



**Proyecto Fin de Carrera**

**SISTEMA DE GESTION Y CONSULTA  
DOCUMENTAL BASADO EN ETIQUETADO  
INTELIGENTE**

**(DOCUMENT MANAGEMENT SYSTEM  
BASED ON SMART LABELING  
CONSULTATION)**

Para acceder al Título de

**INGENIERO EN INFORMÁTICA**

**Autor: Miguel Cortés Tazón**

**Septiembre – 2014**

## **Agradecimientos**

A mi familia, por haberme apoyado y ayudado en todo momento y en los momentos futuros.

A mi director de proyecto Alberto Eloy García y a su hermana Nuria García por toda la ayuda, la dedicación y el interés que me han prestado.

A mis amigos, los cuales nunca se han ahorrado una palabra de ánimo, y por estar en todo momento ahí.

Mil Gracias.

## Resumen

Hasta ahora el Área de Exposiciones de la Universidad de Cantabria ha mostrado la información de las obras que se exponen en el Paraninfo de la Universidad a través de una web creada en texto plano con únicamente enlaces a documentos.

Este documento presenta la solución desarrollada a partir de herramientas de código abierto, mediante la cual se permite gestionar a los administradores de la Exposición todos los contenidos asociados para su consulta de forma visual, esto es sencilla y amena de cara al usuario final.

La aplicación provee una interfaz visual y de fácil manejo, especialmente pensada para el usuario que gestione (y administre) toda la información, pudiendo controlar todos los aspectos de la misma sin tener prácticamente conocimientos de programación web.

La accesibilidad a toda la información está asegurada para el uso de cualquier dispositivo, con el único requerimiento de incluir una conexión de datos (Internet) y un navegador (Explorer, Chrome, Safari...). Además, la aplicación provee un mecanismo basado en etiquetas inteligentes mediante las cuales el usuario es capaz de acceder a la información correspondiente a las obras expuestas in situ, esto es, directamente en la Sala de Exposiciones y delante de la obra en cuestión.

## **Abstract**

So far the “Área of Exposiciones” of the University of Cantabria has shown the information about the artworks through a website developed in plain text with only links to documents.

This paper presents the developed solution, using open source tools which the exposition administrators can manage all the contents associated to the exhibitions in a visually way, simple and enjoyable for the end user.

The application provides a specially design for the user to manage all the information and control all aspects without any kind of web programming skills with a user-friendly visual interface.

The accessibility of all information is guaranteed towards the use of any device, with the only requirement to have a data connection (Internet) and a browser (Explorer, Chrome, Safari ...). In addition, the application provides based smart tags as the elected mechanism allowing to the user to access the information on the nearest exhibited artwork.

# Índice

<b>1. Introducción.....</b>	<b>5</b>
<b>1.1 Motivación y descripción del problema.....</b>	<b>5</b>
<b>1.2 Objetivos.....</b>	<b>6</b>
<b>1.3 Estructura de la memoria.....</b>	<b>6</b>
<b>2. Análisis de requisitos.....</b>	<b>7</b>
<b>2.1 Metodología del desarrollo.....</b>	<b>8</b>
<b>2.2 Requisitos funcionales.....</b>	<b>9</b>
<b>2.3 Casos de uso de los actores.....</b>	<b>10</b>
<b>2.4 Requisitos no funcionales.....</b>	<b>13</b>
<b>2.4.1 Tecnologías.....</b>	<b>15</b>
<b>2.4.2 Herramientas.....</b>	<b>17</b>
<b>2.4.3 Herramientas CMS.....</b>	<b>18</b>
<b>3. Diseño.....</b>	<b>21</b>
<b>3.1 Fases del diseño.....</b>	<b>22</b>
<b>3.2 Diseño del gestor de contenidos.....</b>	<b>22</b>
<b>4. Implementación.....</b>	<b>26</b>
<b>4.1 Taxonomías.....</b>	<b>26</b>
<b>4.2 Tipos de contenido.....</b>	<b>30</b>
<b>4.2.1 Añadir obras.....</b>	<b>33</b>
<b>4.2.2 Añadir visita virtual.....</b>	<b>38</b>
<b>4.3 Views.....</b>	<b>49</b>
<b>4.3.1 Detalle de las vistas desarrolladas.....</b>	<b>50</b>
<b>4.3.2 Programas del gestor de contenidos que utilizan las vistas.....</b>	<b>53</b>
<b>4.3.3 Otros elementos del gestor de contenidos.....</b>	<b>57</b>
<b>5. Conclusiones y trabajos futuros.....</b>	<b>62</b>
<b>5.1 Conclusiones.....</b>	<b>62</b>
<b>5.2 Trabajos futuros.....</b>	<b>63</b>
<b>6. Bibliografía.....</b>	<b>64</b>

## 1. Introducción.

En este capítulo se detallarán los motivos que llevaron a realizar el proyecto, la descripción del problema inicial, su propuesta de resolución y los objetivos que se pretenden alcanzar con su realización. Además, se comentará la estructura del presente documento.

### 1.1 Motivación y descripción del problema.

La sociedad está completamente digitalizada y las TIC se encuentran a la orden de día. Antes de ir a cualquier lugar buscas información referida al sitio para saber si merecerá o no la pena ir.

En este caso, en la Sala de Exposiciones de la Universidad de Cantabria, desea ofrecer, una aplicación moderna y de fácil uso que sea capaz de categorizar y mostrar todo el contenido de la sala de exposiciones.

Actualmente en la Figura 1 se muestra el diseño que tienen en la página web del Área de Exposiciones:

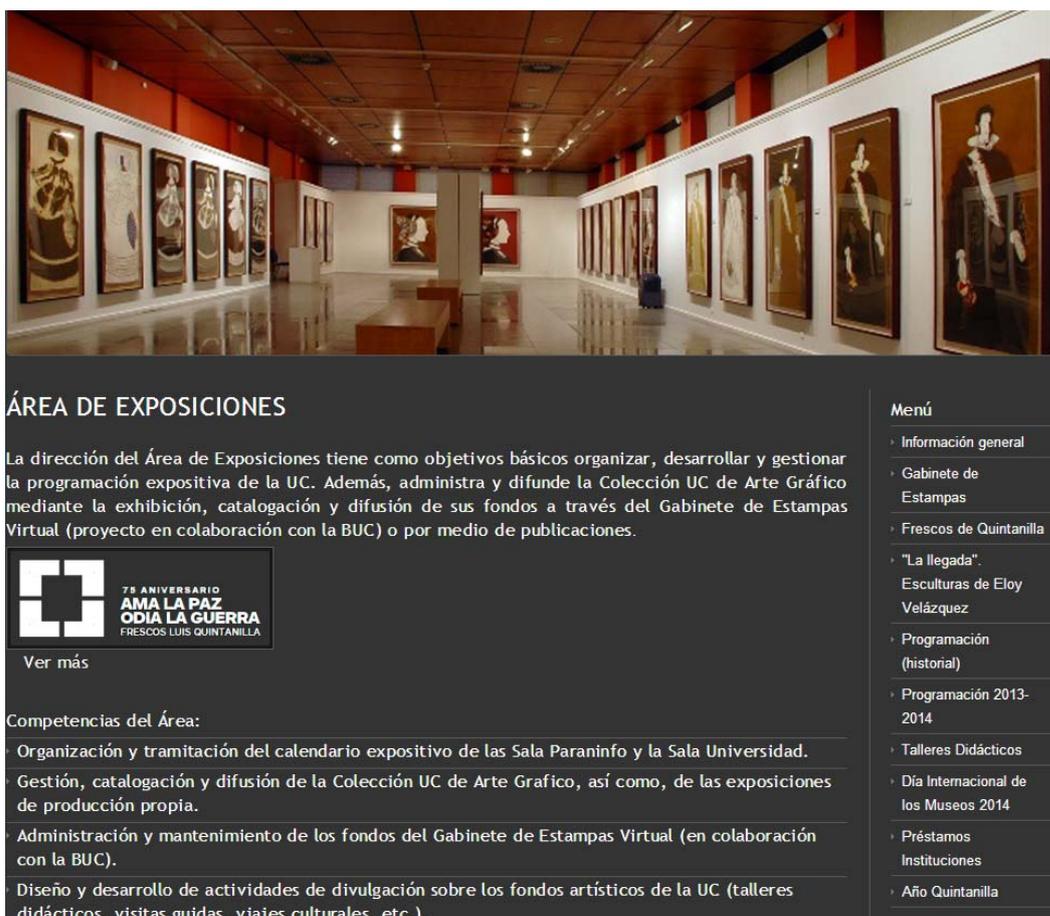


Figura 1 Diseño de la web anterior

Se puede apreciar que es una página web creada en texto plano en la cual solo disponen de un menú que permite el acceso a contenidos estáticos (en este caso documentos PDF).

No hay posibilidad de ofrecer al usuario una forma sencilla de visualizar las obras dentro de la misma página.

También destaca el desfasado diseño de la misma, no teniendo ninguna opción sobre la web 3.0, cómo por ejemplo un botón que facilite la difusión de la información por medio de las redes sociales y por supuesto su mala visualización en los dispositivos móviles de última generación.

## **1.2 Objetivos.**

Diseño y desarrollo de una aplicación web en la que se podrán categorizar las diferentes exposiciones y sus respectivas obras, dotándola de capacidad para interactuar con tecnologías como códigos QR o visitas virtuales.

Aplicación de un gestor de contenidos que permita la manipulación de datos de forma extremadamente sencilla para usuarios que no tengan conocimientos de informática, permitiéndoles gestionar la aplicación al completo.

## **1.3 Estructura de la memoria.**

En primer lugar hay que hablar del análisis de los requisitos, en el que se incluye todo el proceso de un ciclo software, haciendo especial hincapié en los requisitos funcionales tales cómo los casos de uso y no funcionales que abarca todos los programas y lenguajes utilizados para la realización del proyecto.

Después se expone la fase de diseño, en la que se observa cómo va a ir tomando forma el gestor de contenidos para en última instancia hablar de la fase de implementación, de cómo funciona y cómo se ha desarrollado el gestor de contenidos.

También al final de la memoria se habla de las conclusiones y trabajos futuros, así como de la bibliografía utilizada.

## **2. Análisis de requisitos.**

En este capítulo, se van a plantear, de manera detallada, aquellas funcionalidades y características con las que el sistema ha de contar para cumplir con los objetivos del proyecto.

Hay que tener en cuenta que el análisis de los requisitos de ingeniería de sistemas e ingeniería del software, es una de sus etapas críticas, ya que de él depende, en gran medida, el éxito de un proyecto en desarrollo. Éste comprende las tareas que van a determinar las necesidades o condiciones que deben satisfacerse en el nuevo producto. Además hay que tener en cuenta las posibles exigencias contradictorias de las distintas partes interesadas (stakeholders), tales como beneficiarios o usuarios, para buscar un punto de equilibrio.

Es aquí donde la interacción con el cliente es fundamental. Por ello a lo largo del proyecto se realizaron diversas reuniones técnicas tanto con los gestores como con los usuarios del Área de Exposiciones. Gracias a estas reuniones se establecieron las pautas para los diferentes tipos de requisitos.

Estos requisitos se han categorizado teóricamente en requisitos funcionales, y requisitos no funcionales.

## 2.1 Metodología del desarrollo

En el desarrollo de cualquier proyecto software es primordial utilizar una metodología para el desarrollo de un sistema, ya que con ella se estructurará, planificará y controlará las tareas necesarias durante el desarrollo. Para la realización de este sistema se siguió una metodología de las denominadas, de desarrollo ágil, concretamente la de Proceso Racional Unificado (PRU) [1].

De acuerdo con esta metodología se establecieron iteraciones con una duración en los entornos de trabajo de cuatro semanas. Cada una de estas iteraciones del ciclo de vida, incluye las fases de: Ingeniería y Análisis del sistema, Análisis de los Requisitos, Diseño, Codificación, Prueba y Evaluación, tal y como se puede ver en la Figura 2.1:

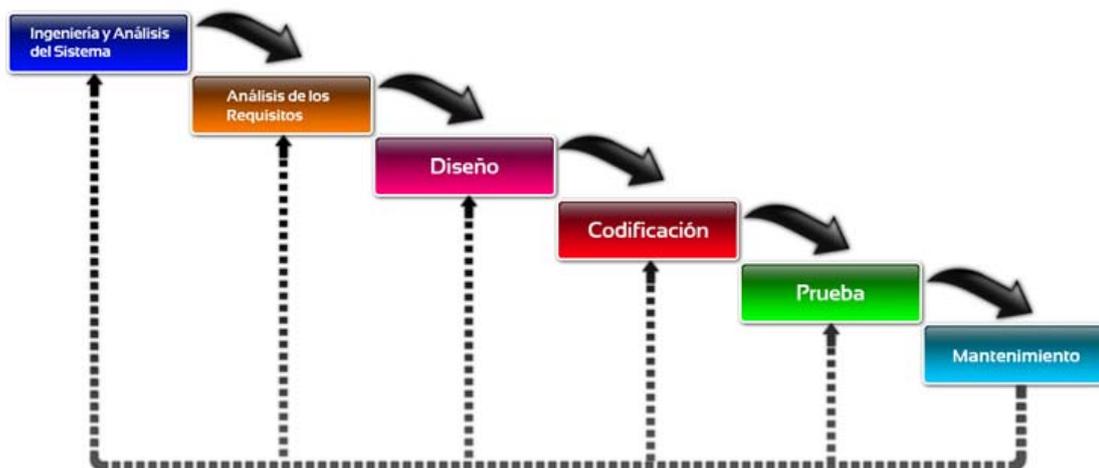


Figura 2.1 Fases del proyecto

Al final de cada una de estas iteraciones, el equipo se reunió para evaluar las prioridades del proyecto.

## 2.2 Requisitos funcionales.

Los requisitos funcionales, definen las funcionalidades que el producto debe proveer. Para encontrarlos, primeramente se debe identificar a los actores que van a utilizar el sistema, para así definir las funcionalidades requeridas por cada uno.

En este caso, los actores para los que se realizará un análisis de requisitos funcionales, estarán repartidos en dos grupos:

- Usuario no autenticado: Estos usuarios no tendrán permisos para identificarse en el sistema. Solo podrán acceder a la información de carácter público.

De acuerdo con la aplicación práctica del sistema estos usuarios serán los visitantes del Área de Exposiciones.

- Usuario autenticado: Todo usuario autenticado tendrá todas las funcionalidades descritas en usuarios no autenticados. Los actores de este grupo serán:
  1. Usuarios de gestión. Podrán realizar el proceso completo desde la creación de contenido hasta la presentación de este al usuario no autenticado. En este rol estarán los trabajadores del Área de Exposiciones.
  2. Usuarios de administración. Tendrán todas las funcionalidades descritas en el grupo de usuarios de gestión. Además podrán gestionar el aspecto del sitio, los permisos de los usuarios y archivos y podrán realizar labores de mantenimiento y actualización de la plataforma. Este usuario será la persona responsable del mantenimiento del software.

## 2.3 Casos de uso de los actores.

A continuación se va a hacer un estudio de los casos de uso asociados a los actores indicados anteriormente. Teniendo en cuenta que un caso de uso es una técnica utilizada para realizar la captura de requisitos de un sistema informático, cada caso de uso va a proporcionar uno o más escenarios que indican cómo debe actuar el sistema con los distintos usuarios o con otros sistemas.

