre otras opciones, crear un bloque o una página, en cuyo caso indicaremos la url para acceder a ella.

### 5.3.1 Vistas de instrumentos por familias

Dentro del grupo instrumentos (de igual forma que ocurrirá con accesorios en una próxima descripción de su vista), tenemos diferentes familias (viento, cuerda, etc), a su vez compuestas por instrumentos ([trompeta, flauta,etc], [violin, violonchelo, etc],etc).

Para cada familia de instrumentos vamos a generar una vista (ej. instrumentos\_cuerda) que listará todas los instrumentos de esa familia (violin...).

Por cada instrumento (violín) mostraremos su nombre, el cual enlazará a una página donde se mostrará un listado con todos los productos violín agrupados por fabricante.

En este caso, utilizaremos otra vista (en nuestro caso le hemos llamado instrumentos y se describirá en el punto siguiente) a la que se le pasa como parámetro el tipo de instrumento (ej. instrumentos/violin).

Nombre de la vista: instrumentos\_cuerda

Tipo de vista: término

Campos: el término del vocabulario (cuerda). Importante completar Reescribir salida con vistas/instrumentos/[name]

Filtros: Taxonomía: vocabulario -> es igual a Instrumentos de cuerda

Criterio de ordenación ->Taxonomía: término Ascendente (orden alfabético)

Estilo: Tabla

Número de resultados a mostrar: 0 (ilimitado)

Contenedor

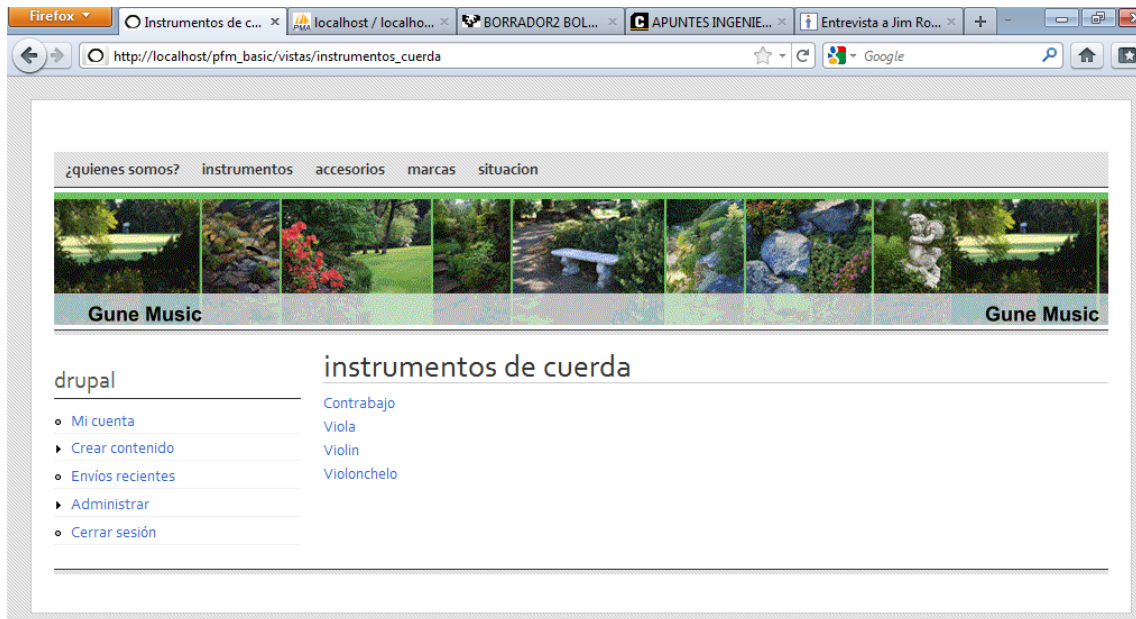
Página

Acceso: abierto

Ruta de página: vistas/instrumentos\_cuerda

Como vemos, hemos reescrito la salida del campo a mostrar (vistas/instrumentos/[name]), de modo que enlace con la página predefinida de instrumentos. En este caso, el código [name] se refiere al nombre del instrumento en cada caso.

Como hemos dicho, a partir de esta vista creamos una página, a la cual hemos asignado una ruta (vistas/instrumentos\_cuerda). Más adelante, indicaremos dicha ruta en la página de familias de instrumentos para poder navegar a estas vistas. Por el momento, la vista preliminar de la vista resulta como se observa en la imagen.



Al seleccionar alguna de los tipos de instrumentos mostrados, el portal nos redirige a la dirección *vistas/instrumentos/[name]*, donde [name] es el identificador del término seleccionado y *vistas/instrumentos* el path de la vista con paso de parámetro que describimos a continuación.

### 5.3.2 Vista de instrumentos con argumento

En este caso vamos a crear una vista con argumentos. A diferencia con los filtros usados en el caso anterior, el argumento pasado permite generar contenido dinámico en la vista en función del parámetro pasado.

Por ejemplo, en caso de seleccionar sobre la vista anterior la categoría violín, el portal nos redirige a la página correspondiente, sustituyendo el parámetro por el identificador del término violin.

Al configurar la vista hay que tener en cuenta que podemos definir, por ejemplo; el título de la página asociada en base al parámetro pasado. Para ellos en la casilla Título del campo argumento usaremos %1.

The screenshot shows the Drupal administration interface for editing a view named 'instrumentos'. The browser address bar indicates the URL is `http://localhost/pfm_basic/admin/build/views/edit/instrumentos`. A green message at the top states 'The view instrumentos has been saved.' The main content area is titled 'edit view instrumentos' and includes tabs for 'lista', 'add', 'editar', 'importar', and 'tools'. Below the tabs, it says 'View instrumentos, displaying items of type Nodo.' and provides options for 'Exportar', 'Clone', and 'View "Instrumentos"'. The configuration area is divided into several sections: 'View settings' (Description/Tag: None), 'Basic settings' (Name: Defaults, Title: instrumentos, Style: Tabla, etc.), 'Relationships' (None defined), 'Arguments' (Taxonomía: Term), 'Fields' (Content: marca Predeterminado, Node: Título Tipo / Serie, Node: Cuerpo Descripción de producto, Product: Sell price Precio, Node: Link Detalles), 'Sort criteria' (Content: marca asc, Node: Título asc), and 'Filters' (None defined). At the bottom, there are buttons for 'Guardar', 'Cancelar', and 'Eliminar', along with a note: 'Click on an item to edit that item's details.'

