



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

**Desarrollo de un Portal Informativo y Divulgativo del Patrimonio
Artístico de la Ciudad Universitaria de Caracas (CUC).
Módulo: Paseo Virtual de la Plaza Cubierta del Rectorado y Obras de
la Síntesis de las Artes Mayores.**

**Trabajo Especial de Grado presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres Jorge E. Gutiérrez Roa C.I. 18.491.643
y Belkis Z. Chacón Pacheco C.I. 13.693.175
para optar al título de Licenciado en Computación
Tutora: Profa. Yusneyi Y. Carballo Barrera**

Ciudad Universitaria de Caracas, 24 Octubre de 2013

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR – CENEAC

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **Desarrollo de un Portal Informativo y Divulgativo del Patrimonio Artístico de la Ciudad Universitaria de Caracas (CUC). Módulo: Paseo Virtual de la Plaza Cubierta del Rectorado y Obras de la Síntesis de las Artes Mayores**, presentado por los Bachilleres **Jorge E. Gutiérrez Roa (C.I. 18.491.643)** y **Belkis Z. Chacón Pacheco (C.I. 13.693.175)**, a los fines de optar al título de **Licenciado en Computación**, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 24 de Octubre de 2013, a las 5:30pm horas, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala de Postgrado de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondió a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar el trabajo con la nota de _____ puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 24 de Octubre de 2013.

Profa. Yusneyi Carballo Barrera
(Tutora)

Profa. Mariantonia Palacios
(Jurado)

Prof. Robinson Rivas Suárez
(Jurado)

AGRADECIMIENTOS

Llegado este momento de reflexión y pensamiento, me es muy grato recordar a todos aquellos que en los más difíciles momentos han estado a mi lado, me han acompañado y me han impulsado a seguir y luchar por alcanzar mis metas.

Ante todo quiero agradecer a Dios, todopoderoso, por hacer que esté aquí, por entregarme a un hogar maravilloso y rodearme de personas verdaderamente valiosas. A todos los ángeles por estar a mi lado, cuidarme y guiarme.

A mis padres, Isabel Roa y Juan Gutiérrez, siempre complacientes, en errores y aciertos, quienes me educaron, me formaron y me dieron la posibilidad de llegar hasta donde estoy hoy, llevando el pan a la mesa de nuestro hogar y luchando para sacarlo adelante.

A mi abuela, Josefa Díaz, formadora de valores, por ser una verdadera maestra de la vida, por sus lecciones, por su cariño inalienable.

A mi tutora, Yusneyi Carballo, quien tuve la dicha de conocer desde mi primer semestre en esta ilustre casa de estudios y en quien hallé un docente de vocación, comprometida con su labor.

A la Profa. Mariantonia Palacios, de quien he podido revivir un compromiso que no sentía desde mis inicios como universitario, un compromiso con la Ciudad Universitaria de Caracas y su valor como patrimonio para todos y cada uno de nosotros.

A la Lic. Nila Sareet, de la COPRED, por su compromiso con este desarrollo, quien aportó material invaluable para la culminación del mismo, y se mostró siempre presta a colaborar.

A Ronald Morey, amigo y compañero de estudio, por sus excelentes recomendaciones sobre el desarrollo e implementación de este proyecto.

A mi amiga y compañera de proyecto, Belkis Chacón, por permitir la culminación de este T.E.G.

A todos y cada uno de ustedes, amigos y faltantes, vaya de mi parte mi más valiosa muestra de agradecimiento.

Jorge E. Gutiérrez Roa

AGRADECIMIENTOS

A Dios y mis ángeles guardianes, quienes siempre me han acompañado en este largo recorrido.

A mis padres, Zenón Chacón Guerrero y María Margarita Pacheco de Chacón, mis pilares fundamentales por haberme guiado por el mejor camino, por estar siempre presente en los buenos y malos momentos.

A mis hermanos, Marcos, Carlos, y Sonia, por su apoyo, comprensión y amor.

A mis sobrinos Carol, Jessica, Markielis, Jean Franco, Hanna, Gabriel, Andrés para quienes he tratado de ser el mejor ejemplo a seguir, les brindo todo mi apoyo y amor.

Al Sr. Alfredo Schmeichler por ser un gran hombre, por su ayuda, apoyo y enseñanza que están siempre presente.

A mi amiga y hermana del alma Carolina Tineo por todos estos años compartidos.

A mi amigo y compañero de tesis, Jorge Gutiérrez, gracias por siempre estar conmigo por tu paciencia, comprensión y escucharme en todo momento.

A mis amigas Brenda López y Araceli Mayora Lombano por bríndame su apoyo y darme siempre un consejo acertado.

A mis amigos de estudios Neisy Quintana, Alexis Colmenares, Kevin Delgado, Carlos Méndez, Héctor Blandin.

A nuestra Tutora, Yusneyi Carballo Barrera por permitirnos trabajar bajo su tutela además de ser gratificante es un honor, estoy muy agradecida con usted por indicarnos siempre el mejor camino a seguir y prestarnos toda su colaboración para poder finalizar este proceso.

A la Profesora de la Escuela de Artes Mariantonia Palacios por su valioso apoyo y colaboración.

A todos los mencionados y faltantes, mis más sinceras gracias.

Belkis Z. Chacón Pacheco



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN
CENTRO DE ENSEÑANZA ASISTIDA POR COMPUTADOR - CENEAC

Desarrollo de un Portal Informativo y Divulgativo del Patrimonio Artístico de la Ciudad Universitaria de Caracas (CUC). Módulo: Paseo Virtual de la Plaza Cubierta del Rectorado y Obras de la Síntesis de las Artes Mayores.

Autores: Jorge Eduardo Gutiérrez Roa

Belkis Zulay Chacón Pacheco

C.I: 18.491.643

C.I: 13.693.175

Correo: jorgegutierrezr@gmail.com

Correo: belkisch@gmail.com

Tutora: Profa. Yusneyi Carballo Barrera

Fecha: Ciudad Universitaria de Caracas, 24 de Octubre de 2013

RESUMEN

El presente Trabajo Especial de Grado se enfocó en el desarrollo de un módulo que permita visualizar las obras de artes y los espacios arquitectónicos de la Ciudad Universitaria de Caracas, así como obtener información de los mismos y de sus autores. Este módulo forma parte de un portal dedicado a la divulgación del patrimonio artístico, denominado “El Arte en la Ciudad Universitaria de Caracas”. Particularmente en este Trabajo Especial de Grado se han incluido el área y las obras pertenecientes al proyecto Síntesis de las Artes Mayores de Carlos Raúl Villanueva que se encuentran en la Plaza Cubierta del Rectorado.