Vamos a determinar simplemente la funcionalidad sin entrar a detallar la interacción con otros actores del sistema.

- El usuario no autenticado solamente podrá ver las exposiciones publicadas en la página así como la información relativa a ellas y a las obras expuestas en cada una, además podrá realizar un tour virtual en la que podrá contemplar dicha exposición, como se muestra en la Figura 2.2.

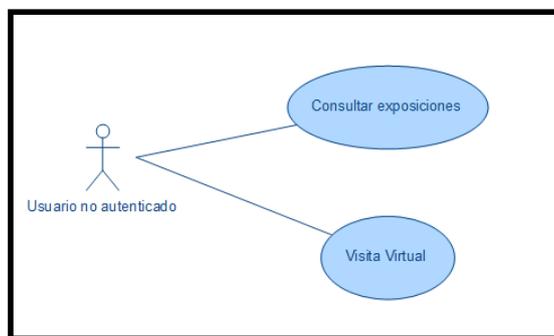


Figura 2.2 Usuario no autenticado

- El usuario autenticado, en este caso el usuario de gestión es la parte más crítica del sistema como puede apreciarse en la Figura 2.3. De esta forma va a poder realizar las siguientes operaciones:
  - Crear, editar o eliminar una exposición.
  - Agregar contenido a una exposición, editarlo o eliminarlo.
  - Generar códigos QR de cada cuadro contenido en una exposición.
  - Asignar las diferentes exposiciones a cada menú disponible, además podrá controlar la visibilidad de las mismas.
  - Exportar una exposición completa (con su respectivo contenido).
  - Importar una exposición previamente exportada.
  - Crear, editar o eliminar una visita virtual.

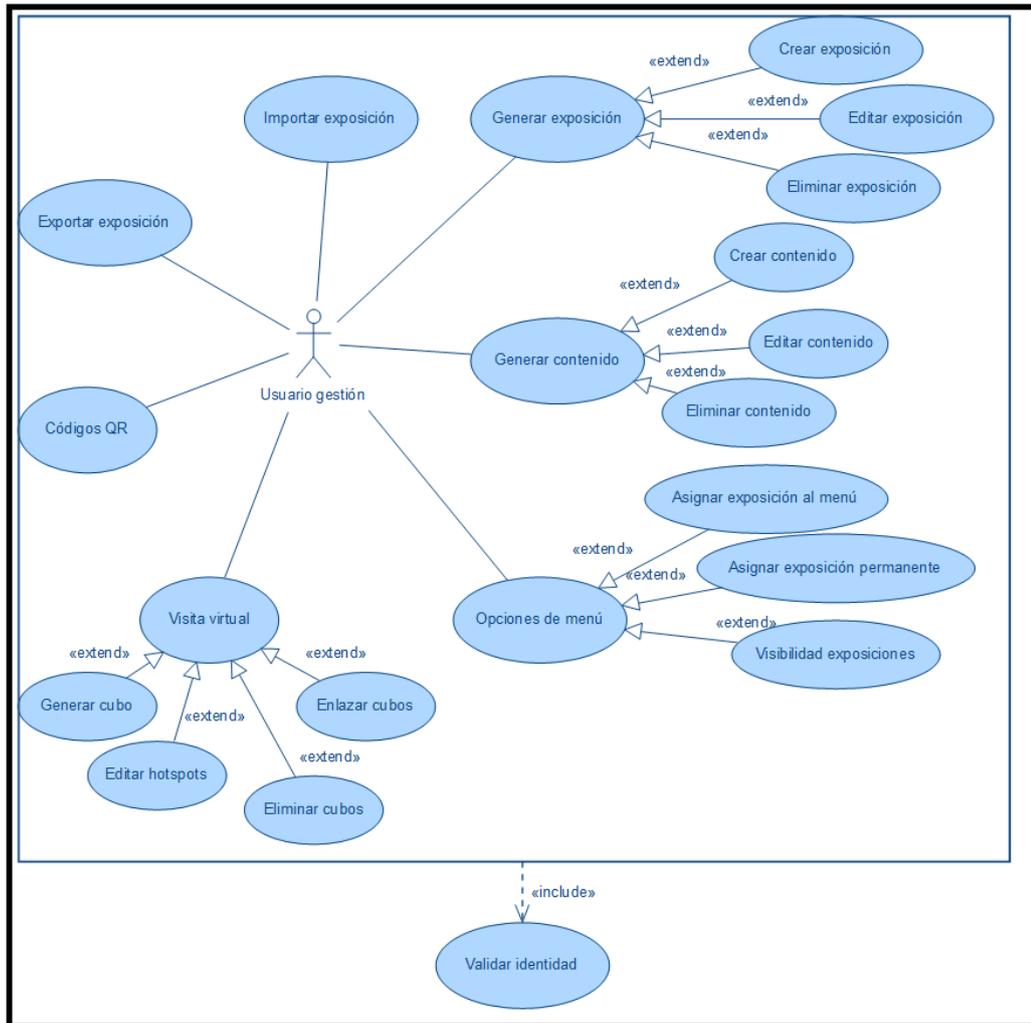


Figura 2.3 Usuario de gestión

- El usuario de administración dada su especialización podrá realizar además de todas las operaciones anteriormente descritas en el usuario de gestión las que se indican en la Figura 2.4.
  - Gestionar los usuarios
  - Asignación de permisos
  - Gestión de módulos
  - Mantenimiento de la plataforma web
  - Cambiar la estética de la plataforma web
  - Modificar base de datos

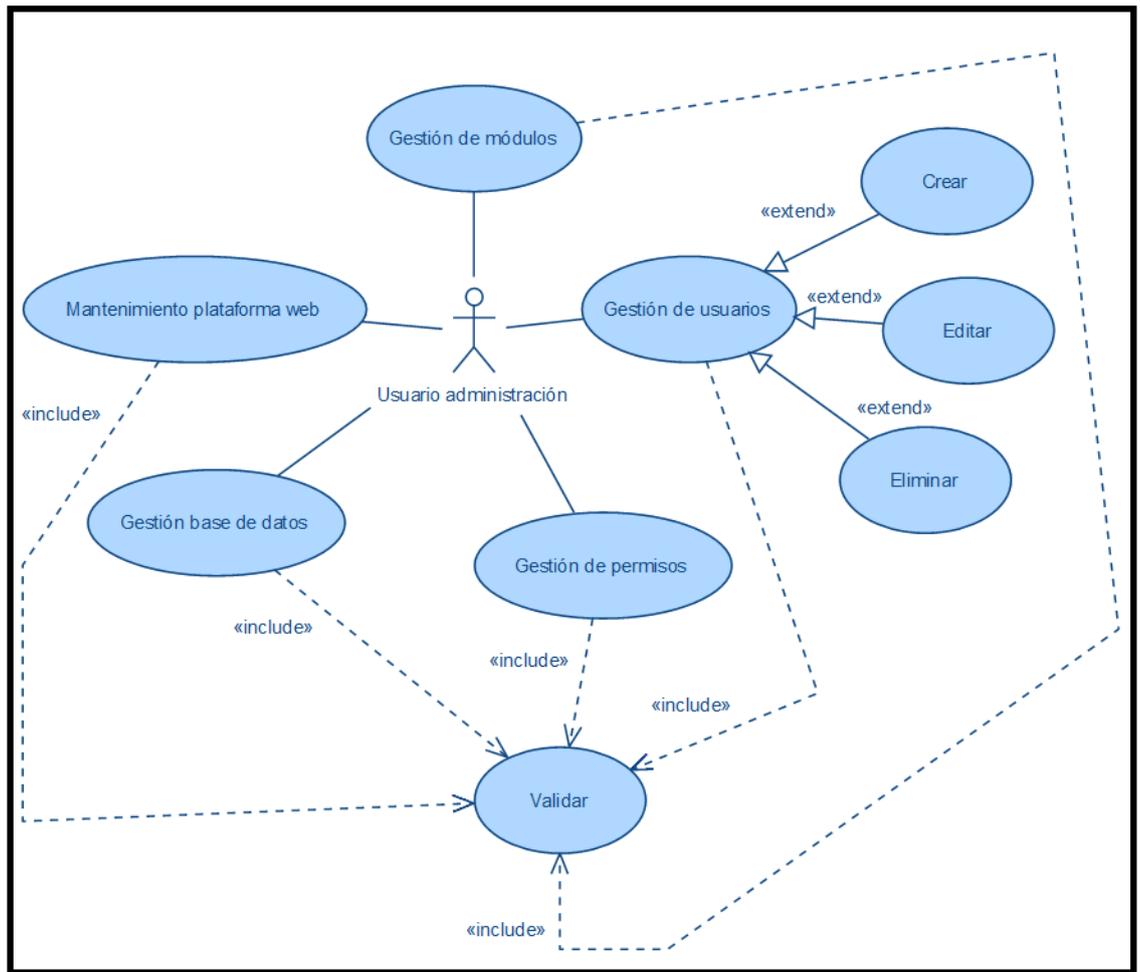


Figura 2.4 Usuario de administración

## 2.4 Requisitos no funcionales.

La definición de requisito no funcional especifica que es aquel que se refiere a todos los requisitos que ni describen información a guardar, ni funciones a realizar, es decir, especifica criterios que pueden usarse para juzgar la operación de un sistema. De acuerdo con las necesidades de la aplicación requerida por el Área de Exposiciones se identifican los siguientes items:

- Servidor Web

Lo primero que necesitará la aplicación, será un servidor en el que estar alojada, de manera que pueda mostrar todo su contenido a los usuarios a través del navegador.

Habitualmente los CMS( Content Management System - Sistema de gestión de contenidos) requieren que el servidor pueda generar páginas dinámicas, como por ej. a través de CGI o lenguajes interpretados como PHP[2]. De entre los diferentes tipos de servidores web que existen en el mercado, el más utilizado por los usuarios de CMS es Apache[3], ya que se trata de una herramienta de código abierto.

- Servidor de la base de datos

La base de datos que será la encargada de almacenar todo el contenido del sitio, junto con su configuración, usuarios y roles, será MySQL[4].

Toda información de la web quedará almacenada en la base de datos. Por otro lado, los scripts en PHP serán los encargados de trabajar con dicha información para convertirla en lo que el usuario final vea en la web. La decisión tomada fue la de utilizar MySQL5.

- Gestión de la base de datos

No es del todo necesario, pero ayudará al desarrollador o administrador a la hora de desarrollar o extender la aplicación. Es una herramienta que ofrece una interfaz para gestionar la base de datos y así el usuario podrá conocer todos y cada uno de los atributos de cada tabla. La opción más utilizada es PHPMyAdmin[5].

- Lenguaje de programación

Tiene que existir un lenguaje de programación encargado de transformar la información desde la base de datos hasta el servidor web. Para dicha tarea es necesario utilizar PHP5 (aunque esté probado con éxito desde la versión 4.4, la comunidad recomienda el uso de la versión 5), ya que posee conectividad tanto con MySQL (como es el caso de la Base de Datos del sistema a tratar).

- Usabilidad

La usabilidad en internet es la facilidad con que las personas pueden utilizar una herramienta web con el fin de alcanzar un objetivo concreto.

De cara al usuario no autenticado la herramienta dispone de unos menús para visualizar los distintos contenidos de forma rápida y sin complicaciones.

Una de las grandes bondades de esta aplicación es la facilidad de uso que tiene. En los requisitos iniciales, se especificó que la aplicación fuese construida para que fuese manejada por personas sin destreza en el ámbito informático. Es por eso que se tomó la decisión de crear un wizard, para hacer paso a paso cualquier modificación en el contenido de la aplicación.

- Accesibilidad

La accesibilidad web se refiere a la capacidad de acceso a la Web y a sus contenidos por todas las personas, independientemente de su edad y/o conocimientos tecnológicos.

Uno de los puntos más importantes que se tratarán aquí es diseño responsive (diseño adaptable), el cual se ha usado en la aplicación web. Gracias a este la aplicación se redimensiona en función del dispositivo que la esté visualizando (tabletas, móviles, etc.)

Ahora se analizarán a fondo los sistemas gestores de contenidos (CMS), junto con todas las tecnologías y herramientas de las que se sirven para su funcionamiento. Servirá de base para la presentación de la propuesta concreta necesitada por la sala de Exposiciones de la Universidad de Cantabria.

## 2.4.1 Tecnologías

Las principales tecnologías que se han utilizado son:

- **HTML (HyperText Markup Language) [6]:**  
Es un lenguaje utilizado para darle formato al texto. Es la base para la creación de páginas web. La forma de modelado del texto se realiza a partir del uso de etiquetas.
- **PHP (hypertext Pre-processor):**  
Es un lenguaje de programación interpretado, de código abierto, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente para la interpretación en el lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otro tipo de programa. El servidor se encarga de ejecutar el código fuente escrito en PHP y envía su resultado HTML al navegador.
- **MySQL:**  
Es un sistema de gestión de base de datos (SGBD) multiusuario, multiplataforma y de código abierto. Suele combinarse con el lenguaje, mencionado con anterioridad (PHP) a la hora de crear aplicaciones web. Se encuentra escrito en C y C++, y emplea el lenguaje SQL (Structured Query Language) para las consultas a la base de datos.
- **CSS (Cascading Style Sheets- Hojas de Estilo en Cascada)[7]:**  
Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (eXtensible Markup Language), y por extensión en XHTML (eXtensible Hypertext Markup Language).
- **Diseño Responsive**  
Es una técnica de diseño y desarrollo de sitios y aplicaciones web que permite que las páginas se adapten al tamaño, la resolución y orientación de la pantalla, y por tanto al dispositivo del usuario.

Se hace mediante el uso de un código HTML único pero que se presenta de manera distinta mediante:

- La separación entre el contenido y la presentación: todos los estilos están definidos en las CSS.
- **Layouts basados en grids:** la información se organiza en ejes verticales y horizontales. Tendremos definidos diferentes *layouts* en

diferentes CSS, por ejemplo uno de 3 columnas, uno de 2 columnas y uno de 1 columna.



Figura 2.5 Layouts basados en grids

- **Fluids Grids**, es decir, el uso de medidas relativas que permitan que el contenido se pueda adaptar realmente como se ve en la imagen anterior. Permite utilizar todo el espacio disponible y evitar el desplazamiento horizontal.
- **Media Queries**, permite cargar dinámicamente las diferentes CSS que hemos definido en función del tamaño de pantalla, su resolución o su orientación.
- **Configuración del meta viewport**: mediante este *meta* indicamos que nuestra web es flexible para adaptarse a los diferentes anchos y resoluciones de pantalla, le indicamos al navegador que no aumente o reduzca la página, que use el zoom por defecto.