En ese caso, visualizaremos los productos categorizados como violines agrupados por sus fabricantes y a su vez por tipo o serie de violín (acústicos, eléctricos, serie silent, etc).



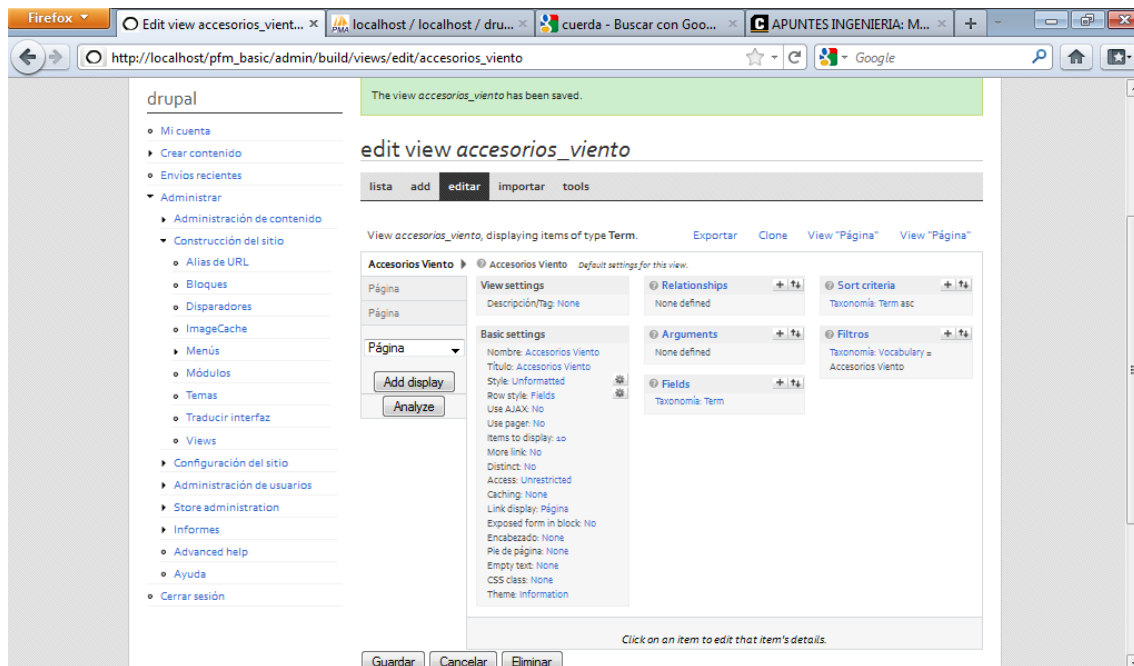
| drupal     |                             | violin  |         |                       |  |
|------------|-----------------------------|---|---------|-----------------------|--|
| Fabricante | Tipo/Serie                  | Descripcion de producto   | Precio  | Detalles              |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 4/4, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania.  | 199,00€ | <a href="#">vista</a> |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 3/4, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania.  | 199,00€ | <a href="#">vista</a> |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 1/2, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania.  | 194,00€ | <a href="#">vista</a> |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 1/4, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania.  | 180,00€ | <a href="#">vista</a> |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 1/8, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania.  | 180,00€ | <a href="#">vista</a> |  |
| Gluga      | Violin con Arco y Estuche   | Genial III, 1/16, Arco y abeto solido, perfil inovatado, guarnicion en ebano, barnizado a mano, arco nuez de ebano, estuche ligero forado tipo mochila. Resina pots, hecho a mano en Rumania. | 180,00€ | <a href="#">vista</a> |  |
| Stoner     | Violinas (solo instrumento) | Masina, 4/4, Fondo y arco de arce nizado, tapa de abeto, hecho y barnizado a mano, instrumento calibrado  | 397,00€ | <a href="#">vista</a> |  |
| Stoner     | Violinas (solo instrumento) | Elyria, 4/4, Fondo y arco de arce barnizado, tapa de abeto escogida, hecho y barnizado a mano, instrumento calibrado.   | 470,00€ | <a href="#">vista</a> |  |
| Stoner     | Violinas Electricas         | Studio II Electric, Transductor situado debajo del puente, controles de volumen y tono, arco nuez de ebano y estuche.   | 322,00€ | <a href="#">vista</a> |  |

Desde la última columna Detalles, se enlaza a la ficha del producto donde si hay imagen disponible podrá visualizarse.

### 5.3.3 Vistas de accesorios por familias

De igual manera a como hemos descrito para el grupo instrumentos, hemos procedido con los accesorios.

Para cada familia de accesorios vamos a generar una vista (ej. accesorios\_viento) que listará todas los accesorios asociados con instrumentos que pertenecen a la familia de instrumentos de viento.



Nombre de la vista: accesorios\_viento

Tipo de vista: término

Campos

Taxonomía: término

Reescribir salida: vistas/accesorios/[name]

Filtros

Taxonomía: vocabulario

Es igual a Accesorios Viento

Criterio de ordenación

Taxonomía: término

Ascendente (orden alfabético)

Estilo: Tabla

Número de resultados a mostrar: 0 (ilimitado)

Contenedor

Página

Acceso: abierto

Ruta de página: vistas/accesorios\_viento

Como vemos, hemos reescrito la salida del campo a mostrar (vistas/accesorios/[name]), de modo que enlace con la página predefinida de instrumentos. En este caso, el código [name] se refiere al nombre del instrumento en cada caso.

Por cada accesorio se muestra su nombre, a través del cual enlazaremos a una página donde se mostrará un listado con todos los accesorios de la misma categoría (categorizados por nombre en este caso) agrupados por fabricante.

Para ello utilizaremos otra vista (que en este caso hemos llamado accesorios) a la que se le pasa como parámetro el nombre del accesorio elegido.