Este portal web está dirigido a los usuarios con interés en conocer y recorrer mediante una aplicación virtual el acervo artístico de los espacios de la Plaza Cubierta del Rectorado de la Ciudad Universitaria de Caracas en donde se fusiona el desarrollo de las artes y del urbanismo. Se deja abierta la posibilidad de completar los paseos virtuales por las otras zonas de la Ciudad Universitaria de Caracas. Para el desarrollo de este portal se combinaron las tecnologías PHP (lenguaje de programación), MySQL (Sistema Manejador de Bases de Datos), *CakePHP* (*Framework*) y la metodología FDD (*Feature Driven Development*), además de varias librerías de JQuery para el manejo de imágenes y las panorámicas del paseo virtual.

Palabras Clave: paseo virtual, museo virtual, visita virtual, Ciudad Universitaria de Caracas, obras de arte, pintura, escultura, mosaicos, Síntesis de las Artes Mayores, Carlos Raúl Villanueva, Universidad Central de Venezuela, CUC Patrimonio Mundial.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I.....	4
<i>PLANTEAMIENTO DEL PROBLEMA</i>	<i>4</i>
1.1. Situación actual.....	4
1.2. Justificación e importancia	5
1.3. Objetivo general del TEG	6
1.4. Objetivos específicos	6
1.5. Alcance de la aplicación.....	6
1.6. Potenciales usuarios	7
CAPÍTULO II	8
<i>MARCO TEÓRICO.....</i>	<i>8</i>
2.1 Visita virtual	8
2.2 Evaluación de algunos paseos o museos virtuales	9
2.3 Evaluación de museos virtuales con colecciones y galerías de imágenes	11
2.4 Herramientas para crear paseos virtuales y galería de imágenes	17
2.5 Formato de almacenamiento para imágenes.....	19
2.6 Fotos panorámicas.....	20
2.7 Herramientas para el diseño gráfico digital.....	22
2.8 Patrón de arquitectura	24
2.9 Tecnologías para el desarrollo del portal	25
2.10 Metodología utilizada	27
CAPÍTULO III.....	30
<i>MARCO APLICATIVO.....</i>	<i>30</i>
3.1 Desarrollo de un modelo global	30
3.2 Lista de funcionalidades	32
3.3 Perfiles de usuarios.....	36
3.4 Planeación por funcionalidad	38
3.5 Diseño de la base de datos	39
3.6 Diseño por funcionalidad.....	42
3.7 Construcción por funcionalidad.....	50
3.8 Interfaz del portal web	69
3.9 Pruebas y resultados.....	77
CONCLUSIONES Y RECOMENDACIONES.....	80
4.1 Conclusiones.....	80
4.2 Recomendaciones y Trabajos Futuros	81
BIBLIOGRAFÍA	82
ANEXOS	86
<i>ANEXO A: CASOS DE USOS.....</i>	<i>86</i>
<i>ANEXO B: TEST DE ACEPTACIÓN</i>	<i>90</i>
<i>ANEXO C: MANUAL DE USUARIO.....</i>	<i>94</i>

ÍNDICE DE FIGURAS

Figura 1. Vista de la Colección <i>The Metropolitan Museum of Art</i>	11
Figura 2. Vista de la Colección del Museo Nacional de Prado	12
Figura 3. Vista de la Colección de <i>National Gallery of Art</i>	13
Figura 4. Vista de las Colecciones de Google Art Project	15
Figura 5. Vista de las Obras por Artista	16
Figura 6. Vista de todas las Obras por un Artista con Breve Reseña	16
Figura 7. Vista de la Obra por un Artista con más Información	17
Figura 8. Foto Clásica o de Papel	21
Figura 9. Foto Panorámica	21
Figura 10. Foto Cilíndrica	21
Figura 11. Foto Esférica	21
Figura 12. Proceso de Desarrollo Basado en Funcionalidades (FDD)	29
Figura 13. Diagrama de Clases del Portal	31
Figura 14. Modelo de Datos del Portal Web	41
Figura 15. Diagrama de Secuencia: “Ficha Informativa” de los Elementos	42
Figura 16. Diagrama de Secuencia: “Ficha de Autor”	43
Figura 17. Diagrama de Secuencia: Paseo Virtual	43
Figura 18. Diagrama de Secuencia: Mapa y sus elementos	44
Figura 19. Diagrama de Secuencia: Conjunto Arquitectónico	44
Figura 20. Diagrama de Secuencia: Galería de Imágenes con Ordenamiento	45
Figura 21. Diagrama de Secuencia: Listar	46
Figura 22. Diagrama de Secuencia: Ver Detalle	46
Figura 23. Diagrama de Secuencia: Eliminar	47
Figura 24. Diagrama de Secuencia Editar: Autor, zona, tipo de elemento y usuario	48
Figura 25. Diagrama de Secuencia Editar: Panorámica y elemento	48
Figura 26. Diagrama de Secuencia Agregar: Autor, zona, tipo de elemento y usuario	49
Figura 27. Diagrama de Secuencia Agregar: Panorámica y elemento	49
Figura 28. Función AJAX que Realiza la Carga Dinámica	51
Figura 29. Código de obtención de campos y rutas del elemento en PHP	52

Figure 30. Código de obtención de datos del elemento en PHP	53
Figure 31. Instrucción que utiliza panorama()	54
Figura 32. Función advancedPanorama().....	55
Figure 33. Fragmento de función de carga	55
Figure 34. Obtención de datos que utilizará las funciones de <i>Google Maps</i>	56
Figura 35. Querys utilizados en la consulta de base de datos	58
Figura 36. Construcción del nombre del objeto.....	62
Figura 37. Validar asociación de imágenes	62
Figura 38. Impresión Mediante un Recorrido	64
Figura 39. Código de Detección de Coordenadas	65
Figura 40. Función de Agregar Enlace	66
Figura 41. Función de Remover Enlace	66
Figura 42. Funciones de Autenticación	67
Figura 43. Funciones de Eliminar	69
Figura 44. Interfaz de Página de Inicio	70
Figura 45. Interfaz de ¿Por Qué Somos Patrimonio?	72
Figura 46. Interfaz de Conjunto.....	73
Figura 47. Interfaz de Paseo Virtual	74
Figura 48. Interfaz de Galería	75
Figura 49. Interfaz de Administración	76
Figura 50. Interfaz de Listar Autores	77
Figure 51. Diagrama de caso de uso para el usuario con perfil visitante.....	87
Figura 52. Diagrama de caso de uso para el usuario con perfil administrador.....	88
Figura 52. Diagrama de caso de uso para el usuario con perfil registrado	89