## 2.4.2 Herramientas.

Una de las prioridades del proyecto era el uso de aplicaciones libres, es por ello que las herramientas utilizadas en este proyecto han sido fundamentalmente opensource:

- **Firebug[8]:**  
Es una extensión, inicialmente del navegador Firefox y ya incluida en los demás navegadores, creada y diseñada especialmente para desarrolladores y programadores web. Es un paquete de utilidades con el que se puede analizar (revisar velocidad de carga, estructura DOM 27 (Document Object Model)), editar, monitorizar y depurar el código fuente, CSS, HTML y JavaScript de una página web de manera instantánea.
- **PHPMYAdmin:**  
Es una herramienta escrita en PHP que permite manejar la administración de MySQL a través de páginas web. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos.
- **NotePad++[9]:**  
Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. De soporte nativo a Microsoft Windows. Se parece al Bloc de notas en cuanto al hecho de que puede editar texto sin formato y de forma simple. No obstante, incluye opciones más avanzadas que pueden ser útiles para usuarios avanzados como desarrolladores y programadores. Se distribuye bajo los términos de la Licencia Pública General de GNU.
- **Servidor Apache:**  
Es un servidor web multiplataforma, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Normalmente los usuarios recurren a sistemas denominados WAMP[10] o LAMP[11] que incluyen varias de las tecnologías que se han especificado anteriormente. Este servidor solo se ha utilizado en el proceso de desarrollo y testeo de la herramienta hasta su colocación en los servidores de la Universidad de Cantabria.
- **CMS:**  
Es un programa que permite la implementación de frameworks, para la creación y administración de contenidos, principalmente en portales web. Esta es la parte fundamental, porque nos permitirá integrar todas las funcionalidades y requerimientos especificados anteriormente.

### 2.4.3 Herramientas CMS

Estas siglas, en inglés, corresponden a un Sistema de Gestión de Contenidos (Content Management System).

Los CMS surgieron bajo la necesidad de distintas organizaciones de gestionar una gran cantidad de contenido en sus páginas web, el cual se veía afectado por frecuentes actualizaciones (como es el caso de periódicos, publicaciones corporativas, ) por personal no cualificado.

Los sistemas gestores de contenido se han convertido en una herramienta de mucha importancia debido al gran auge de las páginas web con abundante contenido y donde el usuario participa activamente (blogs, redes sociales, foros). A continuación se exponen algunos de los puntos más importantes que hacen útil y necesaria la utilización de un CMS:

- Inclusión de nuevas funcionalidades en el portal web. Como se ha comentado anteriormente, los CMS permiten agregar nuevos módulos o componentes, que son pequeñas aplicaciones que aportan funcionalidad extra al sitio. La inclusión de este tipo de aplicaciones en la web suele ser muy sencilla, lo que contribuye al crecimiento futuro de la página.
- Mantenimiento de gran cantidad de páginas. En una web con muchas páginas es necesario un sistema para distribuir los trabajos de creación, edición y mantenimiento entre un equipo de personas. Para estos se asigna a cada usuario una serie de permisos de acceso a las diferentes áreas.
- Reutilización de objetos o componentes. Los CMS permiten la recuperación y reutilización de componentes y contenido.
- Páginas interactivas. Como se ha explicado anteriormente, cuando un usuario realiza una petición de una página, el servidor genera una página dinámica, creada según las peticiones del usuario. Esto permite el uso de buscadores, por ejemplo, donde el sistema genera una página con unos resultados que no existían antes de la petición. Para ello, el servidor se conecta a la base de datos y extrae los datos necesarios.
- Cambios en el aspecto de la web. Al existir una clara separación entre contenido y aspecto (generalmente utilizando CSS), un cambio en el diseño de la página no afecta nada más que a los archivos que contienen las hojas de estilo.
- Control de acceso. Gracias a la gestión de usuarios que presentan los CMS, se puede controlar el acceso a las distintas partes de la página dependiendo de los permisos que tengan los usuarios.

Dentro de los diferentes CMS, actualmente destacan:

1. Drupal[12]:

Es un sistema muy configurable, que permite la publicación de artículos, imágenes: foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

2. Joomla[13]:

Incluye características como: mejorar el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls(encuestas), calendarios, búsqueda en el sitio web e internacionalización del lenguaje.

3. Wordpress[14]:

Inicialmente enfocado a la creación de blogs, se ha convertido en el más popular de la blogosfera . Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y la enorme comunidad de desarrolladores y diseñadores.

Si bien la elección de cualquiera de ellos sería posible, en nuestro caso se ha decidido el uso de Drupal al ser una gran opción cuando el sistema que se vaya a implementar sea un sitio complejo. Además de flexibilidad, soporta una amplia variedad de estructuras web; pueden definirse reglas muy detalladas sobre los contenidos y el lugar en el que se desea o no mostrar y, a diferencia de Joomla, construir nuevos tipos de contenido.

Por su parte Wordpress es un sistema demasiado estático, en el que adaptar contenidos resulta demasiado complicado. Sin embargo el potencial de Drupal conlleva un grado de dificultad, mayor que el de WP y Joomla, para comprender lo que el sistema ofrece y la manera de configurarlo. Las ventanas de administración de la web poseen un gran número de opciones y configuraciones, que dificultan algo su interpretación, y la flexibilidad del sistema obliga a dedicar tiempo a pensar en la mejor manera de conseguir lo que se quiere antes de realizarlo. En lo que a flujo de trabajo se refiere, Drupal se encuentra un escalón por encima, con la instalación de módulos creados para aquellos que buscan tener personas con roles diferentes que controlen el contenido del sitio. Con todo las principales razones por las cuales Drupal resulta más atractivo frente a las otras dos opciones se pueden resumir en las siguientes:

- Está desarrollado mediante el modelo de tres capas que independiza los datos del diseño, por lo que casi cualquier diseño es aplicable a Drupal.
- Extensibilidad: más que poseer una amplia variedad de cosas, es más bien una buena base en la que poder integrar todo tipo de módulos.

- El sistema base de gestión de contenido es limpio, flexible y potente. Los diferentes contenidos, son tratados de una manera uniforme; una noticia, un evento, un enlace o mismamente un proyecto son un objeto de contenido, y sobre cualquiera de ellos se pueden activar los comentarios, presentar como un listado o a página completa, realizar búsquedas...
- Sistema de categorías (taxonomías). Drupal cuenta con este sistema para categorizar todo el contenido, siendo mucho más sencillo el trabajar con ello, en cuanto a reglas y permisos se refiere.
- Rápido y optimizado para no saturar el servidor. El consumo de recursos del servidor que hace Drupal es mínimo. Para ello, el sistema de Caché permite acelerar y reducir dicho consumo.
- El control para los administradores sobre el sistema es amplio. Permisos por usuario y módulo. Informes de registros, informes de formularios, están por defecto disponibles y son muy fáciles de exportar a excel.
- Una gran comunidad en crecimiento. Existen un gran número de usuarios, de sitios y de empresas que utilizan Drupal, asegurando que el soporte, y los desarrolladores, es poco probable que llegue a desaparecer. Desde el punto de vista de los posibles clientes, esto significa una garantía de que, incluso cuando el creador del sitio no pueda ser contactado, siempre se va a poder encontrar a otra persona o empresa que pueda darles soporte o extienda la web.

### **3. Diseño**

Una vez que se han realizado el análisis de requisitos, casos de uso y las herramientas que se van a utilizar, se procede a analizar y realizar el diseño de la aplicación.

El diseño web es una actividad consistente en la planificación, diseño e implementación de sitios web y páginas web. No es simplemente una aplicación del diseño convencional, ya que requiere tener en cuenta algunas de las funciones vistas, como pueden ser: la accesibilidad, usabilidad.

La unión de un buen diseño con una jerarquía bien elaborada de contenidos aumenta la eficacia de la web como canal de comunicación e intercambio de datos.

La clave del diseño óptimo de una aplicación es elaborar una buena estrategia de contenidos para las personas a las que va a llegar, proporcionándoles una experiencia satisfactoria como usuarios. Para optimizar la web hasta conseguir dicho objetivo, se necesita medir y evaluar los datos que podamos obtener de los usuarios.

### **3.1 Fases del diseño.**

A la hora de diseñar el sitio, lo primero que hay que tener claro es la meta a alcanzar de acuerdo a los objetivos del proyecto:

- Construir una plataforma web que siguiera la estética de la página principal del Campus Cultural.
- Crear un gestor de contenidos intuitivo y de fácil uso con el que controlar la aplicación entera.
- La aplicación web ha de ser completamente compatible con dispositivos móviles.
- Buscar más formas de interacción entre la aplicación y dispositivos móviles.
- Incluir una visita virtual, actualizable con las diferentes exposiciones que haya.

### **3.2 Diseño del gestor de contenidos.**

Es lo primero que hay que diseñar. Gracias a él los usuarios gestionarán todo el contenido que quieran mostrar en la aplicación web.

Inicialmente, gracias a los requisitos del proyecto, se sabe lo siguiente:

1. Existirán dos tipos de exposiciones, una llamada exposiciones y otra llamada frescos.
2. Dentro de las anteriores exposiciones se podrá guardar contenido de tres tipos distintos (obra, grabado, fresco) y cada una de ellas tendrá campos diferentes en los que guardar los respectivos datos que le correspondan a cada contenido.
3. Cualquier tipo de contenido así como cualquier exposición podrá ser creada, editada o eliminada.
4. Estas exposiciones se deberán poder asignar a los diferentes menús requeridos, u ocultarse en caso de no querer ser mostradas.
5. Todas las obras expuestas deberán de llevar asociado un código QR.
6. Las exposiciones y su contenido asociados se podrán exportar e importar a un archivo externo en formato zip.
7. Se podrá gestionar una visita virtual, creándola desde cero pudiendo enlazar con diferentes salas y dentro de las mismas se podrá colocar puntos de interés que redirijan a la información de las obras o que extiendan la información de las mismas.
8. Todo los puntos anteriores deben ser compatibles con smartphones o tablets.

De acuerdo con la filosofía de Drupal se deben de tener claro los siguientes conceptos:

## **Taxonomías**

La Taxonomía (taxonomy) es el sistema mediante el cual Drupal clasifica el contenido y es uno de los módulos del núcleo de Drupal. Se pueden definir los vocabularios propios como grupos de términos de taxonomía. Cada tipo de vocabulario puede ser agregado como uno o más tipos de contenido y por ello, los nodos en el sitio pueden ser clasificados según agrupaciones en categorías, etiquetas o como cualquier cosa que se elija.

Gracias a esta funcionalidad, aquí se pueden crear los dos tipos de exposiciones y asociarles unos campos con la información que necesiten (imagen, nombre, fecha...).

Luego dicha información se introduce mediante un "wizard" (asistente de creación de exposición), programado en php.

Con esto queda resuelto el punto 1.

### **Tipos de contenido (Nodo)**

Creando un nodo defines el tipo de contenido a introducir en la aplicación, incluyendo sus campos y como deben ser estos.

Teniendo en cuenta lo descrito en el párrafo anterior, con esto queda resuelto el punto número 2, ya que se puede crear tres tipos de contenidos distintos (fresco, grabado y obra) con sus correspondientes campos cada uno, y crear un campo de tipo taxonomía para asociar cada obra con su exposición correspondiente (las definidas en el punto anterior).

Mediante un programa php se crea un wizard (asistente de tipo de contenido) en el que se da la opción de agregar contenido, editarlo u eliminarlo, resolviendo el paso 3 también.

## **Views**

Es considerado el módulo más potente por la comunidad de Drupal. Gracias a la interfaz gráfica que se instala al activar éste módulo, se permite al usuario realizar consultas a la base de datos de su sitio web. Además puede mostrarlas de una forma determinada y en el lugar deseado. En este caso el administrador de la herramienta es el encargado de gestionar las vistas creadas.

La interfaz de views está dividida en diversos apartados o secciones y que según la forma en que se configuren cada uno de los parámetros de dichos apartados, será la responsable del aspecto final que tengan las vistas.

De esta forma se resuelve el punto número 4. Se crean unas views con una serie de

filtros (muchos de ellos definidos por campos creados en cada nodo), y así se muestran las exposiciones.

Por último se crea un programa php el cual gestionará los campos de los diversos tipos de contenidos y de las taxonomías que los cambia en función de lo que se quiere mostrar.

### **Importar / Exportar exposición**

El primer paso consiste en realizar una búsqueda exhaustiva de un módulo que incluya esta funcionalidad a la aplicación en la comunidad de Drupal.

Desafortunadamente los módulos ya creados que otorgan este tipo de funcionalidad solo te exportan los datos de las obras a formato CSV [15], y debido a la distribución de las taxonomías (exposiciones) y de los tipos de contenidos (obras) en esta aplicación no se importa la relación entre ellos que los identifica dentro del mismo conjunto.

Además cada exposición y obra contiene un campo imagen y aún no hay ningún módulo que lo exporte / importe.

Debido a las dificultades halladas, habrá que crear un programa en php que permita exportar e importar exposiciones manteniendo la coherencia que existe entre ellas.

Con este programa quedará resuelto el punto número 6.

### **Visita virtual**

Primero se tiene que elegir un programa para integrarlo en la aplicación y que lo muestre. Pero debe tener una serie de requisitos:

- Debe de ser un programa Open Source.
- Debe de estar construido en HTML 5, ya que es la única forma de que se reproduzca en la gran mayoría de dispositivos móviles.

El primer paso no supone mucho problema porque rápidamente se encuentran varios programas que cumplen el requisito.

El problema es que la mayoría de ellos están escritos en Flash [16], y eso incumple el segundo requisito, por lo que la búsqueda se limita bastante.

Finalmente se consigue encontrar un programa que cumple con los dos requisitos. Para llevarlo a la aplicación, se crean cinco programas php correspondientes a subir un tour, asignar puntos de interés, eliminar tour, enlazar diversos tours y asignar al correspondiente menú el tour que se quiera mostrar en ese momento.

Con esto quedaría solventado el punto 7.

### **Plantillas Responsive**

Ya se ha hablado anteriormente del significado de que algún elemento sea responsive.

Para solucionar el punto 8 se introducirá una plantilla de este tipo que redimensione toda la aplicación.

### **Generación de códigos QR [17]**

Gracias a estos códigos, los usuarios podrán acceder directamente a la información relativa a las exposiciones por medio de un dispositivo que interprete las url contenidas en la matriz de dibujo que conforme el código (mediante un smartphone o un tablet).

Se implementa mediante un programa en php que conecta con el Google Api y devuelve el qr. Para su posterior manipulación se instalará un módulo llamado doomPDF[18], que otorga la funcionalidad de generar documentos en PDF[19] y guardarlos.

Por lo tanto se selecciona la exposición deseada, se envía a la Google Api[20] todas las url de las obras que contengan la exposición, y cuando devuelva los códigos QR se guardan en un pdf, resolviendo de esta forma el punto 5.

Además de todo lo anterior, se decidió la introducción de las siguientes mejoras adicionales al diseño:

- Un sistema de url's limpias.
- El módulo css injector para modificar los diferentes css de forma más rápida.
- El módulo devel, que incluye una serie de herramientas que facilitan el desarrollo de la aplicación web y su testeo.
- El módulo Backup and Migrate que proporciona una manera sencilla de exportar/importar la base de datos de la aplicación.
- La apariencia para que se parezca lo más posible a la página del Campus Cultural de la Universidad de Cantabria.
- Se integra una herramienta para que se puedan compartir los enlaces de las obras de forma fácil mediante las redes sociales.

## 4. Implementación.

Por no sobrecargar la memoria no se introducirá ningún fragmento de código. Para crear el asistente paso a paso, se utilizan los formularios definidos en el Form API [21].

### 4.1 Taxonomías.

De acuerdo con el diseño planteado, se comienza creando las taxonomías, que se corresponden con las exposiciones de la aplicación, tal y como se muestra en la Figura 4.1.1.