Como vemos, hemos reescrito la salida del campo a mostrar (vistas/accesorios/[name]), de modo que enlace con la página predefinida de instrumentos. En este caso, el código [name] se refiere al nombre del instrumento en cada caso.

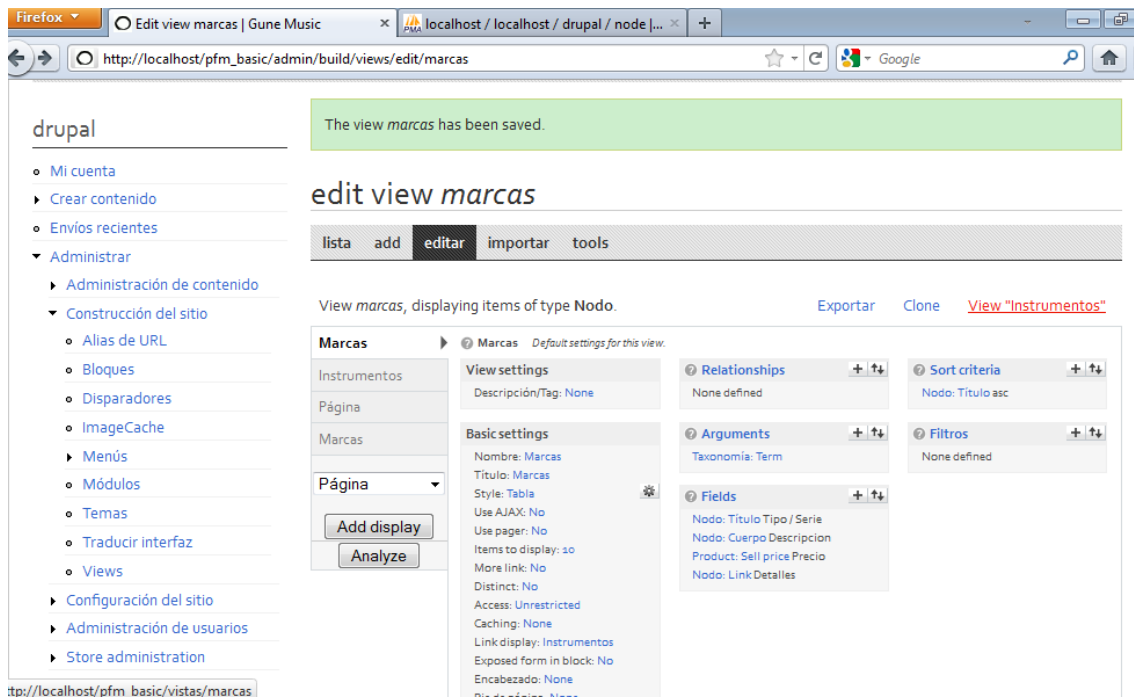
Como hemos dicho, a partir de esta vista creamos una página, a la cual hemos asignado una ruta (vistas/accesorios\_viento). Más adelante, indicaremos dicha ruta en la página de familias de instrumentos para poder navegar a esta vistas. Por el momento, la vista preliminar de la vista resulta como se observa en la imagen.



Al seleccionar alguna de los tipos de accesorios, el portal nos redirige a la dirección *vistas/accesorios/[name]*, donde [name] es el identificador del término seleccionado y *vistas/accesorios* es el path de la vista con paso de parámetro que describimos a continuación.

### 5.3.4 Vista de accesorios con argumento

En este caso el desarrollo de la vista conceptualmente es igual a la descrita en el apartado para la Vista de instrumentos con argumentos. Como ya sabemos, el argumento pasado permite generar contenido dinámico en la vista en función del parámetro pasado.



Mostramos a continuación el resultado de pasar como argumento limpieza a la vista descrita. (Hay que tener en cuenta que para el proceso de pruebas se ha cargado un subconjunto del total de productos). Aparecerán en la consulta tanto accesorios como instrumentos agrupados por Tipo/Serie.

### 5.3.5 Vista de productos por marca

Esta es una vista similar a las que hemos llamado instrumentos y accesorios. Le pasaremos como parámetro el nombre comercial del fabricante y nos muestra todos los productos de ese fabricante agrupados por tipo o serie. Para no introducir información redundante en este caso se suprime el campo marca puesto que ya sabemos el fabricante que vamos a consultar (si se quiere en detalles, obtendremos la ficha completa del producto).

Si interesara, y puede que sea una buena modificación, se puede añadir una agrupación más por tipo de producto que incluya el tipo de producto (instrumento o accesorio). En este caso habrá que incluir un campo nuevo (como se hizo con marca) al tipo de contenido producto.

¿quienes somos? instrumentos accesorios marcas ofertas situacion contacta

**Gune Music** **Gune Music**

inicio de sesión

Nombre de usuario: \*

Contraseña: \*

Iniciar sesión

Entrar usando OpenID

Crear nueva cuenta

Solicitar una nueva contraseña

**adams**

| Tipo / Serie                        | Descripción  | Precio    | Detalles              |
|-------------------------------------|--|-----------|-----------------------|
| Escobillones y cepillos de limpieza | Cepillo para limpiar boquillas de Metal o de Ebonita | 6,00€     | <a href="#">vista</a> |
| Escobillones y cepillos de limpieza | Escobill   | 8,00€     | <a href="#">vista</a> |
| Kits de limpiezas y utensilios      | Kit mantenimiento Trompeta Stradivarius.             | 30,00€    | <a href="#">vista</a> |
| Trompeta profesional                | Trompeta Stradivarius                                | 2.431,00€ | <a href="#">vista</a> |

### 5.3.6 Vistas de productos en oferta

Para terminar con las vistas, creamos una vista que filtre por el término de vocabulario oferta. Si su valor es Sí mostrará el producto en el listado.