ÍNDICE DE TABLAS

Tabla 1. Cuadro Comparativo de Aplicaciones Web con Paseo Virtual	10
Tabla 2. Características Significativas de los Formatos de Imagen	19
Tabla 3. Tipos de Fotografías	21
Tabla 4. Comparación entre Programas para el Tratamiento de Fotos	22
Tabla 7. Funcionalidades y Permisos de los Usuarios del Portal.....	37
Tabla 5. Lista de actividades.....	38
Tabla 6. Descripción de la Base de Datos y las Relaciones.....	40
Tabla 8. Casos de usos: Resumen y actores	86

INTRODUCCIÓN

Dar a conocer el patrimonio cultural o natural es una de las vías más importantes para protegerlo y conservarlo, además de lograr que las personas conozcan más el entorno en el que viven y se desarrollan. El objetivo no es la simple transmisión de información, pues a través de la difusión, se pretende que cada persona, de forma individual y en conjunto con la sociedad, unifique actitudes para transformar su comportamiento con respecto al patrimonio cultural.

El avance tecnológico permite la creación de espacios virtuales, en donde la sociedad puede disfrutar de sus sitios históricos, posibilitando así que un número mayor de personas acceda a esta riqueza. Una de las formas para masificar la riqueza cultural e incluso natural, es a través de los denominados paseos virtuales, los cuales, en los últimos tiempos, están alcanzando un desarrollo extraordinario.

Actualmente la Ciudad Universitaria de Caracas (CUC) no tiene un espacio virtual en donde pueda divulgarse en forma integral su valor patrimonial, aunque existen galerías y reseñas digitales. Por ello, para conocer sus instalaciones emblemáticas, se tiene que visitar la CUC de forma física. Por tal motivo, surge la idea de crear un portal que permita, a través de paseos virtuales, galerías y contenido informativo, conocer la riqueza de las obras de arte de nuestro recinto universitario, incluyendo en este TEG la zona de la Plaza Cubierta del Rectorado y obras del proyecto “Síntesis de las Artes Mayores” del arquitecto Carlos Raúl Villanueva.

El objetivo fundamental de este Trabajo Especial de Grado es el desarrollo de la investigación documental, de antecedentes, de aspectos metodológicos, técnicos e informáticos que permitan la construcción del portal “El Arte en la Ciudad Universitaria de Caracas”, orientado a informar y dar acceso a información sobre las principales obras de arte de la Ciudad Universitaria de Caracas a través de un paseo virtual. Es una que responde a la necesidad expresada de masificar el acceso al Patrimonio de la CUC a través de la web, permitiendo su

recorrido remoto o en forma virtual y con la ayuda de mapas digitales, lo cual permitirá conocer nuestra UCV sin necesidad de visitar el recinto físicamente.

Para dar cumplimiento a los requerimientos identificados en el trabajo de seminario, se desarrolló un portal que combina las tecnologías PHP (lenguaje de programación), MySQL (Sistema Manejador de Bases de Datos), *CakePHP (Framework)* y la metodología FDD (*Feature Driven Development*), además del API de Google Map con el uso de georeferenciación y varias librerías de JQuery para el manejo de imágenes y las panorámicas del paseo virtual.

A continuación, se dará una visión general de cada uno de los capítulos que estructuran este documento:

En el capítulo I, se describe la situación actual, justificación e importancia del tema de este desarrollo; se expone el objetivo general y los objetivos específicos del Trabajo Especial de Grado y se presenta el alcance de la aplicación y los usuarios potenciales.

En el capítulo II, se presentan la definición de paseo virtual, la evaluación de diversos paseos virtuales y museos en donde se puedan extraer características para considerar su pertinencia en el desarrollo de la TEG, se revisan las diversas herramientas para crear paseos virtuales, galerías y los diversos formatos de almacenamiento más adecuados para el portal, tipos de fotografía y posibles alternativas de *software* para el tratamiento de las mismas, herramientas de diseño gráfico digital, patrón de arquitectura, tecnologías para el desarrollo del portal web y la metodología utilizada.

En el capítulo III, se realizará el desarrollo del modelo general en función de los requerimientos, el planeamiento y agrupación por funcionalidad, el diseño de la base de datos, diseño y construcción por funcionalidades, se muestran las diversas interfaces que conforman el portal además de las pruebas funcionales y de aceptación por parte de los usuarios.

Finalmente, se presentan las conclusiones del trabajo realizado, recomendaciones, las fuentes consultadas y, como información complementaria (anexos), se muestra los diagramas de casos

de uso desarrollados en la investigación del seminario y el modelo de cuestionario para la realización de las pruebas de la aplicación y un manual de usuario.

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

En este capítulo se presenta la situación actual, justificación e importancia de crear un portal que permita divulgar información sobre nuestro acervo artístico a través de las galerías de imágenes y de un paseo virtual, el objetivo general del Trabajo Especial de Grado, sus objetivos específicos, el alcance de la aplicación y los usuarios potenciales.

1.1. Situación actual

La Ciudad Universitaria de Caracas constituye un auténtico museo vivo y dinámico de arte del siglo XX; pero quizás, este mismo dinamismo lo hace pasar inadvertido para muchos de sus visitantes y, sorprendentemente, también para muchos de los miembros de la comunidad universitaria. Los usuarios se han acostumbrado tanto a sus espacios y a las obras que los rodean, que olvidan a menudo la riqueza de este patrimonio; además, las obras de arte de la Ciudad Universitaria de Caracas no son un mero agregado decorativo, ni han sido incorporadas como un artículo de lujo o de ostentación, ellas responden a un intención de integración.

Existe una gran diversidad de contenidos bibliográficos sobre las obras de arte de la Ciudad Universitaria de Caracas. Sin embargo, los recursos digitales son menos numerosos y diversos. Dentro de éstos se consiguen trabajos como el realizado para el centenario de Villanueva, con un extenso contenido multimedia sobre la Ciudad Universitaria de Caracas. Este trabajo es accesible a través de Internet por medio de su página web¹.

Adicionalmente, el portal web de la Facultad de Arquitectura de la Universidad Central de Venezuela² ofrece un contenido fotográfico de la Ciudad Universitaria de Caracas, acompañado de algunas descripciones.

¹ <http://www.centenariovillanueva.web.ve>

² <http://www.fau.ucv.ve/ciudaduniversitaria.htm>

Por último, el portal de la Universidad Central de Venezuela³ destaca la declaración de la Ciudad Universitaria de Caracas como Patrimonio por la UNESCO y añade una galería que incluye 108 obras de arte inscritas en el proyecto de “Síntesis de la Artes”, en donde también se pueden conseguir infografías, algunos mapas e información sobre recorridos, pero no son portales dinámicos.