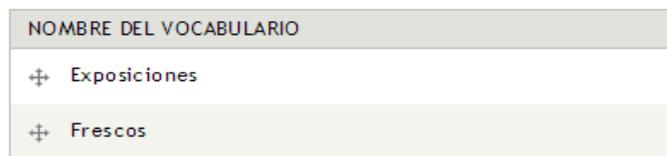


Figura 4.1.1 Vocabularios de la aplicación

Cada taxonomía tiene asociados unos campos, que se corresponden con la información que se desea guardar en cada uno de ellos. En la Figura 4.1.2 se analiza la taxonomía exposiciones:

ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO
+ Nombre	name	Campo de texto del nombre del término
+ Imagen exposición	field_informacion_imagen_expo	Imagen
+ Año	field_informacion_ano_expo	Texto
+ Periodo de muestra	field_informacion_expo_periodo_m	Texto
+ Lugar de muestra	field_informacion_expo_lugar_mue	Texto
+ Información	field_informacion_expo	Texto largo
+ Descripción	description	Área de texto de la descripción del término
+ activa	field_activa	Booleano
+ visibilidad	field_visibilidad	Booleano
+ permanente	field_permanente	Booleano

Figura 4.1.2 Campos de las taxonomías

Se observa que cada exposición cuenta con un nombre, un campo de imagen, un campo de año, un campo que indica el periodo de muestra, el lugar de muestra, la descripción de la exposición y la información de la exposición.

Así mismo se ha implementado tres campos booleanos, con los que se manejan las opciones de visibilidad más adelante en las views:

- Con el campo activa a 1, la exposición se muestra en la pestaña del menú exposición actual, a 0 se muestra en exposiciones anteriores
- Si el campo de visibilidad es igual a 1, se muestra la exposición a los usuarios, en caso contrario se oculta.
- Si el campo permanente es 1, la exposición se muestra en la pestaña del menú exposición permanente.

A continuación se crea el programa PHP que permite controlar lo creado anteriormente en la aplicación de Drupal, basado en el diagrama de flujo descrito en la Figura 4.1.3.

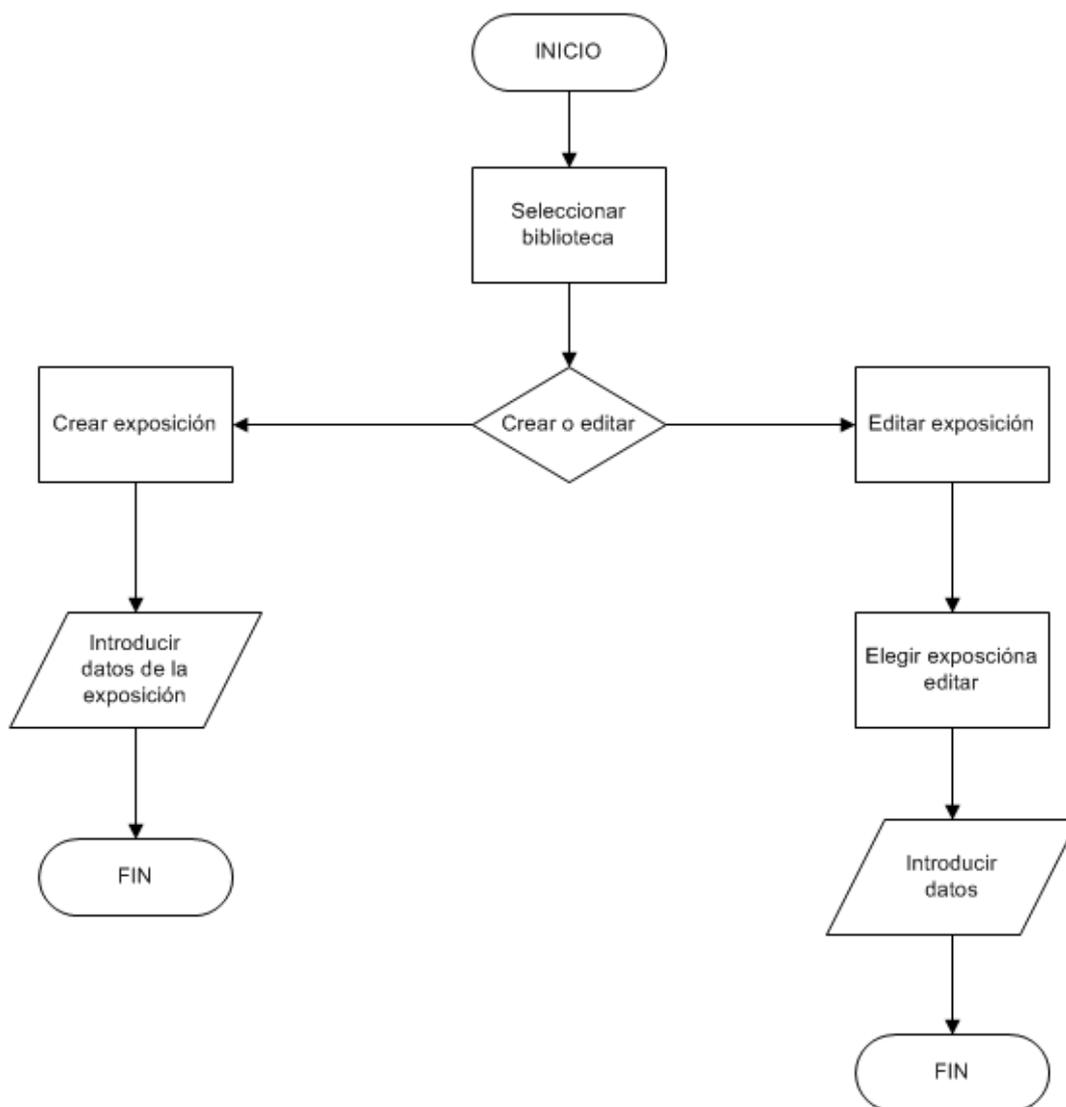


Figura 4.1.3 Diagrama de flujo del programa crear exposición

El programa crea un asistente paso a paso, en el que se seleccionan los datos necesarios para completar la exposición, y finalmente se crea un término de taxonomía con los datos requeridos y se guarda en la aplicación.

Este es el aspecto que tiene la opción de crear/editar una nueva exposición, tal y como se muestra a continuación en las Figuras de la 4.1.4 a la 4.1.8.



**Editar exposición: Paso 1**

Crear exposición

Selecciona la biblioteca donde se creará la exposición

Exposiciones ▾

Siguiente

Figura 4.1.4 Elegir biblioteca



**Editar exposición: Paso 2**

▼ Crear exposición

Elige la opción de exposición a realizar

Crear nueva exposición

Editar exposición existente

Anterior Siguiente

Figura 4.1.5 Crear o editar



▼ Crear exposición

Rellena los datos para crear una exposición nueva

**Nombre Exposición \***

Año Exposición

Período de muestra

Lugar de muestra

Introduce la descripción de la exposición

Ruta de la imagen a guardar

Seleccionar archivo Ningún archivo seleccionado

Extensiones permitidas: jpg, jpeg, png, gif

Anterior Siguiente

Figura 4.1.6 Introducir datos de la exposición



Figura 4.1.7 Seleccionar exposición a editar

▼ Editar exposición

**Nombre Exposición \***  
Fábulas de Samaniego: La Naturaleza Humana

**Año Exposición**  
2014

**Periodo de muestra**  
9 mayo al 28 junio

**Lugar de muestra**  
Paraninfo de la Universidad

**Introduce la descripción de la exposición**

La exposición Fábulas de Samaniego: La naturaleza humana es un proyecto que recoge 27 grabados relacionados con la obra de Félix María de Samaniego (1745-1801), pertenecientes a la colección del Ayuntamiento de Vitoria-Gasteiz y depositados para su conservación y difusión en Biblioteca Universitaria de Las Nieves (UPV). Estas estampas de finales del s.XVIII y principios del siglo XIX realizan un recorrido por las distintas fábulas, cuya primera edición se publicó en Valencia, en el año 1781 bajo el título Fábulas en verso castellano para el uso del Real Seminario Bascongado. La obra recoge 157 fábulas, distribuidas en 9 libros en donde se analiza y ridiculiza la naturaleza humana al

**Ruta de la imagen a guardar**  
[Seleccionar archivo] Ningún archivo seleccionado  
Extensiones permitidas: jpg, jpeg, png, gif

Anterior    Finalizar

Figura 4.1.8 Introducir datos a editar

## 4.2 Tipos de contenido

En la Figura 4.2.1 se muestran todos los tipos de contenido:

NOMBRE
Artículo (Nombre de sistema: article) Utilice <i>artículos</i> para contenido que dependa de fecha y hora, como ser noticias, artículos de prensa o entradas de blogs.
Frescos (Nombre de sistema: frescos)
Grabado (Nombre de sistema: grabado)
Obra (Nombre de sistema: obra)
Panorama (Nombre de sistema: panorama)
Página básica (Nombre de sistema: page) Use las <i>páginas básicas</i> para su contenido estático, como una página "Acerca de...".
agenda-cultural (Nombre de sistema: agenda_cultural)
pagina inicio (Nombre de sistema: pagina_inicio)
posible-panorama (Nombre de sistema: posible_panorama)
sub_pano (Nombre de sistema: sub_pano)

Figura 4.2.1 Tipos de contenido de la aplicación

Se describe su función:

- Frescos: crea un tipo de contenido con unos campos determinados.
- Grabado: crea un tipo de contenido con unos campos determinados.
- Obra: crea un tipo de contenido con unos campos determinados.
- Panorama: crea un tipo de contenido con los datos HTML para lanzar la aplicación de tour virtual en la view correspondiente.
- Página inicio: crea el contenido que se mostrará en la portada de la aplicación
- Posible-panorama, sub\_pano: crea valores necesarios en cada tour que determinan el tipo de tour que es y datos de inicialización

Para no sobrecargar la memoria, solamente se explica cómo funciona el tipo de contenido Frescos, mostrado en la Figura 4.2.2.

ETIQUETA	NOMBRE DE SISTEMA	TIPO DE CAMPO	CONTROL
+ Title	title	Elemento del módulo Nodo	
+ Imagen	field_imagen	Imagen	Imagen
+ Año	field_fecha	Texto	Campo de texto
+ Autor	field_autor	Texto	Campo de texto
+ Tipo	field_tipo	Texto	Campo de texto
+ Dimensiones	field_dimensiones	Texto	Campo de texto
+ Propiedad	field_propiedad	Texto	Campo de texto
+ Descripción	field_descripci_n	Texto largo	Área de texto (varias filas)
+ Comentarios	field_comentarios	Texto largo	Área de texto (varias filas)
+ Referencias	field_referencias	Texto largo	Área de texto (varias filas)
+ Etiqueta_nfc	field_etiqueta_nfc	Booleano	Casillas de selección / botones de opciones
+ exposiciones	field_exposiciones	Referencia de término	Casillas de selección / botones de opciones
+ frescos	field_frescos	Referencia de término	Casillas de selección / botones de opciones
+ Información complementaria	field_adicional	Archivo	Archivo

Figura 4.2.2 Campos del tipo de contenido Frescos

Se puede observar que hay 2 grupos de campos:

- El primer grupo hace referencia a los datos de cada obra que se vaya a guardar (título, imagen, año, autor, tipo, dimensiones, propiedad, descripción, comentarios, referencias, etiqueta nfc, información complementaria).
- El segundo grupo serían los campos exposiciones y frescos. Estos campos son del tipo referencia de término, es decir campos de las taxonomías previamente creadas, y será la forma de asociar cada obra a su correspondiente exposición.

En la Figura 4.2.3 se observa que en todos los campos también hay que gestionar su presentación, en este caso a través de las views.

CAMPO	ETIQUETA	FORMATO
+ Imagen	<Hidden> ▼	Imagen ▼
+ Año	Encima ▼	Predeterminado ▼
+ Autor	Encima ▼	Predeterminado ▼
+ Tipo	Encima ▼	Predeterminado ▼
+ Dimensiones	Encima ▼	Predeterminado ▼
+ Propiedad	Encima ▼	Predeterminado ▼
+ Descripción	Encima ▼	Predeterminado ▼
+ Comentarios	Encima ▼	Predeterminado ▼
+ Referencias	Encima ▼	Predeterminado ▼
+ Información complementaria	Encima ▼	Archivo genérico ▼
<b>Oculto</b>		
+ Etiqueta_nfc	Encima ▼	<Hidden> ▼
+ exposiciones	Encima ▼	<Hidden> ▼
+ frescos	Encima ▼	<Hidden> ▼

Figura 4.2.3 Gestión de la presentación del tipo de contenido Frescos

Como se ve en la imagen, se define en qué lugar va a ir la etiqueta que identifica al campo (si es que hace falta), y se ocultan los campos que no quieren que se muestren en la presentación final de cara al usuario.

## 4.2.1 Añadir obras

En este punto se tratan las opciones programadas en PHP (siguiendo el diagrama de flujo mostrado en la figura 4.2.1.1) para añadir las obras, que se corresponden con los tipos de contenidos frescos, grabados, obras.

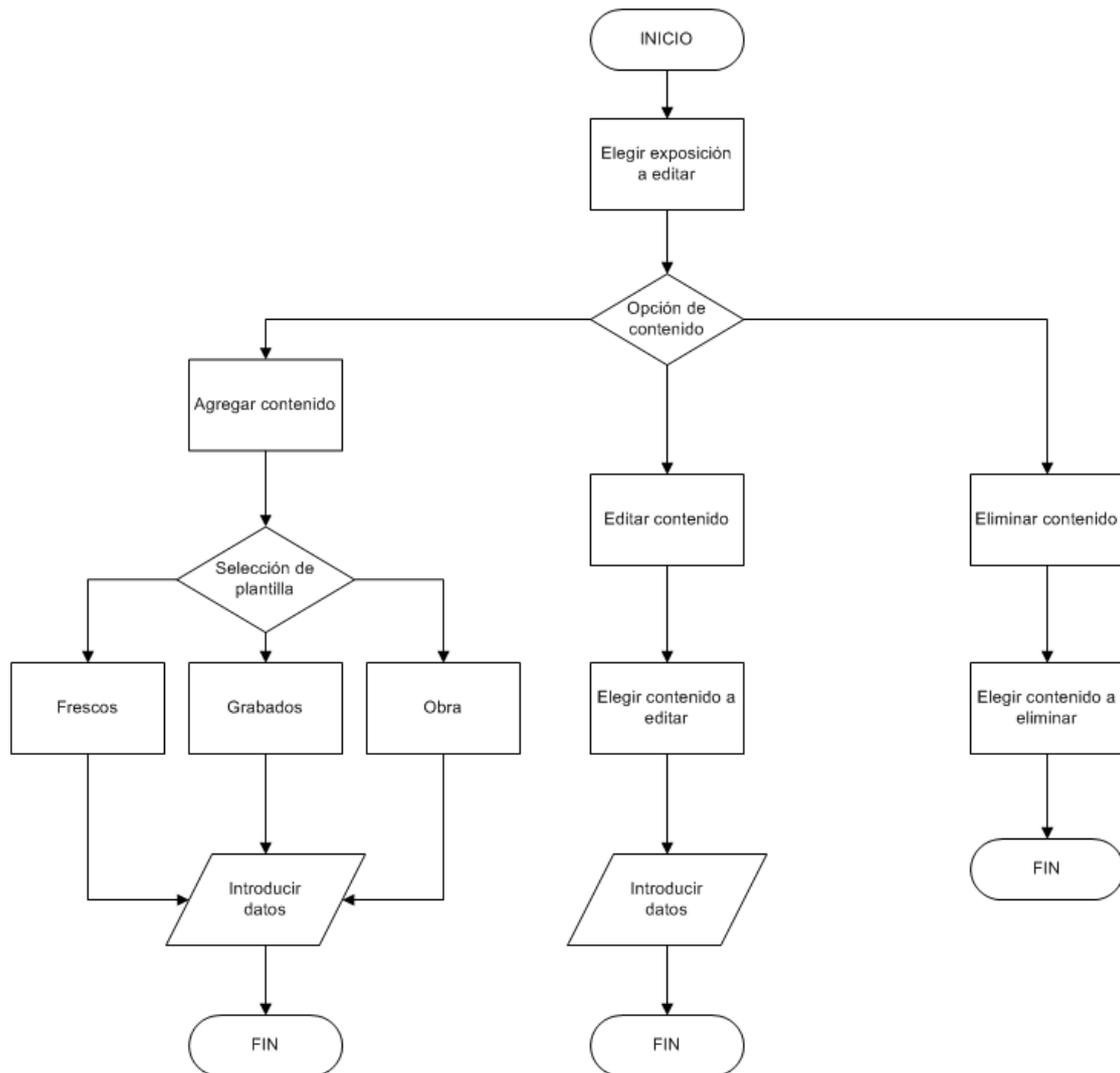


Figura 4.2.1.1 Diagrama de flujo del programa Editar exposición

En el ejemplo que comprenden las Figuras desde la 4.2.1.2 a la 4.2.1.11 se hacen los pasos necesarios para crear un tipo de contenido de tipo grabado, su edición y finalmente su eliminación.