¿quienes somos? instrumentos accesorios marcas ofertas situacion contacta

**Gune Music** **Gune Music**

inicio de sesión

Nombre de usuario: \*

Contraseña: \*

Iniciar sesión

Entrar usando OpenID

Crear nueva cuenta

Solicitar una nueva contraseña

**ofertas**

| Fabricante | Tipo / Serie                        | Descripción   | Precio  | Detalles              |
|------------|-------------------------------------|---|---------|-----------------------|
| Stentor    | Violines (solo instrumento)         | Elysia, 4/4, Fondo y aros de arce bellamente rizado, tapa de abeto escogida, hecho y barnizado a mano, instrumento calibrado. | 470,00€ | <a href="#">vista</a> |
| Adams      | Escobillones y cepillos de limpieza | Cepillo para limpiar boquillas de Metal o de Ebonita  | 6,00€   | <a href="#">vista</a> |
| Adams      | Escobillones y cepillos de limpieza | Escobill  | 8,00€   | <a href="#">vista</a> |
| Adams      | Kits de limpiezas y                 | Kit mantenimiento Trompeta Stradivarius.  | 30,00€  | <a href="#">vista</a> |

## 5.6 Estructura: páginas y menús

Hemos llegado a un momento en el cual hemos activado y configurado varios de los módulos más importantes del portal y hemos creado varias páginas para el filtrado y visualización de instrumentos, accesorios, marcas y ofertas.

Sin embargo, no disponemos de menús y enlaces que nos permitan navegar y acceder a dicho contenido. Por esta razón, nos disponemos a continuación a crear las páginas necesarias para ubicar todo el contenido y los menús y enlaces correspondientes para acceder a dicho contenido.

### 5.6.1 Página de inicio

Comenzaremos por crear una página principal. Como administradores crearemos un nuevo nodo de tipo página, con las siguientes características:

Título: inicio

Formato de entrada: PHP Code

Cuerpo:

```
<?php
echo "<div align='center'>";
echo "<a href='node/6' style='text-decoration:underline'>
<font class='extra'><strong>Entrar</strong></font></a><br><br><br>";
print swf('video.swf');
echo "<font class='small'>Gune Music ©2011.
<a href='mailto:tienda@gunemusic.com'>tienda@gunemusic.com</a></font>
<br> C/ Rafael Alberti, 22 - 01010 - Vitoria - Teléfono: 945 197887 <br></div>";
?>
```

Hemos creado una página con título *Inicio*, hemos insertado código PHP para colocar una animación en flash (que se interpreta gracias a que el módulo SWF Tools está activado) y un link para Entrar.

Finalmente hemos colocado esta página como página de inicio del portal: revisamos la configuración del sitio web, donde debemos indicar la url de la página que queremos hacer principal. De esta forma, al acceder al portal web, el gestor nos muestra por defecto esta página.

### 5.6.2 Páginas: Quienes somos, Situación, Contacto

Creamos tres páginas estáticas con información de la tienda. Para las tres definimos el título, el cuerpo (insertamos una imagen en el caso de Situación) y asociamos a un item de menú de primary links estableciendo su peso u orden (1 6 y 7) en cada caso.

## Quienes somos

¿quienes somos? instrumentos accesorios marcas situacion contacta

**Gune Music** **Gune Music**

inicio de sesión

Nombre de usuario: \*

Contraseña: \*

Iniciar sesión

Entrar usando OpenID

Crear nueva cuenta

Solicitar una nueva contraseña

¿quienes somos?

La empresa Gune Music S.L. se dedica a la distribución, venta y mantenimiento de equipamiento e instrumentos musicales ofreciendo un servicio profesional y experimentado a empresas, instituciones, particulares, orquestas y grupos musicales.

En [www.gunemusic.com](http://www.gunemusic.com) encontrarás todo tipo de instrumentos así como accesorios y libros. También realizamos alquiler, afinaciones y transporte de pianos. Para nosotros es importante la venta pero también el mantenimiento y la posible reparación de los instrumentos musicales.

[http://localhost/pfm\\_basic/node/6](http://localhost/pfm_basic/node/6)

## Situación

¿quienes somos? instrumentos accesorios marcas **situacion** contacta

**Gune Music** **Gune Music**

inicio de sesión

Nombre de usuario: \*

Contraseña: \*

Iniciar sesión

Entrar usando OpenID

Crear nueva cuenta

Solicitar una nueva contraseña

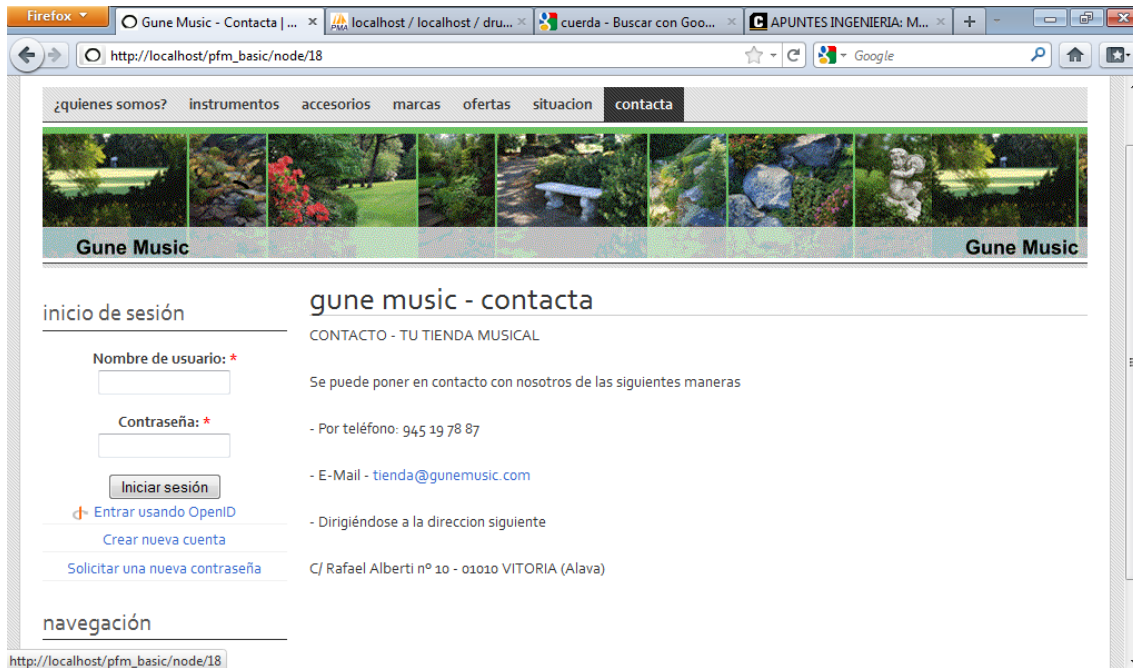
navegación

- Envíos recientes

situacion

GuneMusic.es, [contacto@gunemusic.com](mailto:contacto@gunemusic.com)  
© Rafael Alberti, 21 - 01000 - Vitoria - Teléfono: 943 497887