1.2. Justificación e importancia

Cuando a un espacio se le da la distinción de Patrimonio Cultural de la Humanidad, quien regenta este espacio tiene la obligación de conservarlo. Por tal motivo investigadores del Consejo de Preservación y Desarrollo (COPRED) de la UCV y docentes de la Escuela de Artes de la Facultad de Humanidades y Educación, tienen la iniciativa de proponer a docentes y estudiantes de la Escuela de Computación de la Facultad de Ciencias, un trabajo en conjunto para la creación de un portal en donde se pueda divulgar información sobre el acervo artístico, a través de galerías e implementando un paseo virtual que permita recorrer la Ciudad Universitaria de Caracas como primer módulo, centrándose en la Plaza Cubierta del Rectorado. La principal función que tendrá el portal informativo es mostrar la diversidad de obras de arte existentes en su espacio arquitectónico, aportando facilidad de uso e interactividad con el usuario. Con este portal se quiere contribuir a la valoración, difusión del legado artístico y creación de conciencia para su preservación.

Las visitas o paseos virtuales representan una forma moderna de mostrar, a los posibles visitantes de una estructura física, un espacio concreto, con el fin de que se exponga, virtualmente, una visión fiel del contenido del recinto físico. Las visitas virtuales interactivas añaden la posibilidad de moverse entre los distintos espacios que lo componen, así como el detallar los elementos importantes que en ella se consideren. Este tipo de visitas virtuales interactúan con el usuario dándole cierta libertad de movimiento.

³ <http://www.ucv.ve>

1.3. Objetivo general del TEG

Desarrollo de un Portal Informativo y Divulgativo del Patrimonio Artístico de la Ciudad Universitaria de Caracas (CUC) que permita realizar paseos virtuales por la Plaza Cubierta del Rectorado y permita conocer la riqueza artística de estos espacios.

1.4. Objetivos específicos

Con la finalidad de alcanzar el objetivo general, se plantearon los siguientes objetivos específicos:

1. Utilizar la metodología seleccionada en el Trabajo de Seminario para la creación de la solución conforme a los requerimientos identificados.
2. Crear la lista y la planificación de funcionalidades de acuerdo a los requisitos.
3. Diseñar el modelo de datos.
4. Diseñar la interfaz de usuario y del administrador.
5. Desarrollar el portal utilizando las tecnologías convenientes para incorporar una galería de imágenes, un paseo virtual con vistas panorámicas, una base de datos para las fichas cartográficas de las obras de arte y las biografías de los artistas y un módulo de administración que permita gestionar el contenido del sitio.
6. Realizar pruebas de aceptación para verificar el funcionamiento del portal y las mejoras necesarias para responder a las expectativas del usuario potencial.

1.5. Alcance de la aplicación

El portal debe permitir a los usuarios realizar las siguientes acciones:

1. Mostrar un plano general de las obras de arte, con la ayuda de mapas y georeferenciación, en donde destaquen las pinturas, mosaicos, esculturas, murales y edificaciones ubicados en la Ciudad Universitaria de Caracas, limitándose en este contexto al caso de estudio de la zona Plaza Cubierta del Rectorado.
2. Proporcionar fichas informativas de las obra de arte y de los autores, con información relevante en cada caso.

3. Permitir el listado de imágenes de las obras de arte con ordenamiento por: nombre, autor, zona, fecha y tipo de obra (escultura, mural, pintura, entre otros).
4. Permitir la agrupación de los elementos (obras de arte, edificaciones, paisajismo, jardín botánico y otras colecciones) que conforman el patrimonio de la Ciudad Universitaria de Caracas dentro de su conjunto arquitectónico.
5. Permitir paseos virtuales mediante vistas panorámicas de los lugares de interés, posibilitando la exploración de las obras de arte deseadas.
6. Permitir la gestión de los contenidos del portal mediante un módulo administrativo, en donde se pueda realizar el proceso de inserción, modificación y eliminación del contenido, generación de recorridos, etiquetado y manipulación de la metadata relacionada con el portal.
7. Incluir las siguientes secciones:
 - i. Inicio
 - ii. ¿Por qué somos patrimonio?
 - iii. Historia de la Ciudad Universitaria de Caracas.
 - iv. Conjunto Arquitectónico.
 - v. Paseo Virtual.
 - vi. Galería.
 - vii. Recorridos.
 - viii. Documentos.
 - ix. Contáctenos.
 - x. Créditos.

1.6. Potenciales usuarios

Principalmente usuarios con intereses culturales, especialmente en el área artística y en la arquitectura, estos pueden ser docente, estudiantes, profesores y demás profesionales. También servirá a todas aquellas personas interesadas en el patrimonio cultural y en las obras de la Universidad Central de Venezuela y de su Ciudad Universitaria.

CAPÍTULO II

MARCO TEÓRICO

La finalidad de este capítulo es presentar la definición de paseo virtual, la evaluación de diversos paseos virtuales y museos en donde se puedan extraer características que se puedan agregar al desarrollo de la TEG, se revisan las diversas herramientas para crear pasos virtuales y las galerías y los diversos formatos de almacenamiento más adecuados para el portal, tipos de fotografía y posibles alternativas de *software* para el tratamiento de las mismas, herramientas de diseño gráfico digital, patrón de arquitectura, tecnologías para el desarrollo del portal web y la metodología utilizada.

2.1 Visita virtual

Se conoce como visita virtual a la simulación de un espacio real; esta se compone de dos o más imágenes dimensionales unidas mediante un *software* especial, creando una imagen interactiva que el usuario puede controlar mediante el computador. Lo que permite decir que una visita virtual es lo más parecido a estar realmente en el sitio (Ecured.cu, s.f.).

Entre las ventajas que poseen las visitas virtuales, se pueden mencionar:

- ✓ Realismo e inmersión en la visita, dado que son imágenes reales.
- ✓ Sólo se necesita una computadora con conexión a Internet y cualquier sistema operativo, así como también un navegador instalado. No se requieren dispositivos especiales adicionales para poder disfrutar de la visita.
- ✓ Hay una gran cantidad de información que puede ser introducida en la visita, lo que permitirá al usuario conocer todos los detalles.
- ✓ Tienen acceso a ella todas las personas que, por motivos de diversa índole, no pueden trasladarse al lugar físicamente.
- ✓ No tiene horario de visita y el acceso es ilimitado, además no hay que pagar entradas.