- **Añadir contenido de tipo grabado**

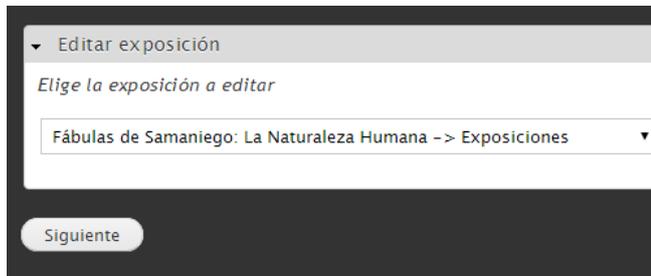


Figura 4.2.1.2 Se elige la exposición



Figura 4.2.1.3 Se elige una opción de contenido

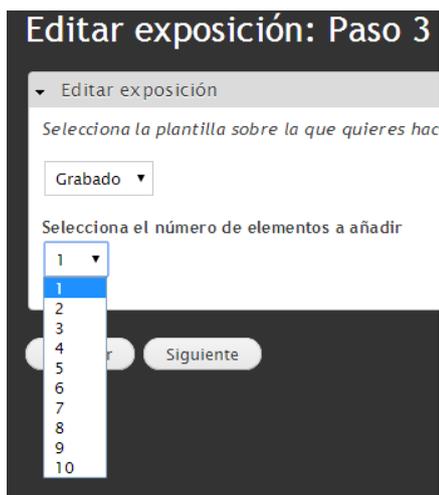


Figura 4.2.1.4 Se selecciona la plantilla deseada (de los tres tipos de contenidos descritos previamente) y el número de obras a añadir

Figura 4.2.1.5.1 Se rellenan los datos que se quieren almacenar

Figura 4.2.1.5.2 Se rellenan los datos que se quieren almacenar

Y con esto queda creado el tipo de contenido, almacenado en grabado y enlazado a la taxonomía seleccionada en el primer paso.

- **Editar contenido:** Se retoma a partir del paso 2.

Figura 4.2.1.6 En este caso se selecciona la opción editar

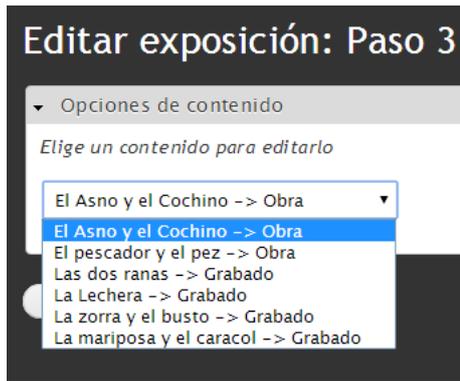


Figura 4.2.1.7 Se selecciona la obra sobre la que queremos editar

The screenshot shows the "Editar contenido" form. At the top, it says "Rellena todos los campos para crear un contenido nuevo". The form contains several input fields: "Título" (filled with "El Asno y el Cochino"), "Año" (filled with "2012"), "Autor" (filled with "Juan Pablo Álvarez"), "Técnica" (filled with "Aguafuerte y aguainta"), and "Dimensiones" (empty). Below these is a section for "Ruta de la imagen a guardar" with a "Seleccionar archivo" button and the text "Ningún archivo seleccionado". It also lists "Extensiones permitidas: jpg, jpeg, png, gif". The "Procedencia" field is filled with "Escuela de Arte y Superior de Diseño de Vitoria". The "Descripción" field contains the text: "Envidiando la suerte del Cochino, un Asno maldecía su destino. <Yo, decía, trabajo y como paja; él come harina, berza, y no trabaja; a mí me dan de palos cada día;".

Figura 4.2.1.8.1 Se edita el contenido deseado

The screenshot shows the "Comentarios" and "Referencias" sections of the form. Both are empty text areas. Below them is a section for "¿Llevará etiqueta NFC?" with radio buttons for "Sí" and "No", where "No" is selected. There is also a "Campo adicional: Ruta de la imagen a guardar" section with a "Seleccionar archivo" button and the text "Ningún archivo seleccionado". It lists "Extensiones permitidas: jpg, jpeg, png, gif, pdf". At the bottom, there are two buttons: "Anterior" and "Finalizar".

Figura 4.2.1.8.2 Se edita el contenido deseado

La aplicación permite dejar campos en blanco, ya que en muchas de las ocasiones no será necesario rellenar todos ellos para la información de una obra en concreto.

- **Eliminar contenido:** Se retoma a partir de la figura 5.2.1.2

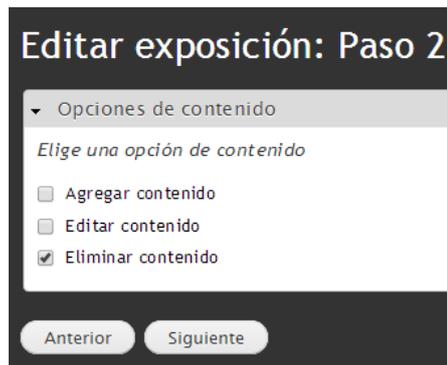


Figura 4.2.1.9 Eliminar contenido

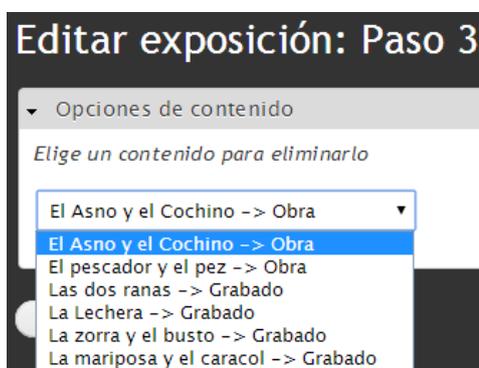


Figura 4.2.1.10 Se elige el contenido a eliminar

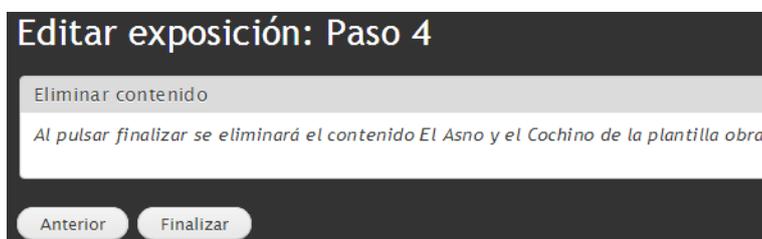


Figura 4.2.1.11 Mensaje de advertencia que indica que se eliminará el contenido

## 4.2.2 Añadir visita virtual.

Antes de entrar a mostrar cómo se ha implementado de cara al usuario, se explica lo que es un panorama y cómo funciona este programa.

Un panorama se define como una serie de fotografías, que juntándolas obtienes una vista de 360 grados.

Para realizar un panorama (por ejemplo una sala de exposición) en el caso del programa se necesitan 6 imágenes, correspondientes con un cubo (las cuatro paredes, el techo y el suelo).

En esta sección se tratan cuatro programas:

- **Cargar panorama**, representado por el diagrama de flujo de la Figura 4.2.2.1

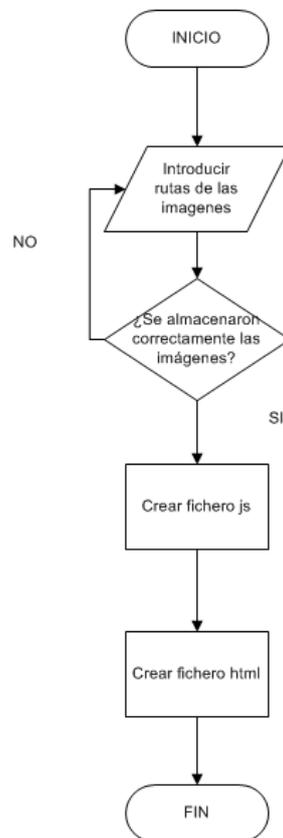


Figura 4.2.2.1 Diagrama de flujo Cargar Panorama

Gestiona la carga del panorama y la generación de scripts necesarios para su funcionamiento.

- **Asignar puntos de interés (hotspots)**, representado en la Figura 4.2.2.2

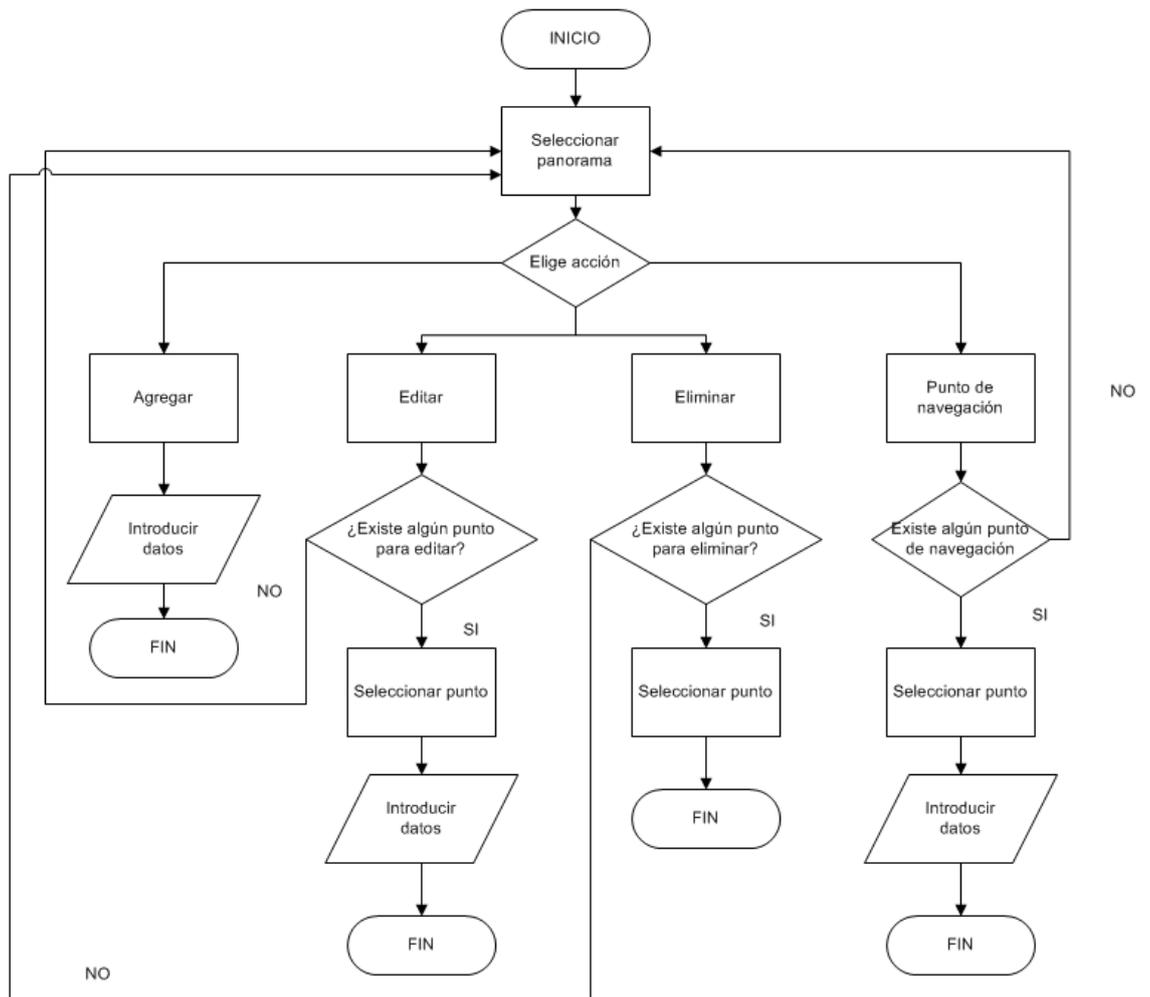


Figura 4.2.2.2 Diagrama de flujo Editar hotspots

Al panorama ya creado y en funcionamiento en la aplicación web se le pueden añadir puntos de interés (por ejemplo una url de la información de cada obra, datos de interés, videos youtube...). Este programa gestiona desde la creación de dichos puntos hasta su posible posterior edición o eliminación.

- **Eliminar panorama**, representado en la Figura 4.2.2.3

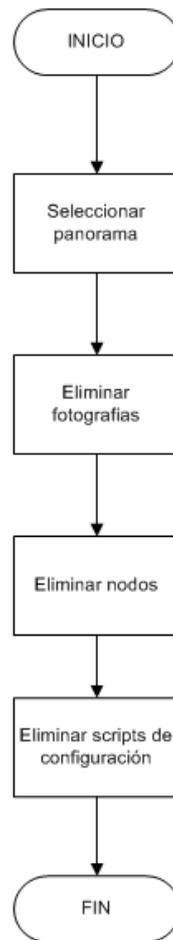


Figura 4.2.2.3 Diagrama de flujo Eliminar Panorama

Simplemente este programa elimina todas las imágenes asociadas y todos los scripts de los que se compone un panorama

- **Enlazar panorama**, representado en la Figura 4.2.2.4



Figura 4.2.2.4 Diagrama de flujo Enlazar Panorama

Para generar una visita virtual (que puedas pasar de una sala a otra), se necesitan varios panoramas. Este programa te permite mediante un punto de interés que se crea de forma automática (una vez creado es posible editarlo en la opción asigna hotspot) unir las diferentes salas dentro de un mismo conjunto, conformando una visita virtual.