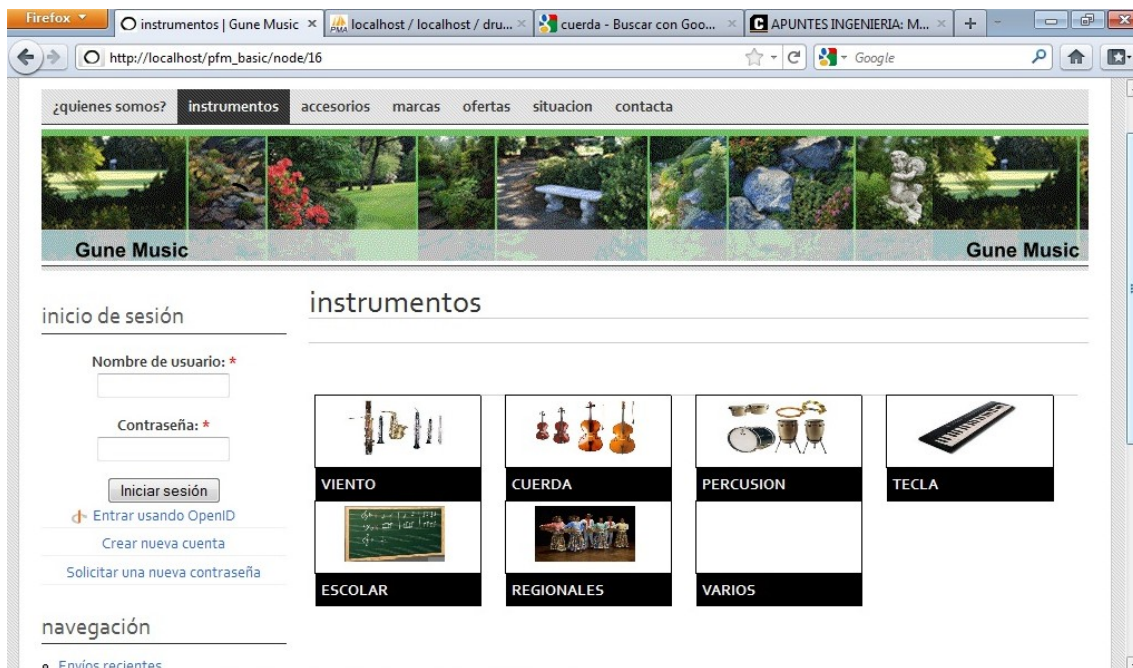
## Contacta



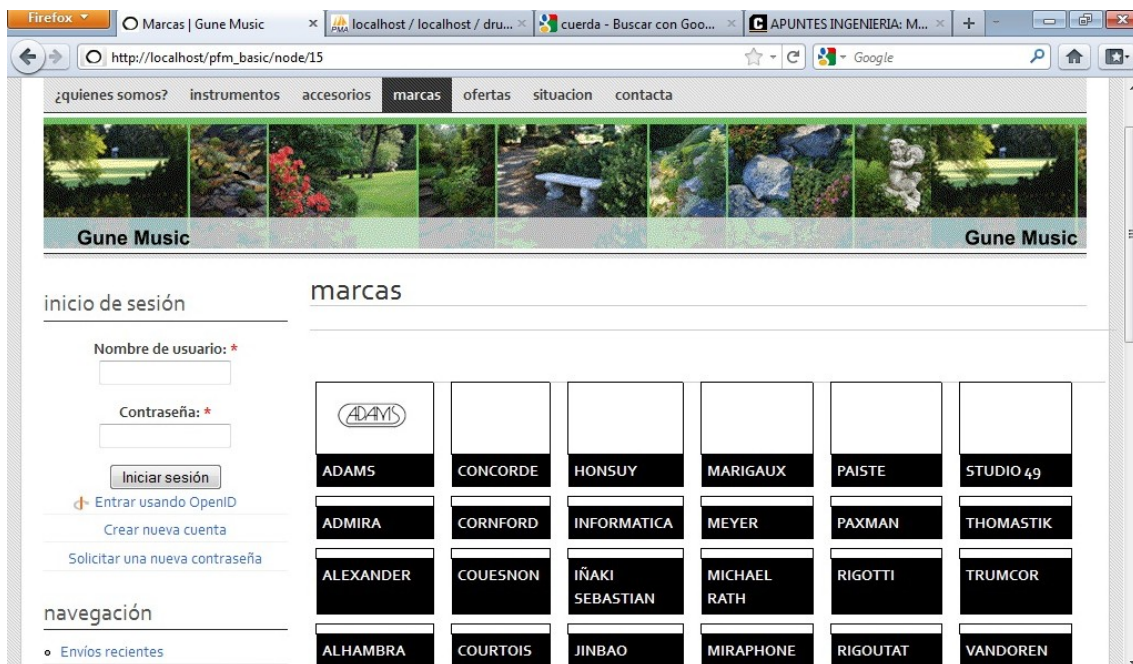
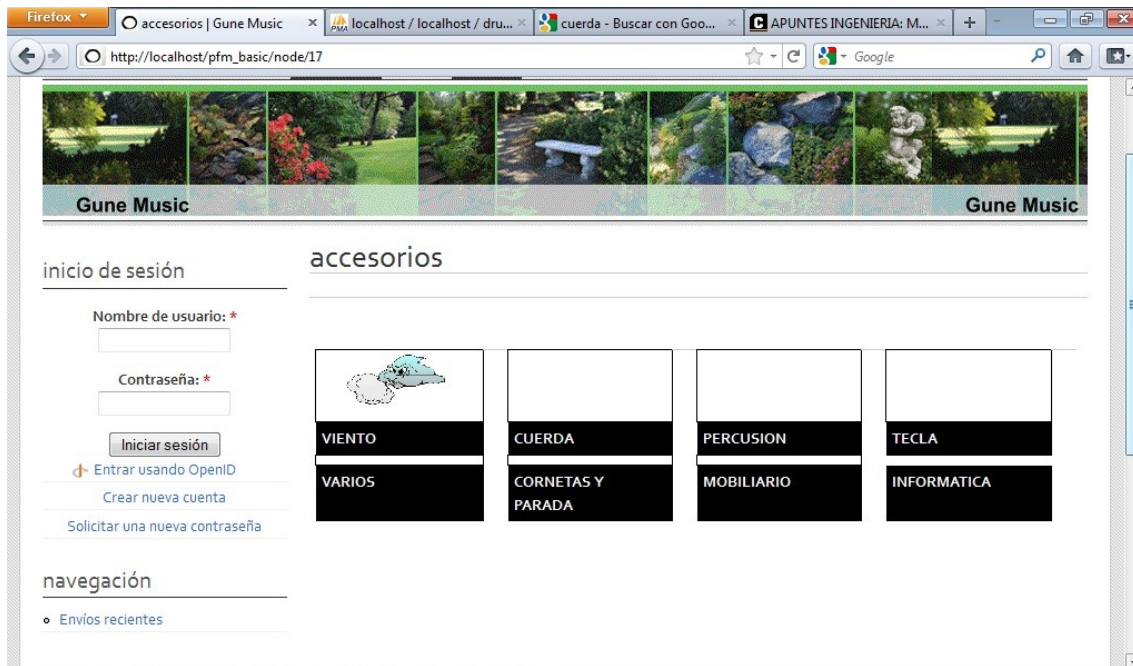
### 5.6.3 Páginas: Instrumentos, Accesorios, Marcas y Ofertas