2.2 Evaluación de algunos paseos o museos virtuales

Museo Virtual de América Latina y el Caribe (*Museo Virtual*, 2010)

Es una plataforma de registro y divulgación del patrimonio en línea, administrada en igualdad de condiciones por múltiples usuarios latinoamericanos y caribeños, creada para fomentar la integración y cooperación en el área cultural, con el objetivo principal de difundir la producción creativa de las diversas culturas que han hecho vida en la comunidad regional.

Basílica Papal San Pablo Extramuros (Basílica de San Pablo, s.f.)

Erigida en el siglo IV D.C., la Basílica de San Pablo Extramuros (Basilica di San Paolo Fuori le Mura) es una de las cuatro mayores basílicas de Roma, la segunda en tamaño después de la Basílica de San Pedro. Es el lugar de sepultura del apóstol San Pablo.

Instituto Nacional de Antropología e Historia (Inah, s.f.)

Es un organismo del Gobierno Federal Mexicano fundado en 1939, para garantizar la investigación, conservación, protección y difusión del patrimonio prehistórico, arqueológico, antropológico, histórico y paleontológico de México. Su creación ha sido fundamental para preservar el patrimonio cultural de ese país.

Museo Thyssen - Bornemisza (Museo Thyssen, 2009)

Creada en España en 1988, la Fundación Colección Thyssen-Bornemisza es pionera de una nueva fórmula de gestión privada de fondos públicos, cuya eficacia se puso de manifiesto desde la apertura del Museo en octubre de 1992. Desde entonces, tanto el Museo como la Fundación que lo gestiona, han crecido favorablemente, su colección se expandió como resultado del préstamo a largo plazo de la Colección Carmen Thyssen-Bornemisza, así como también experimentó la extensión de sus instalaciones con la ampliación del Palacio de Villahermosa en 2004, incrementando el número y variedad de exposiciones temporales, programas didácticos y actividades.

Se realizó un cuadro que permite ver las similitudes y diferencias entre las páginas previamente citadas. Esto nos permite obtener características apropiadas que pueden estar presentes en el desarrollo del paseo virtual del portal.

	Museo Virtual de América Latina y el Caribe	Basilica Papale San Paolo Fuori le Mura	Instituto Nacional de Antropología e Historia	Museo Thyssen - Bornemisza
Idioma del contenido	Español.	Inglés, francés, alemán, italiano y portugués.	Español.	Español e inglés.
Buscador	Sólo tiene opción de búsqueda por título, creador y clave.	No posee.	Sólo busca en la página.	Tiene dos modalidades: búsqueda sencilla y búsqueda avanzada. La búsqueda avanzada discrimina por artista, exposición, videos, obras y actividades.
Paginación	Sí posee. Es requerida debido a la gran cantidad de imágenes.	No posee.	No posee.	No posee.
Paseo Virtual	No posee. Sin embargo, la página está elaborada en Flash.	La página principal despliega un mapa con los puntos importantes a visitar, sin embargo, al ingresar al sitio sólo contiene una panorámica 360 x 360, con posibilidad de ampliar y reducir, sin opción de navegación por diferentes panorámicas. Elaborado con VR o Flash y HTML.	La página principal redirecciona la visita a una ventana nueva. La visita posee controles de dirección, ampliación y reducción, la imagen 360 x 360 ofrece rapidez y sensación de realismo acompañada de buena calidad. El usuario tiene dominio de la página. Carece de ubicación dentro de la panorámica. Elaborado en Flash y HTML.	La visita posee controles de dirección, acompañados de ampliación y reducción de las imágenes, con tiempos lentos de carga, pero con resultados de buena calidad. Ofrece un mapa del lugar al usuario y especifica su ubicación. Elaborado en Flash y HTML.

Tabla 1. Cuadro Comparativo de Aplicaciones Web con Paseo Virtual

Fuente: Los Autores

2.3 Evaluación de museos virtuales con colecciones y galerías de imágenes

En esta sección nos enfocaremos en páginas de museos que contengan colecciones y galerías de imágenes para poder extraer de ellas las mejores presentaciones que nos sugieran buenas ideas para ser utilizadas en el TEG. En el Trabajo de Seminario solo nos enfocamos en las páginas que poseen paseos virtuales sin tomar en consideración las colecciones y galerías de imágenes, las cuales también están presentes en el portal. A continuación destacamos varios museos que poseen presencia en espacios virtuales y están considerados entre los más visitados en el mundo:

“Metropolitan Museum of Art”, en Nueva York: Con más de dos millones de obras de arte procedentes de todas las partes del planeta, el “Metropolitan Museum of Art” es, sin duda alguna, uno de los museos más importantes del mundo. Cuenta con pinturas de autores clásicos como Velázquez o Rembrandt, con piezas procedentes de la antigua Grecia y del Imperio Romano, con una nutrida colección de obras medievales e islámicas, con un templo egipcio, con piezas de Japón y del lejano oriente, entre otros; el número de obras de arte de este museo es realmente increíble, y la procedencia de las mismas no podría ser más variada.

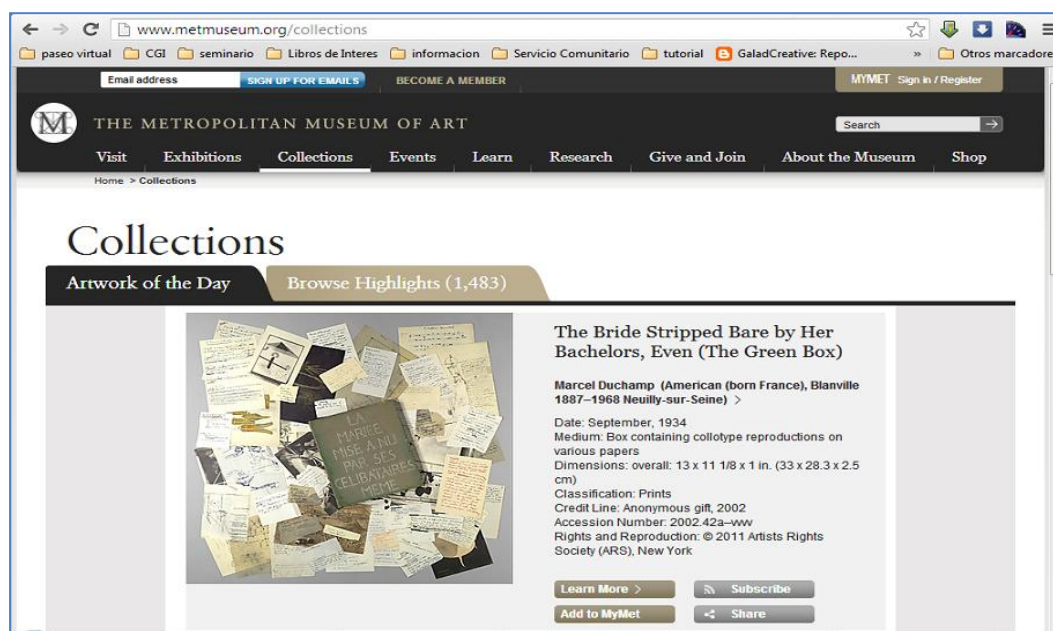


Figura 1. Vista de la Colección *The Metropolitan Museum of Art*

Fuente: (Art T. M., 2013)