Una vez explicado cómo funcionan internamente los programas de los que se compone la visita virtual, se muestra como se crean de cara al usuario que tiene que introducir los datos para generarla:

### 1. Cargar panorama, representado en la Figura 4.2.2.5

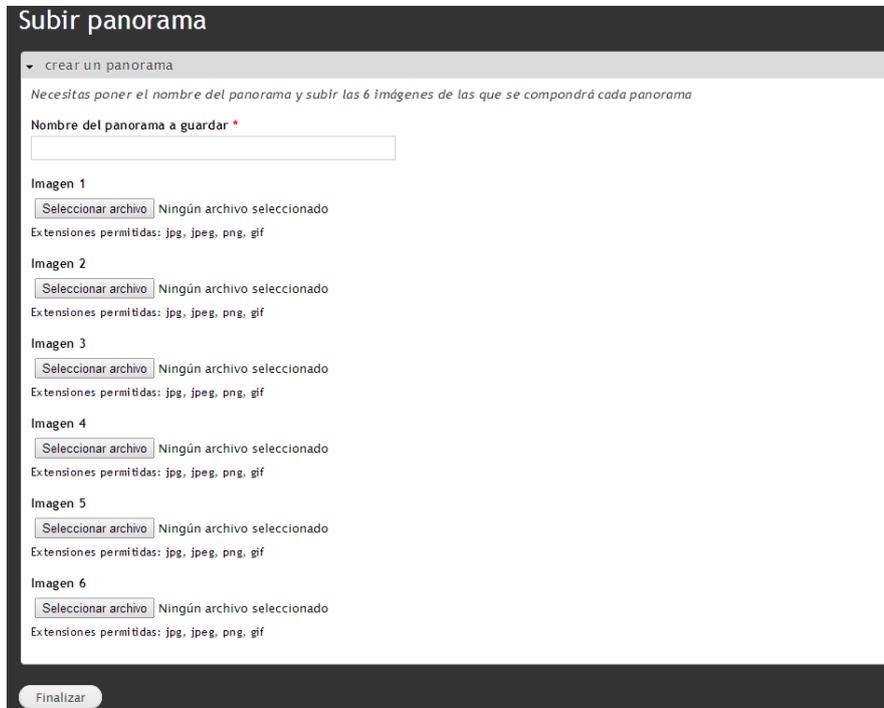


Figura 4.2.2.5 Se introducirá un nombre y se seleccionarán las imágenes

### 2. Asignar puntos de interés, representado en la Figura 4.2.2.6



Figura 4.2.2.6 Se selecciona el panorama al que se quiera aplicar los puntos de interés

En la Figura 4.2.2.7 se muestra las opciones que podemos realizar sobre el panorama escogido, haciendo hincapié en cada una de las opciones.

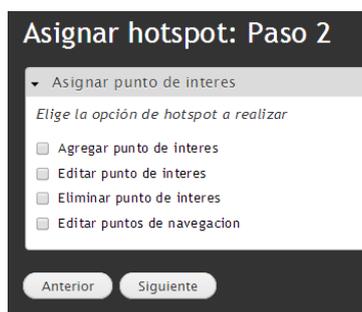


Figura 4.2.2.7 Se selecciona que opción queremos realizar

- **Agregar punto de interés**

En la Figura 4.2.2.8 se elige el tipo de punto de interés a crear (de información o de enlace), la imagen en la que se ubica dicho punto, la información asociada a él y un nombre que lo identifique. La posición en la que el punto se coloca en la foto se corresponde con los píxeles definidos en el eje X e Y.

Figura 4.2.2.8 Crear punto de interés

- **Editar punto de interés**

Si en la Figura 4.2.2.7 se selecciona la opción de editar se llega a la Figura 4.2.2.9. Directamente se muestran los puntos de interés asociados al panorama elegido previamente. Se producirá un error controlado si el panorama no tiene puntos de interés para editar notificando que primero hay que crearlos.

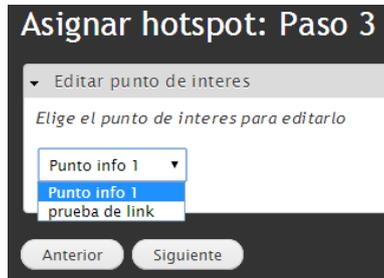


Figura 4.2.2.9 Elección del punto de interés a editar

Finalmente en la Figura 4.2.2.10 se editan los campos deseados.

Figura 4.2.2.10 Edición puntos de interés

- **Eliminar punto de interés**

Si en la Figura 4.2.2.7 se selecciona la opción de editar se llega a la Figura 4.2.2.11. Como en la opción de edición solo se muestran los puntos asociados al panorama. Si no tiene ningún punto de interés se produce un error controlado en el que se indica que primero hay que crear el punto antes de intentar eliminarlo.



Figura 4.2.2.11 Se elige el punto de interés que se desea eliminar

En la Figura 4.2.2.12 se muestra un mensaje que informa de que se va a suprimir el punto de interés



Figura 4.2.2.12 Pulsando finalizar se eliminará el punto seleccionado

- **Editar puntos de navegación**

Son aquellos puntos por los que se accederá a otro panorama enlazado al seleccionado. En la Figura 4.2.2.13 se puede ver los puntos de unión que se tiene con los otros panoramas. Si el panorama seleccionado previamente no tiene enlazado ninguno, no te deja continuar con el proceso.



Figura 4.2.2.13 Se selecciona el punto de navegación deseado

En la Figura 4.2.2.14 se completa la edición con los valores que se deseen

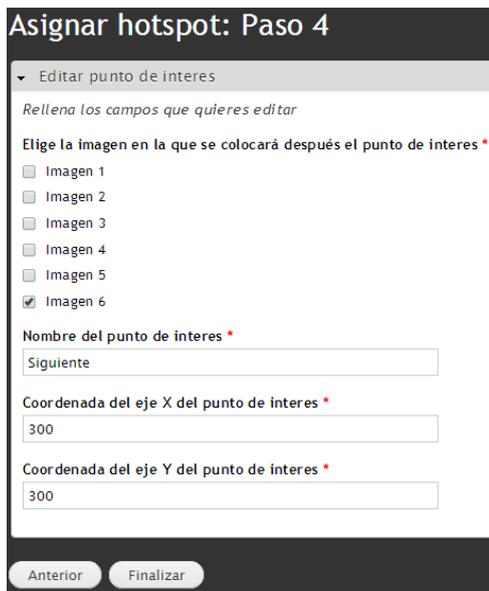


Figura 4.2.2.14 Panel de edición de un punto de interés

### 3. Eliminar panorama

Simplemente se selecciona el panorama y se pulsa finalizar, tal y como se muestra en la Figura 4.2.2.15.

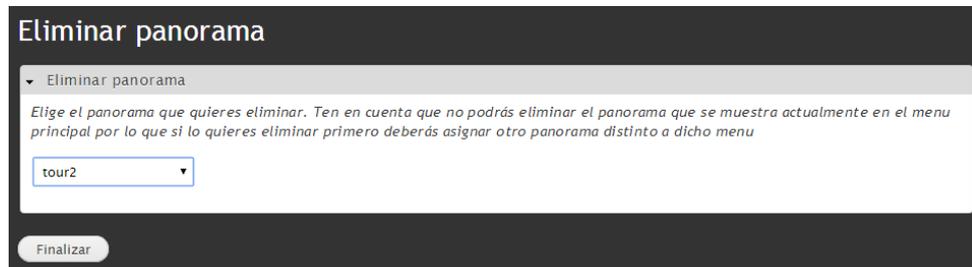


Figura 5.2.2.15 No te dejará por cuestión de coherencia eliminar el panorama que está actualmente activo en el menú, por lo que primero habrá que moverlo de allí.

### 4. Enlazar panorama

Lo primero que hay que hacer para llevar a cabo esta operación es seleccionar el panorama al que le quieres enlazar otro panorama, como se muestra en la Figura 4.2.2.16.

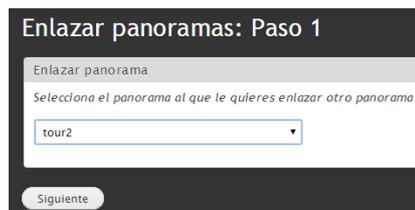


Figura 4.2.2.16 Se selecciona el panorama al que le quieres enlazar otro

En la Figura 4.2.2.17 se crea el punto de interés que actúa de enlace entre ambos panoramas

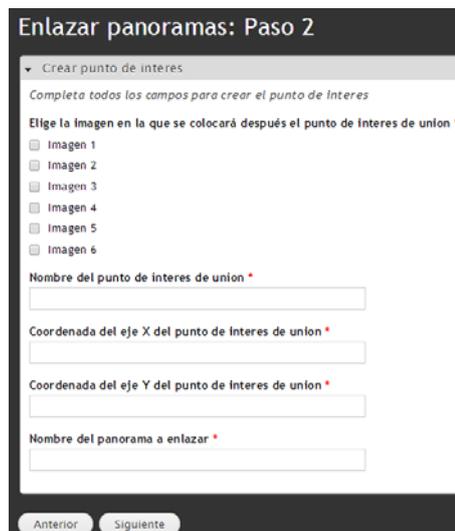


Figura 4.2.2.17 Se crea el punto de interés.

Por último se carga las imágenes que se corresponden con el nuevo panorama enlazado en el diálogo mostrado en la Figura 4.2.2.18.

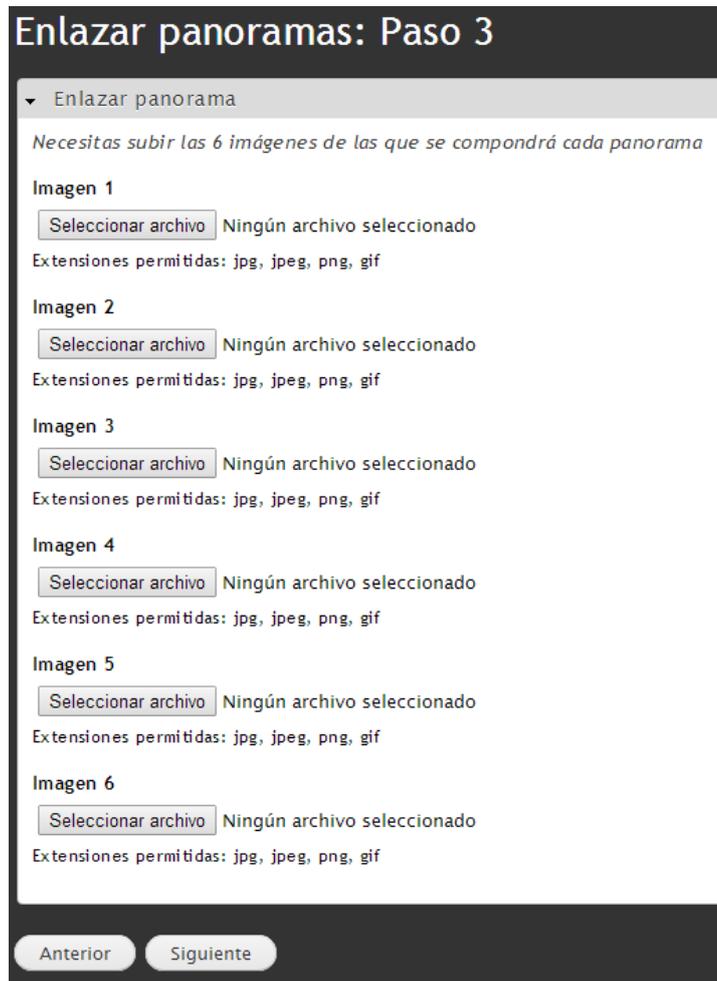


Figura 4.2.2.18 Finalmente se subirán las imágenes de las que se compondrá el nuevo panorama.

## 4.3 Views

Ya se ha hablado en el apartado de diseño de lo que son las views, por lo que ahora se procederá con lo que se ha implementado.

Las vistas utilizadas se corresponden con las mostradas en la Figura 4.3.1.

VIEW NAME	DESCRIPCIÓN	TAG	RUTA	OPERACIONES
Exposicion actual Mostrar: <i>Página</i> In database Tipo: Term		default	/exposicion-actual2	editar
Exposicion permanente Mostrar: <i>Página</i> In database Tipo: Term		default	/exposicion-permanente	editar
Exposiciones anteriores Mostrar: <i>Página</i> In database Tipo: Term		default	/exposiciones-anteriores	editar
Feeds log Mostrar: <i>Página</i> In code Tipo: Feeds log	Feeds log displays for overview, standalone importers and feed nodes.	Feeds	/import/%/log, /node/%/log, /admin/reports/feeds	editar
Listado de cuadros Mostrar: <i>Página</i> In database Tipo: Contenido		default	/listado-de-cuadros	editar
prueba-pano Mostrar: <i>Página</i> In database Tipo: Contenido		default	/prueba-pano	editar

Figura 4.3.1 Vistas utilizadas

Estas vistas se corresponden la gran mayoría de ellas con los elementos que se muestran en la Figura 4.3.2:



Figura 4.3.2 Menús de la aplicación web

A continuación se detalla cómo se han construido, y cuál es su propósito

### 4.3.1 Detalle de las vistas desarrolladas.

- **Vista exposición actual**

En la Figura 4.3.3 se muestra esta vista. En el apartado campos se introducen los datos que se quieren mostrar en la vista y su orden de muestra. Se observa que los campos coinciden con los detallados en el apartado taxonomía. En los filtros, se ha añadido que solo se muestre aquella exposición que sea visible, y que el campo de activa sea igual a 1.

The screenshot shows the configuration interface for a 'Page' display. The 'Page details' section is expanded, showing the following settings:

- TÍTULO:** Exposición actual
- FORMATO:** Unformatted list | Opciones
- Mostrar:** Fields | Opciones
- FIELDS:** A list of fields with 'agregar' buttons: Término de taxonomía: ID del término (ID del término), Término de taxonomía: Nombre, Global: Custom text, Término de taxonomía: Imagen exposición, Término de taxonomía: Información (Información), Término de taxonomía: Lugar de muestra (Lugar de muestra), Término de taxonomía: Período de muestra (Período de muestra)
- FILTER CRITERIA:** A list of criteria with 'agregar' buttons: Término de taxonomía: activa (= 1), Término de taxonomía: visibilidad (= Si)
- SORT CRITERIA:** A list of criteria with 'agregar' buttons.
- PAGE SETTINGS:** Ruta: /exposicion-actual2, Menú: Normal: Exposición actua..., Access: Permiso | Ver contenido publicado
- ENCABEZADO:** agregar
- PIE DE PÁGINA:** agregar
- PAGER:** Use pager: Full | Paged, 10 items, More link: No

Figura 4.3.3 Vista exposición actual

- **Vista exposición permanente**

En la Figura 4.3.4 se detalla cómo es la vista que se corresponde con la exposición permanente.