En el caso de Instrumentos, Accesorios, Marcas y Ofertas tenemos la posibilidad de sacar mediante una vista las agrupaciones de cada apartado en función de los términos de catálogo.

Tenemos una segunda posibilidad de crear el diseño a medida mediante tablas en html. Optamos por esta posibilidad excepto en el caso de ofertas. Aunque es más vistoso, la edición es más compleja y por ello en el momento de redactar este documento están hay iconos sin completar.







Una vez en estas páginas, cada elemento que se muestra enlazará en cada caso a las vistas descritas en los apartados anteriores (vistas con filtro, con argumentos o combinaciones y sucesiones de ambas).

## 5.6.4 Menús

Como se ha ido viendo, hemos creado páginas para ir enlazando con la información y las vistas que nos interesaban.

Se ha podido acceder desde las pestañas de menú creadas en la cabecera. Para cada uno de los casos hemos creado un menú que hemos enlazado con la ruta url asociada a las páginas en su momento:

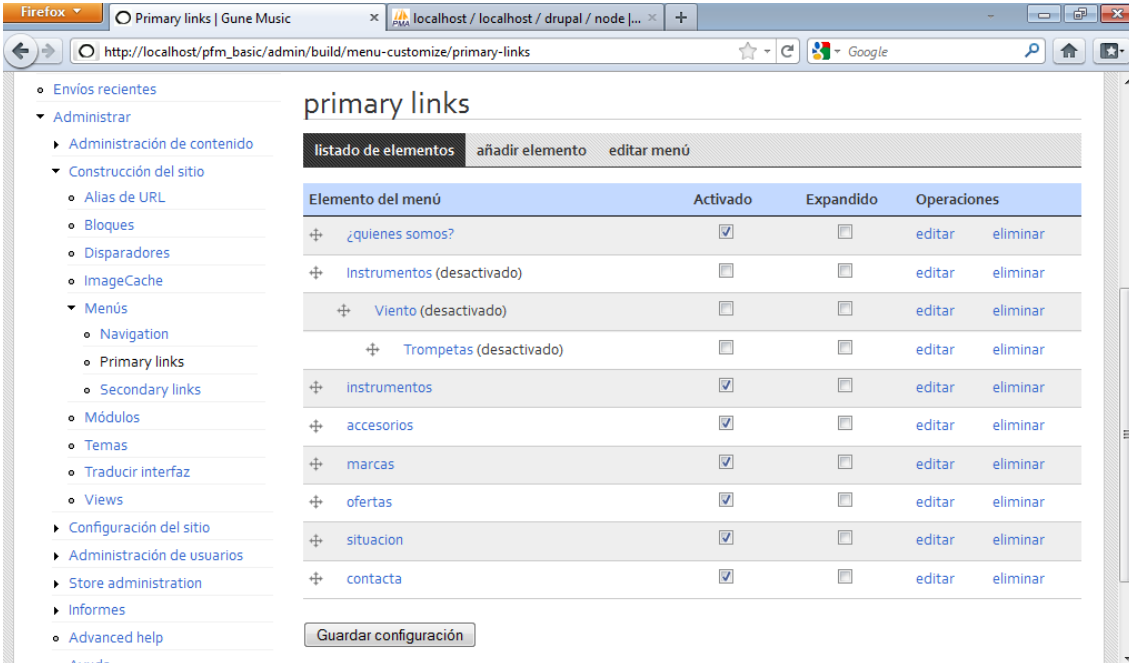
Título del menú: ¿quienes somos?

Descripción:

Ruta: /pfm\_basic/node/6

Elemento padre: enlaces primarios

Todos los menús han sido ubicados en los enlaces primarios que corresponden en nuestro portal al menú superior.



| Elemento del menú            | Activado                            | Expandido                | Operaciones     |
|------------------------------|-------------------------------------|--------------------------|-----------------|
| + ¿quienes somos?            | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + Instrumentos (desactivado) | <input type="checkbox"/>            | <input type="checkbox"/> | editar eliminar |
| + Viento (desactivado)       | <input type="checkbox"/>            | <input type="checkbox"/> | editar eliminar |
| + Trompetas (desactivado)    | <input type="checkbox"/>            | <input type="checkbox"/> | editar eliminar |
| + instrumentos               | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + accesorios                 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + marcas                     | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + ofertas                    | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + situacion                  | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |
| + contacta                   | <input checked="" type="checkbox"/> | <input type="checkbox"/> | editar eliminar |

## 5.7. Carga de productos

De marea similar a como se hicieron las imports de los terminos de los vocabularios , pero haciendo uso del módulo node\_import realizamos la carga inicial de productos de prueba.