Museo del Prado, en Madrid: Cuenta con una de las mayores colecciones de pintura de todo el planeta. Entre sus muchas obras destacan “Las Meninas” de Diego Velázquez, “El caballero con la mano en el pecho” del Greco o “Los fusilamientos del dos de mayo”, de Goya. Pero el Museo del Prado no sólo expone pinturas, también cuenta con una diversa colección de esculturas, dibujos, estampas y otros objetos artísticos de gran valor.



Figura 2. Vista de la Colección del Museo Nacional de Prado

Fuente: (Prado, 2013)

“National Gallery of Art”, en Washington, D.C.: Una de las galerías de arte más famosas del mundo, en su interior se pueden encontrar obras pertenecientes a distintos periodos históricos, entre las que destacan pinturas de los renacentistas Rafael y Leonardo da Vinci y del famoso artista contemporáneo Andy Warhol. El museo también es famoso por su “Jardín de las esculturas”, que expone obras de Miró y de Roy Lichtenstein, entre otros.

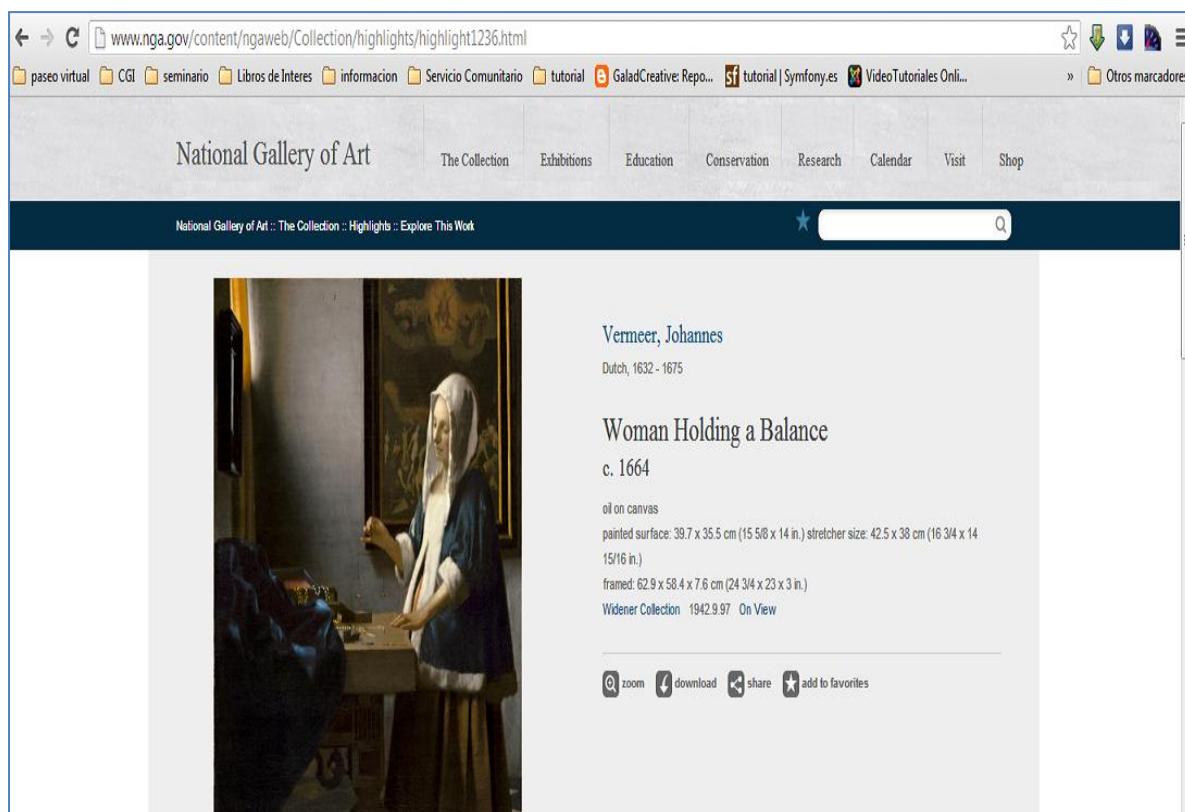


Figura 3. Vista de la Colección de *National Gallery of Art*

Fuente: (Art N. G., 2013)

Museo Reina Sofía, en Madrid: Es famoso por albergar el “Guernica” de Picasso, una de las obras más conocidas de este famoso pintor, y un nutrido grupo de obras de Dalí. Pero el museo no se dedica exclusivamente a la pintura, e incluye obras en todos los formatos. Cuenta con un gran número de tarifas, una de ellas gratuita, y sus colecciones están en un continuo proceso de renovación, por lo que es recomendable consultar el programa de exposiciones antes de visitarlo.

Museo Británico, en Londres: La “Piedra de Roseta”, una colección inigualable de momias egipcias y parte del friso del Partenón, tres de las muchas obras de incalculable valor que alberga el Museo Británico londinense, considerado por muchos historiadores y aficionados al arte como la mejor colección del mundo de su tipo.

“National Gallery”, en Londres: Cuenta con piezas que van desde finales del S. XIII hasta el S. XIX, siendo algunas de las más famosas la “Venus del espejo”, del español Diego Velázquez o “Los Girasoles”, de Van Gogh.

Museo de El Cairo: Diseñado en 1902, el Museo del Cairo se creó con la intención de agrupar todas las obras de arte egipcio que se descubrieron a lo largo de la segunda mitad del S. XIX, para así protegerlas y evitar que fueran robadas. Ubicado en el centro de El Cairo, cuenta con una de las mayores colecciones del mundo de arte egipcio, perteneciente a los distintos periodos de este imperio. Entre otras obras destaca, por ejemplo, el tesoro de Tutankamon, que incluye su máscara funeraria y su ataúd.

Museo Guggenheim, en Bilbao: Este museo resulta doblemente impresionante. Por un lado por su colección de obras de arte contemporáneo, y por otro por lo inusual y extravagante de su diseño, que corrió a cargo del famoso arquitecto Frank Gehry. Ubicado en el corazón de Bilbao, el museo Guggenheim se abrió al público en 1997 y desde entonces ha basado su oferta artística en obras de todo tipo de obras creadas desde la segunda mitad del S. XX hasta nuestros días.