The screenshot shows the configuration interface for a 'Page' display. The 'Page details' section is expanded, showing the following settings:

- TÍTULO:** Exposición permanente
- FORMATO:** Unformatted list | Opciones
- Mostrar:** Fields | Opciones
- FIELDS:** A list of fields with 'agregar' buttons: Término de taxonomía: ID del término (ID del término), Término de taxonomía: Nombre, Global: Custom text, Término de taxonomía: Imagen exposición, Término de taxonomía: Información (Información), Término de taxonomía: Lugar de muestra (Lugar de muestra), Término de taxonomía: Período de muestra (Período de muestra)
- FILTER CRITERIA:** A list of criteria with 'agregar' buttons: Término de taxonomía: permanente (= 1), Término de taxonomía: visibilidad (= Si)
- SORT CRITERIA:** A list of criteria with 'agregar' buttons.
- PAGE SETTINGS:** Ruta: /exposicion-permanente, Menú: Normal: Exposición perma..., Access: Permiso | Ver contenido publicado
- ENCABEZADO:** agregar
- PIE DE PÁGINA:** agregar
- PAGER:** Use pager: Full | Paged, 10 items, More link: No

Figura 4.3.4 Vista exposición permanente

Esta vista se corresponde con la exposición permanente. Es igual que la vista anterior solo que en el apartado de filtros se utiliza la exposición que tenga activa el campo permanente a 1.

- **Listado de cuadros**

En la Figura 4.3.5 tenemos una vista intermedia. Su funcionalidad consiste en agrupar todos los cuadros que irán en exposiciones anteriores. Esto se logra en el apartado contextual filters, seleccionando el identificador de taxonomía (es decir, las exposiciones que están dentro de este vocabulario).

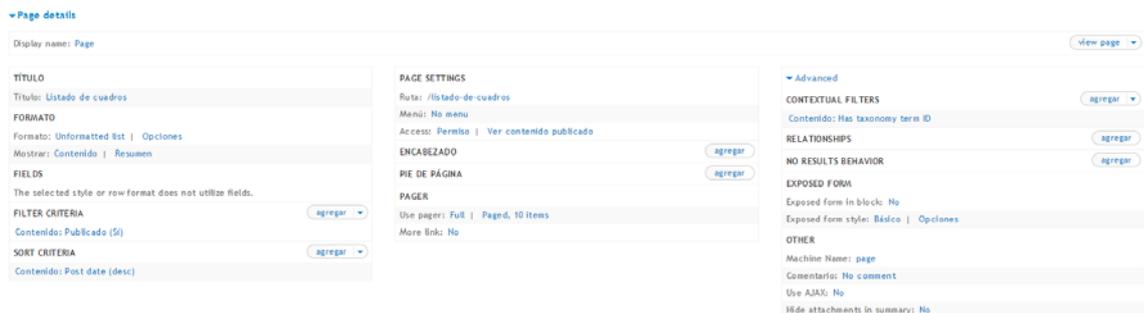


Figura 4.3.5 Vista intermedia

- **Exposiciones anteriores**

En la Figura 4.3.6 se detalla cómo es la vista que se corresponde con las exposiciones anteriores.

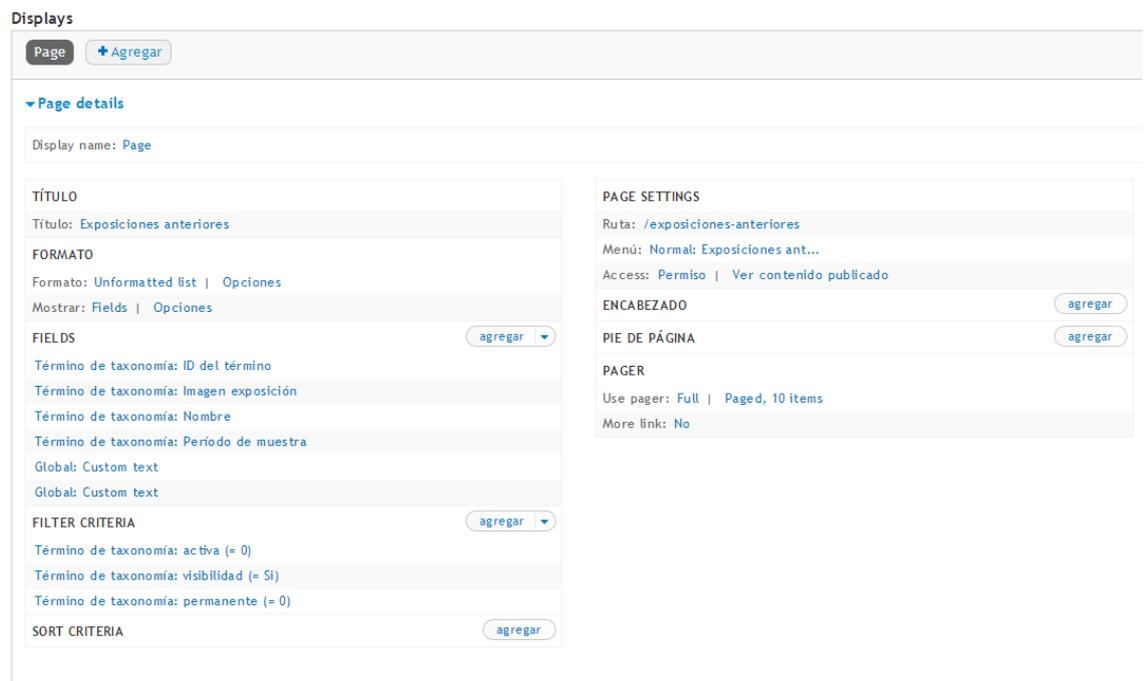


Figura 4.3.6 Exposiciones anteriores

Esta vista es parecida a las anteriores (solo que se aprovecha de la vista anterior descrita), solo que se mostrarán las exposiciones que no sean ni permanentes ni activas, eso sí, siempre que sean visibles.

- **Visita virtual**

En la Figura 4.3.7 se detalla cómo es la vista que se corresponde con la visita virtual.

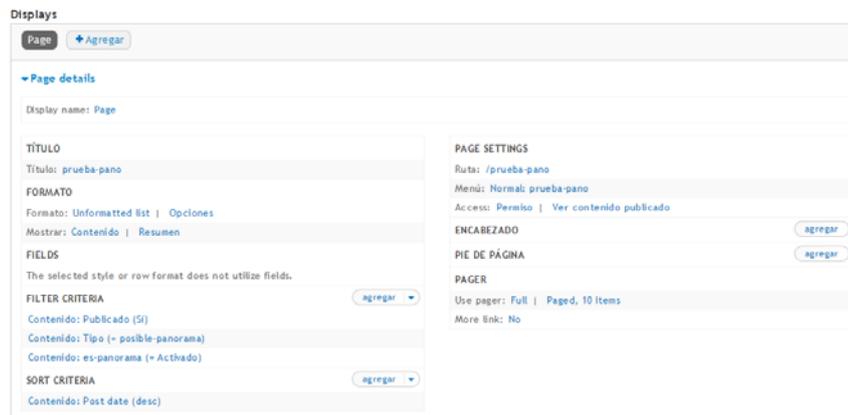


Figura 4.3.7 Vista visita virtual

Activará siempre el panorama cuyo valor en el campo es\_panorama sea igual a 1.

Esta vista es bastante distinta a las anteriores, ya que lo que se hace es crear un campo HTML en el que por medio de un iframe se redirecciona al archivo HTML que lanza el panorama, con este código:

```
<iframe width="800" height="600" src="panorama/leanorama/tour.html"></iframe>
```

### 4.3.2 Programas del gestor de contenidos que utilizan las vistas.

En este apartado se detallan las opciones de asignar exposiciones a los diversos menús y también de la asignación de la visita virtual.

- **Asignar exposición al menú: Exposición actual**

La Figura 4.3.8 representa el diagrama de flujo del programa que gestiona el poder colocar una exposición en el menú actual.



Figura 4.3.8 Diagrama de flujo del programa Asignar exposición actual

En la aplicación web simplemente hay que seleccionar la exposición deseada que se quiere mostrar en la pestaña exposición actual, tal y como se muestra en la Figura 4.3.9.



Asignar exposición

▼ Asignar exposición

*Elige la exposición que quieres que aparezca en la pestaña de menú: "Exposición actual", las otras pasarán automáticamente a estar en la pestaña de menú: "Exposiciones anteriores"*

Fábulas de Samaniego: La Naturaleza Humana -> Exposiciones ▼

Finalizar

Figura 4.3.9 Seleccionar exposición actual

- **Asignar exposición al menú: Exposición permanente**

En esta sección se omite el diagrama de flujo, puesto que sería idéntico al mostrado en la Figura 4.3.8, solo que hay que cambiar el campo field\_activa por el campo field\_permanente.

Por lo que se pasa a mostrar cómo se hace en la aplicación, ilustrado en la Figura 4.3.10



Asignar exposición permanente

▼ Asignar exposición

*Elige la exposición que quieres que aparezca en la pestaña de menú: "Exposición actual", las otras pasarán automáticamente a estar en la pestaña de menú: "Exposiciones anteriores"*

Fábulas de Samaniego: La Naturaleza Humana -> Exposiciones ▼

Finalizar

Figura 4.3.10 Se selecciona la exposición deseada. Al igual que en el caso anterior la exposición que está en permanente pasará a exposiciones anteriores.

- **Asignar visita virtual al menú: Visita virtual**

En la Figura 4.3.11 se muestra el diagrama de flujo que se corresponde con el archivo PHP programado para gestionar el menú de la visita virtual.

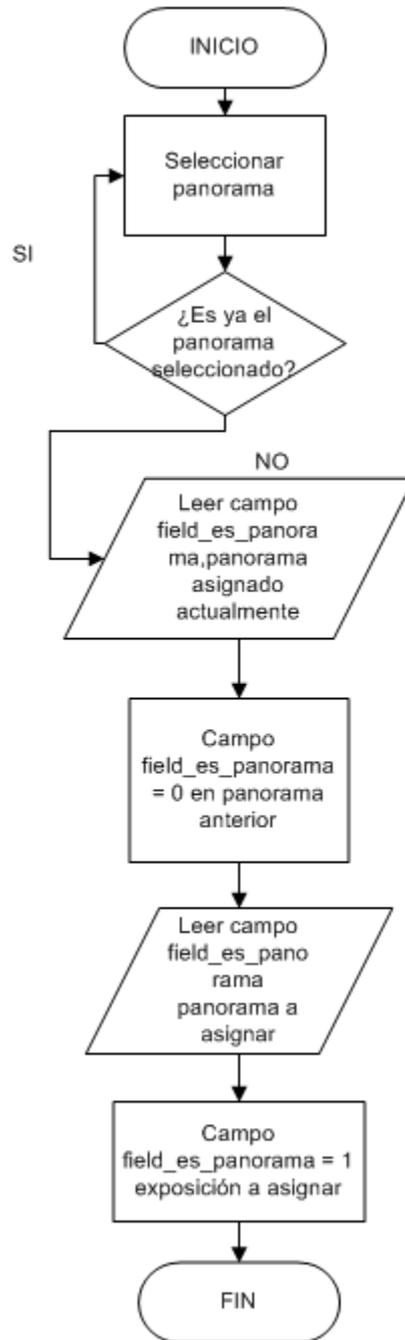


Figura 4.3.11 Diagrama de flujo de Asignar visita virtual

Ya en la aplicación solamente hay que indicar que panorama se quiere mostrar en la pestaña del menú, tal y como se muestra en la Figura 4.3.12.

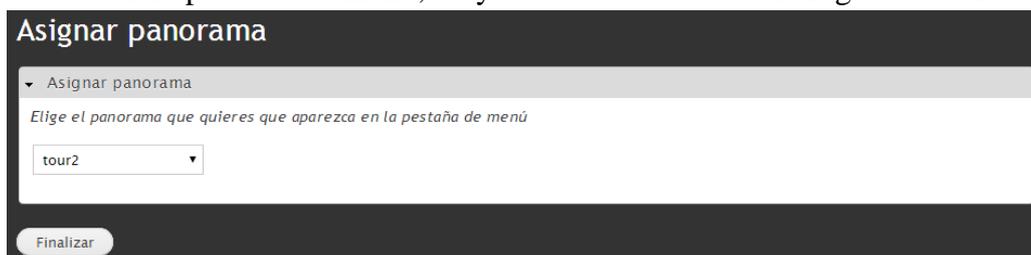


Figura 4.3.12 Al igual que en los casos anteriores, sólo hay que seleccionar la visita virtual que se desee mostrar. La que está activa actualmente se oculta automáticamente.

- **Visibilidad de las exposiciones**

En la Figura 4.3.13 se ve como se realiza la gestión de la visibilidad de las exposiciones. El archivo programado permite hacer multiselección, pudiendo seleccionar varias exposiciones a la vez para hacerlas visibles / invisibles.

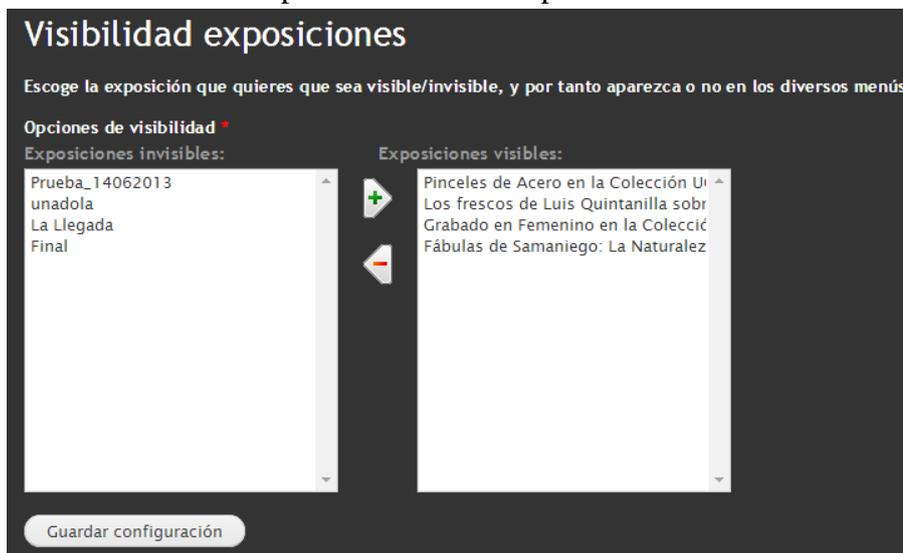


Figura 4.3.13 Desde aquí se podrán gestionar las exposiciones que quieran que se muestre.

### 4.3.3 Otros elementos del gestor de contenidos

En esta sección se exponen la opciones de obtener los códigos QR, y de exportar e importar las exposiciones.

- **Códigos QR**

En la Figura 4.3.14 se muestra el diagrama de flujo correspondiente al programa que gestiona los códigos QR.

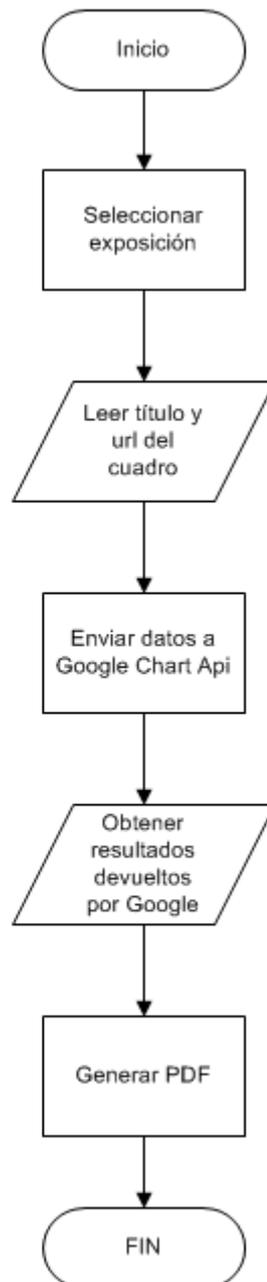


Figura 4.3.14 Diagrama de Flujo de los códigos QR

Dentro de la aplicación web tan sólo hay que seleccionar la exposición de la que se quieran obtener dichos códigos, tal y como se ve en la Figura 4.3.15.