Editamos en una hoja de cálculo los productos que vamos a incorporar. UID, Nombre, Descripción, Catálogo, Precio, Formato, Marca e Imagen serán las columnas que vamos a completar.

| ID | Nombre         | Descripción           | Catalogo     | Precio  | Formato | Marca     | Imagen  |
|----|----------------|-----------------------|--------------|---------|---------|-----------|---|
| 1  | Violin con A   | Genial III, 4/4       | Cuerda, Viol | 199,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 2  | Violin con A   | Genial III, 3/4       | Cuerda, Viol | 199,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 3  | Violin con A   | Genial III, 1/4       | Cuerda, Viol | 194,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 4  | Violin con A   | Genial III, 1/4       | Cuerda, Viol | 180,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 5  | Violin con A   | Genial III, 1/4       | Cuerda, Viol | 180,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 6  | Violin con A   | Genial III, 1/4       | Cuerda, Viol | 180,00  |         | 2 Gliga   | /pfm_basic/sites/default/file/images/student2-p.jpg |
| 7  | Violines (sol) | Messina, 4/4          | Cuerda, Viol | 397,00  |         | 2 Stentor | /pfm_basic/sites/default/file/images/messina-p.jpg  |
| 8  | Violines (sol) | Elysia, 4/4           | Cuerda, Viol | 470,00  |         | 2 Stentor | /pfm_basic/sites/default/file/images/messina-p.jpg  |
| 9  | Violines Ele   | estuche.              | Cuerda, Viol | 220,00  |         | 2 Stentor |   |
| 10 | Serie Silent   | negro, rojo y marron. | Cuerda, Viol | 721,00  |         | 2 Yamaha  |   |
| 11 | Serie Silent   | estuche y arco.       | Cuerda, Viol | 1004,00 |         | 2 Yamaha  |   |
| 12 | Serie Silent   | VS C1: Fun            | Cuerda, Viol | 72,00   |         | 2 Yamaha  |   |
| 13 | Escobillone    | Ebonita               | Viento, Limp | 6,00    |         | 2 Adams   |   |
| 14 | Escobillone    | plastificado          | Viento, Limp | 8,00    |         | 2 Adams   |   |
| 15 | utensilios     | Stradivarius          | Viento, Limp | 30,07   |         | 2 Adams   |   |
| 16 | Trompeta pr    | Trompeta S            | Viento, Tron | 2431,00 |         | 2 Adams   |   |

Guardamos el contenido en un fichero de texto separado por ; y con delimitador de texto “.

Iniciamos el proceso de import que se describe en el video *carga\_masiva.avi* adjunto a la memoria.

Destacar que al Tipo de Contenido Producto ha habido que añadir el campo *marca* para poder agrupar las vistas según este criterio.

### 5.7. Plantillas

Para adaptar el diseño que por defecto trae drupal al formato que nos ha presentado Gune Music. S.L., hemos optado por hacer uso de un thema ya creado con el nombre nitobe.

Una vez cargado se ha activado y configurado para que esconda y ofrezca los elementos deseados.

La cabecera se ha sustituido por la imagen que ha querido GuneMusic S.L.

## 6. CONCLUSIONES

Actualmente el núcleo del sistema está implementado casi en su totalidad: el sistema puede entregarse y ponerse a prueba. Una vez depurado y cargado con la totalidad de sus contenidos, daría soporte a las funciones de acceso a información, consultas de catálogo y tareas de administración.

Quedan pendientes detalles de funcionamiento como puede ser la ficha de detalles del producto. Hemos utilizado la vista que trae la plataforma por defecto y es posible que el diseño mejore mediante una plantilla creada expresamente para nuestros productos.

Otras mejoras que pueden abordarse son las siguientes:

- Incluir un nuevo campo (tipo) en el tipo de contenido Producto y cargar el valor del campo (accesorio, instrumento) a cada producto con el fin de poder separar en la vista de marcas los productos entre instrumentos y accesorios.

- Desarrollar una condición en la plantilla de diseño para que en caso de que el usuario sea anónimo, no esté visible el portlet de validación de la izquierda de manera que se aproveche la sección central al máximo en estos casos. Se podría acceder al portlet que se oculta desde alguna pestaña del menú de cabecera (login o similar) y una vez validados empezaría a ser visible el menú de navegación para los usuarios registrados a fin de que puedan modificarse los contenidos del portal.

- De acuerdo con la empresa, será conveniente definir un modelo de búsqueda de productos en base a los términos que se decidan.

En conjunto mi valoración del proyecto es positiva.

He conseguido completar un piloto que se puede comenzar a probar. Con las pruebas de validación es normal que aparezcan más detalles para pulir y ya se han mencionado algunas mejoras que podrían tenerse en cuenta. En cualquier caso, los requisitos básicos están cubiertos y es suficiente para empezar.

Durante la elaboración de proyecto he tenido oportunidad de aprender como funciona el gestor de contenidos Drupal (había trabajado con Plone y Zope anteriormente) que considero muy versátil para el desarrollo de portales de este tipo. Por ello, es muy posible que lo adopte para dar a conocer otros elementos que a nivel personal me gustaría mostrar (cuadros de pintura).

La instalación no es costosa y la configuración de sus módulos es sencilla y en general agradecida puesto que con poco esfuerzo se consiguen funcionalidades que mediante programación propia supondría mucho esfuerzo desarrollar. Por ejemplo el proceso de login/logout, el cambio de apariencia mediante plantillas son aspectos que tenemos ya implementados nada más terminar de instalar Drupal.

También ha resultado muy sencilla la importación masiva de productos, una tarea compleja de implementar de otra manera y que a través del módulo node import se consigue de manera asombrosamente fácil.