Museo de Historia Natural, en Nueva York: Ubicado al lado del famoso “Central Park” neoyorquino, el “American Museum of Natural History” es un museo dedicado por entero a la ciencia. Cuenta con una de las mayores colecciones de fósiles de dinosaurios del mundo, que incluye al famoso Tyranosaurio, al Triceratops y al Diplodocus, entre otros. Pero, aparte de estos gigantes del Jurásico y Cretácico, el Museo Natural de Historia Americana cuenta con galerías enteras dedicadas a temas como la fauna y la flora del mundo (la réplica de la ballena azul a tamaño real es impresionante), el espacio o la antropología.

Museo del Louvre, en París: No hay más que mirar cuál es la obra más famosa de este museo para darse cuenta de su importancia: “La Gioconda” de Leonardo Da Vinci. El Louvre cuenta con una de las galerías de arte más importantes de Europa y del mundo, así como con un nutrido grupo de esculturas entre las que se encuentra la famosa “Venus de Milo”, otro de los

estandartes del museo. También incluye una extensa colección de obras provenientes del antiguo Egipto, por no hablar de la pirámide de cristal situada en la entrada del museo, que se ha convertido en otro de sus símbolos.

Google Art Project: Es un sitio web que presenta una recopilación de imágenes en alta resolución de obras de arte expuestas en varios museos del mundo, así como un recorrido virtual por las galerías en las que se encuentran. El proyecto fue puesto en servicio por Google el 1 de febrero de 2011 con 1.061 obras de diecisiete museos, entre los que se encuentran el *Tate Britain* de Londres, el Museo Metropolitano de Arte de Nueva York, y la Galería Uffizi de Florencia. La función de exploración de los museos se sirve de la misma tecnología utilizada por *Google Street View*. Aunque todas las obras se muestran en alta resolución, cada una de las diecisiete instituciones intervinientes al iniciarse el proyecto dispuso una de sus obras maestras para ser fotografiada y mostrada en el sitio mediante imágenes de 7.000 mega píxeles.

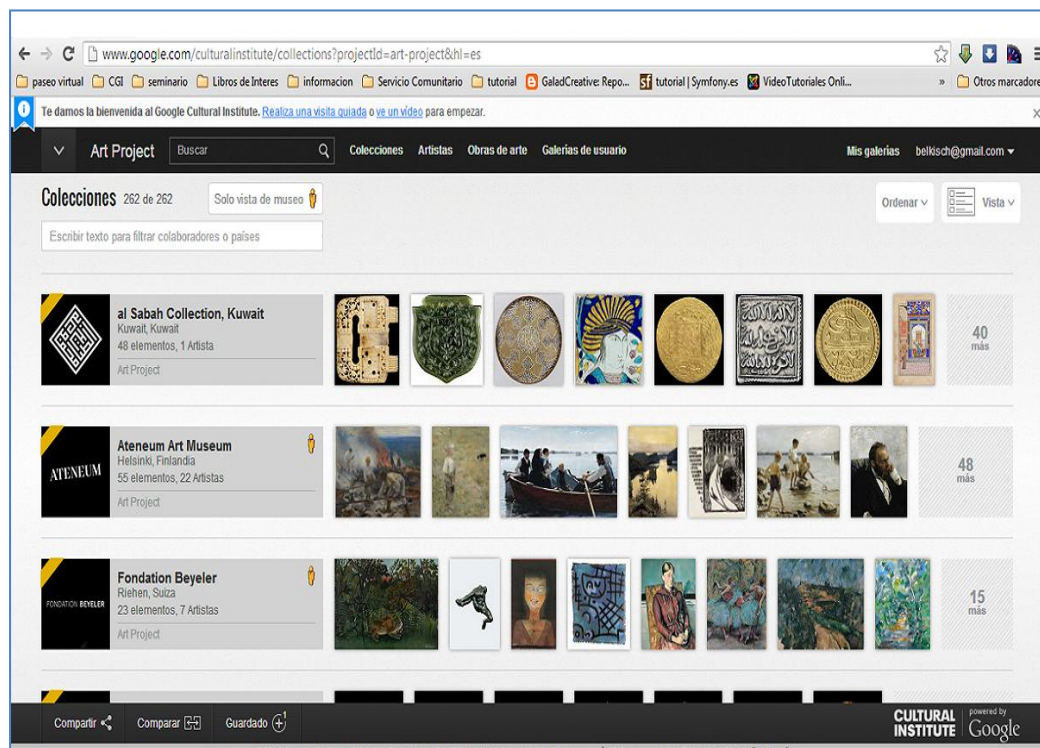


Figura 4. Vista de las Colecciones de Google Art Project

Fuente: (Project, 2011)

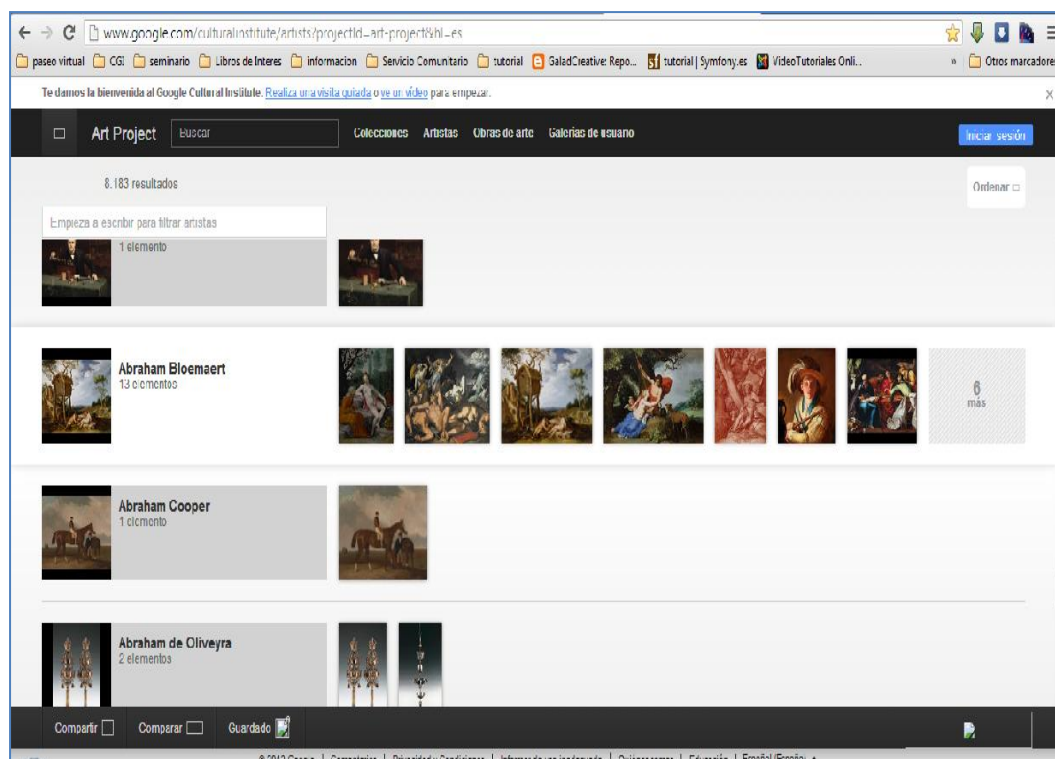


Figura 5. Vista de las Obras por Artista

Fuente: (Project, 2011)