Figura 4.3.15 Eligiendo la exposición deseada te descargará al ordenador un documento con todos los códigos QR pertenecientes a una exposición, mostrándotelos en formato recogido en la Figura 4.3.16.

La mariposa y el caracol



Figura 4.3.16, Código QR

- **Exportar exposición**

En la Figura 4.3.17 se muestra el diagrama de flujo correspondiente al programa Exportar exposición.

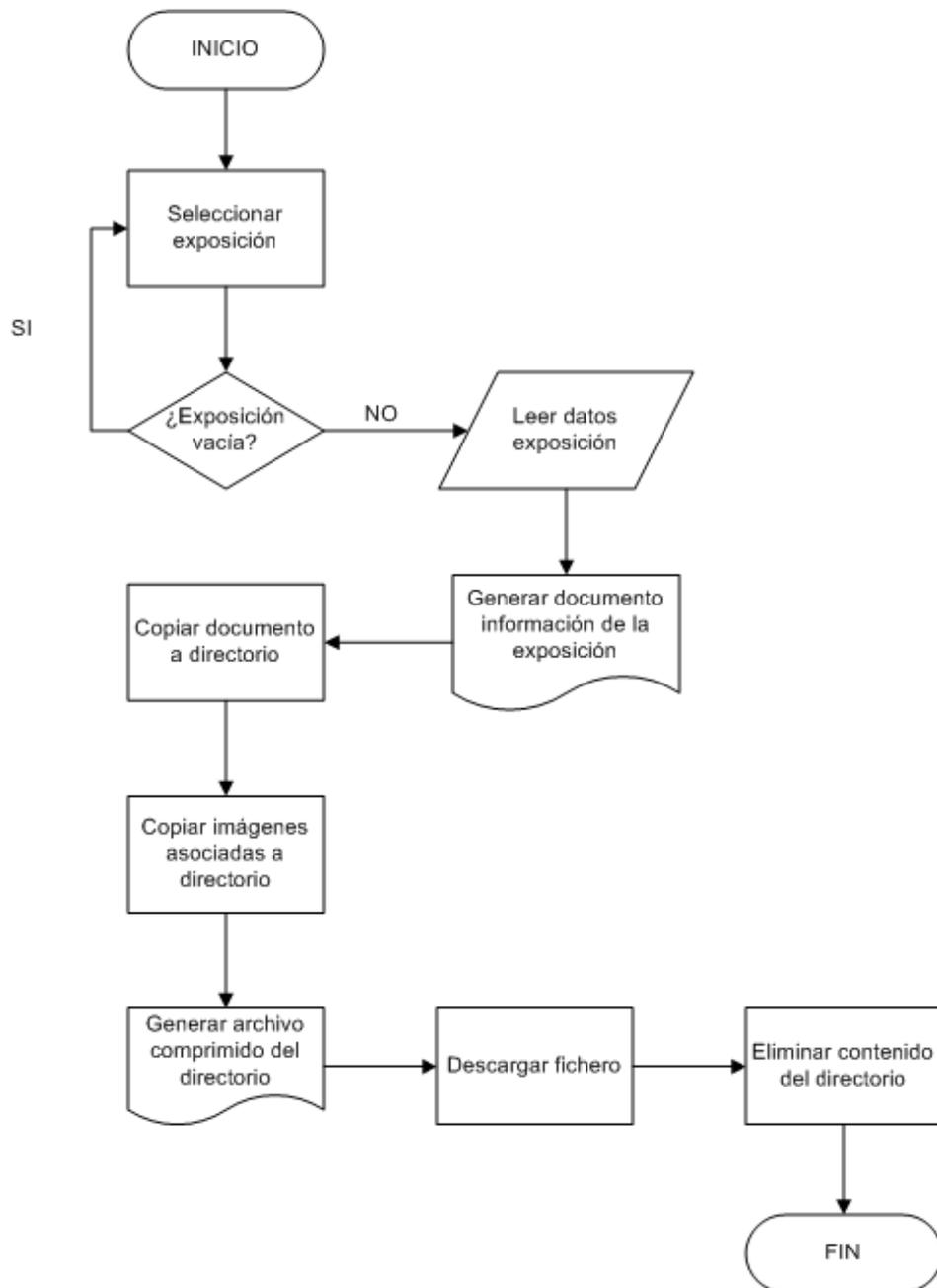


Figura 4.3.17 Diagrama de flujo del programa Exportar exposición

El funcionamiento dentro de la aplicación, tal y como se muestra en la figura 4.3.18, es de un uso extremadamente sencillo puesto que solo habrá que seleccionar la exposición y el gestor te descargará un archivo en formato .zip en el que estará toda la información de la exposición, tanto las imágenes como los datos.



Figura 4.3.18 Exportar exposición

- **Importar exposición**

En la Figura 4.3.19 se muestra el diagrama de flujo correspondiente al programa Exportar exposición.

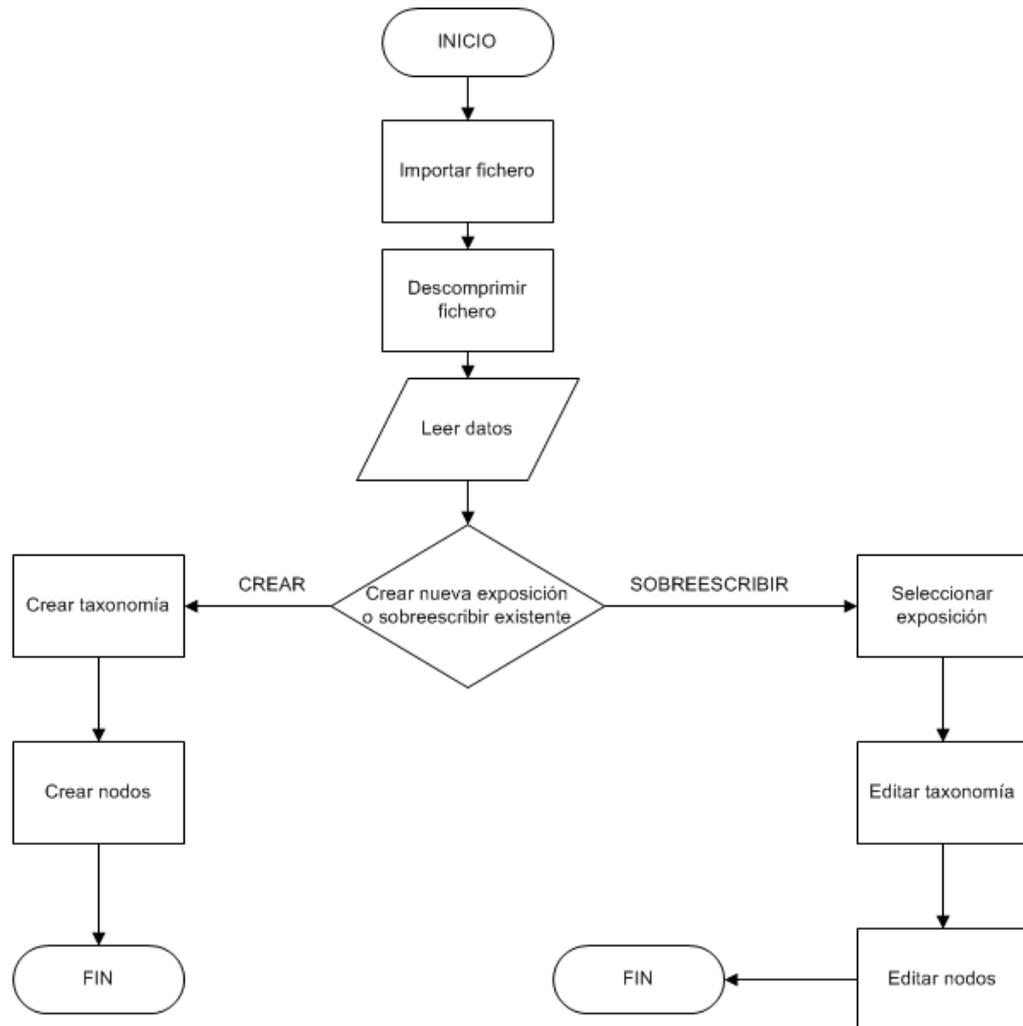


Figura 4.3.19 Diagrama de flujo del programa Importar exposición

Como se observa en la Figura 4.3.20 únicamente hay que seleccionar la ruta donde esté el archivo previamente importado. Al realizar esta opción se borran todos los datos que teníamos previamente sobre la exposición y se reemplazarán por los que hemos seleccionado.



Figura 4.3.20 Importar Exposición

## 5. Conclusiones y trabajos futuros.

A continuación se exponen las principales conclusiones obtenidas a lo largo de este Proyecto, así como algunos de los trabajos que podrían llevarse a cabo en futuros desarrollos.

### 5.1 Conclusiones

Hoy en día existen muchas facilidades para proporcionar la información a los distintos consumidores desde la irrupción de Internet y de los dispositivos móviles. Todos ellos quieren saber qué se van a encontrar a la hora de ir a cualquier sitio, y de esta realidad surge la necesidad del Área de Exposiciones de la Universidad de Cantabria de presentar de una forma visual todos los contenidos mostrados en sus exposiciones.

En esa línea, este Proyecto ha facilitado una herramienta que permite la creación, edición y manipulación de todos los contenidos de las exposiciones realizadas, de una forma sencilla para los administradores de la herramienta, que con un escaso conocimiento informático serán capaces de gestionar todos los contenidos.

Dentro del paradigma de la accesibilidad se han estudiado diferentes mecanismos de acceso a la información, como es el uso de etiquetas inteligentes, accesibles directamente desde cualquier dispositivo móvil. Una primera solución se aporta en forma de etiquetas QR mediante las cuales cualquier visitante del Área de Exposiciones puede obtener la información completa de cada obra expuesta escaneando la etiqueta con su dispositivo inteligente (Smartphone).

Desde el punto de vista del Área de Exposiciones de la Universidad de Cantabria, este Proyecto supone un gran avance para la difusión de todo lo que se expone en la Sala de Exposiciones. Por otra parte, la gestión de contenidos incluye la posibilidad de crear un catálogo con el histórico de exposiciones, facilitando de forma simple un repositorio permanente (y exportable) que hasta el momento debía realizarse de forma manual.

Dentro del apartado de contenidos multimedia, se ha implementado un facilidad para la creación, edición y publicación de las denominadas visitas virtuales, con las cuales desde el navegador se puede ver cómo es la exposición sin necesidad de que el espectador esté allí presente.

Por último, cabe destacar que toda la aplicación ha sido realizada en su totalidad con herramientas opensource, lo cual demuestra que el uso de este tipo de código no está reñido con el desarrollo de aplicaciones avanzadas del mismo estilo que las aplicaciones comerciales, en este caso para una aplicación específica como es la gestión de contenidos artísticos de una forma intuitiva y visual.

## 5.2 Trabajos futuros.

Cualquier herramienta de esta índole está sujeta a modificaciones que mejoren el rendimiento de la misma con el paso del tiempo. Cabe destacar las siguientes razones que podrían motivar el replanteamiento de alguna parte de la aplicación:

- El primer gran cambio resulta directamente relacionado con el CMS. Las actualizaciones de la versión de Drupal pueden conllevar la modificación de alguna parte del código desarrollado.
- De la misma forma, el uso de código abierto está sujeto a continuas actualizaciones por parte de los usuarios/desarrolladores, lo que en ocasiones provoca revisiones sobre los códigos ya desplegados.
- Además, el uso de tecnologías avanzadas como son las utilizadas por los dispositivos móviles también provoca una gran dependencia a la hora de que una aplicación pueda soportar funcionalidades que no estaban accesibles en el momento de su desarrollo.

Con todo, el desarrollo planteado pretende ser lo más genérico y portable posible, por lo que las actualizaciones de sistemas y dispositivos no debieran de suponer una degradación del comportamiento de la aplicación. Sin embargo, si que existen algunas líneas de trabajo que podrían ser abordadas en un futuro Proyecto:

- El uso de etiquetas inteligentes solamente ha sido implementado en forma de etiquetas pasivas (QR). Una primera mejora a introducir sería aprovechar los dispositivos NFC [22] incluidos en los nuevos smartphones, pudiendo por ejemplo, mediante una etiqueta colocada debajo del cuadro en la sala, acercar solamente el teléfono inteligente y obtener la información de la obra mostrada en la aplicación web sin necesidad de estar conectado a Internet.
- El desarrollo de visitas virtuales supone la creación de un “entorno virtual” que no siempre es fácil de implementar. Una vía de mejora sería la sustitución de las actuales “fotos en cubo” en verdaderos “mundos virtuales” del estilo, por ejemplo, de los desarrollados en sistemas muy bien conocidos, como es el caso de Google Maps[23] y Google Earth [24], con edificios completamente digitalizados.

## 6. Bibliografía.

- Thomson, L., & Welling, L. (2005). "Desarrollo web con PHP y MySQL." Anaya.
- [1] Proceso Racional Unificado (PRU): <http://es.slideshare.net/siudy/proceso-unificado-6715119>
- [2] PHP(Hypertext PreProcessor): <http://php.net/>
- [3] Apache: <http://httpd.apache.org/>
- [4] MySQL: <http://www.mysql.com/>
- [5] PHPMyAdmin: [http://www.phpmyadmin.net/home\\_page/index.php](http://www.phpmyadmin.net/home_page/index.php)
- [6] HTML (HyperText Markup Language): <http://www.w3schools.com/html/>
- [7] CSS: <http://www.w3schools.com/css/>
- [8] Firebug: <https://addons.mozilla.org/es/firefox/addon/firebug/>
- [9] NotePad++: <http://notepad-plus-plus.org/>
- [10] WAMP server: <http://www.wampserver.com/en/>
- [11] LAMP server: <http://lamphowto.com/>
- [12] Drupal: <https://www.drupal.org/>
- [13] Joomla: <http://www.joomla.org/>
- [14] WordPress: <https://es.wordpress.com/>
- [15] CSV: <https://docs.python.org/2/library/csv.html>
- [16] Flash: <http://get.adobe.com/es/flashplayer/>
- [17] QR: <http://www.prepressure.com/library/technology/qr-code>
- [18] doomPDF: <http://www.prepressure.com/library/technology/qr-code>
- [19] PDF: <http://www.adobe.com/es/products/acrobat/adobepdf.html>
- [20] Google Api: <https://developers.google.com/apis-explorer/#p/>
- [21]Form-API:  
[https://api.drupal.org/api/drupal/developer!topics!forms\\_api\\_reference.html/7](https://api.drupal.org/api/drupal/developer!topics!forms_api_reference.html/7)
- [22] NFC: <http://andro4all.com/2013/12/que-es-el-nfc-y-utilizarlo-en-android>
- [23] Google Maps: <https://maps.google.es/>
- [24] Google Earth: <https://earth.google.es/>