Lo mismo podemos decir de la carga de las imágenes que asociamos a los productos. Con la configuración de los módulos habituales, el problema ya viene solucionado.

Sin embargo, la creación de elementos específicos muy concretos exige un conocimiento de Drupal más profundo que lleva un tiempo de estudio y que no es tan ventajoso respecto a un desarrollo a medida. Es el caso de la creación de búsquedas no estándar que suponen atacar la base de datos mediante consultas sql ejecutadas desde php.

También la creación de menús con desplegados verticales sobre los que ir eligiendo items que vayan abriendo a su vez desplegados de menú, requieren encontrar del módulo que permita implementarlos (no siempre con los resultados de diseño deseados, como ha ocurrido) o la programación específica y compatible con las plantillas Drupal instaladas.

Aunque al comienzo, una instalación sencilla abre un enorme abanico de posibilidades para trabajar, hay puntos como los mencionados en el caso de las búsquedas y menús, que exigen para su realización un tiempo y conocimiento avanzados ya que no se pueden abordar de manera inmediata a través de las herramientas estándar de Drupal.

En nuestro caso, el problema de los menús se han atajado mediante páginas intermedias entre las vistas y los items del menú de cabecera. Se trata de una solución costosa en el sentido de que implica la creación de código html, pero más rápida que programar un nuevo bloque de menú.

Este punto de los menús junto con la definición de los catálogos para clasificar los productos, han sido los que más tiempo y esfuerzo han supuesto en el diseño y su implantación.

En cuanto a las vistas, Drupal nos ha permitido listar los productos filtrados tal como era requisito de los productos.

La información en la red es muy abundante y clara sobre drupal en general y sobre el módulo de las vistas en particular. A través de esta información, hemos conseguido configurar vistas con filtros y pasos de parámetros.

Afortunadamente las vistas diseñadas como están, han permitido adaptarlas a nuestros requisitos. Si esto no hubiera sido así, habríamos necesitado código php y sql para definir las de acuerdo a nuestras necesidades.

En resumen, podemos decir respecto al esfuerzo de aprendizaje de drupal, que inicialmente es sencillo adaptar sus herramientas para la mayoría de los requisitos, pero que en el caso de necesidades menos genéricas es necesario profundizar en la documentación, hacer muchas pruebas que no dan resultado y tener que programar de cero en el peor de los casos-

La documentación y el soporte de Drupal lo considero muy bueno. Se nota que es un proyecto activo en el que hay trabajando mucha gente. Esto permite que se documenten ingentes cantidades de artículos e incluso videotutoriales muy completos.

Como hemos visto, el proyecto no se ha terminado completamente. Las últimas fases del desarrollo y el estudio de una forma estructurar los catálogos para conseguir el resultado requerido han necesitado más del tiempo esperado.

Sobre todo, el proyecto se ha retrasado con el estudio de alternativas de solución. El desconocimiento de Drupal junto con el miedo a no encontrar las salidas necesarias para las necesidades marcadas eran aspectos favorables a elegir una solución a medida basada en la implementación a medida.

Después de la corrección de la segunda práctica, tomó fuerza la alternativa de Drupal por todas las razones explicadas en el Estudio de Viabilidad de este documento. Afortunadamente, la elección ha sido adecuada porque hay una cantidad de trabajo ya realizado previamente por la comunidad que de haber tenido que desarrollar desde cero no habría permitido la entrega del proyecto en el tiempo previsto.

Aun con todo, esta decisión y el estudio posterior de la herramienta Drupal, sobre la que el autor de este proyecto no tenía conocimientos previos, han dilatado el plazo de ejecución del proyecto hasta el punto de hacer peligrar la entrega del piloto en el plazo establecido.

A nivel personal, al margen de que tecnológicamente es la alternativa acertada, ha resultado más enriquecedor el estudio de las posibilidades de Drupal y sus módulos, su configuración y adaptaciones que la programación de múltiples pantallas basadas en php y mysql con sentencias repetitivas y bucles interminables.

Respecto a las fases del proyecto, me he sentido más a gusto trabajando sobre la máquina que sobre el papel, en las últimas fases del proyecto que en las primeras.

Visto el resultado final y analizando la importancia de haber re-estudiado las alternativas y el impacto que tiene sobre el proyecto (de poderse entregar a no poderse entregar) la elección de la mejor alternativa, queda claro la importancia de haber trabajado al detalle en cada una de las fases, independientemente de la afinidad que uno sienta con cada una de ellas.

Como alumno, haber seguido el proyecto completamente en la totalidad de sus fases ha sido muy interesante. Además dadas las circunstancias he podido ver en la práctica el valor de cada una de ellas en el resultado final. Como hemos visto, en nuestro caso una mala realización del estudio de viabilidad, podría haber dado al traste con el proyecto.

En general, como colofón del Master del Software Libre, el Trabajo Fin de Master es adecuado. Es un repaso muy bueno sobre todo de la asignatura de Implantación de Sistemas de Software Libre, tiene aspectos básicos de las asignaturas de Desarrollo y Bases de Datos. Caso de haber tenido tiempo de alojar el piloto en una máquina de producción, habríamos tenido ocasión de trabajar los conceptos de las asignaturas de Administración Sistemas Operativos y Seguridad.

Respecto al Máster, las expectativas se han visto cubiertas por parte de este alumno. Han sido asignaturas muy densas y bien llevadas y entretenidas a mi juicio, las correspondientes a Administración de S.O. Linux Y Seguridad Avanzada.

He aprendido bastante de Bases de Datos por el desconocimiento previo y el nivel de exigencia de las prácticas.

Las asignaturas de Introducción de Software Libre e Implantación de Sistemas resultaron en su momento algo teóricas aunque en el caso de Implantación han encontrado su desarrollo práctico con este proyecto.

En el otro extremo, la asignatura de Herramientas de Software Libre apenas a aportado nuevos conocimientos y su elección como optativa tal vez no fuera del todo acertada. Sin embargo, a la hora de elaborar la documentación de este proyecto, el conocimiento del editor de textos de OpenOffice así como el conocimiento del programa de presentaciones ha sido bastante útil.