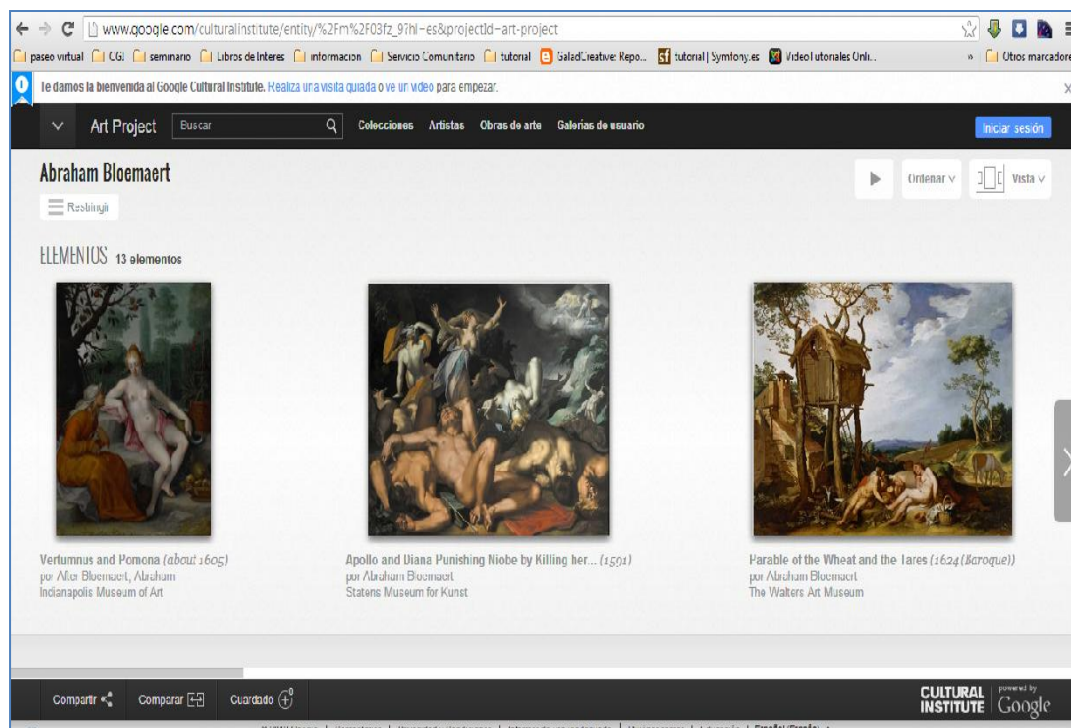


Figura 6. Vista de todas las Obras por un Artista con Breve Reseña

Fuente: (Project, 2011)

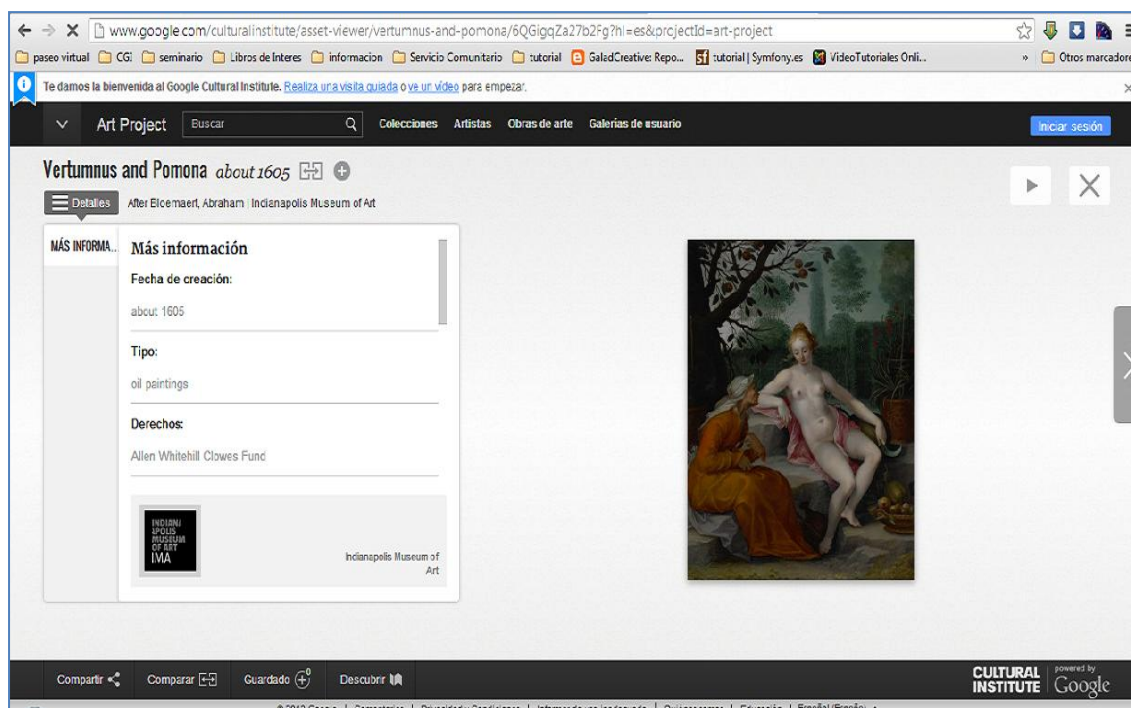


Figura 7. Vista de la Obra por un Artista con más Información

Fuente: (Project, 2011)

Análisis de las páginas visitadas

Las páginas visitadas tienen características o funcionalidades muy parecidas. En todas las colecciones o galerías de imágenes se presentan de forma similar, es decir, la imagen y la información respectiva según la clasificación si es pintura, escultura, entre otros. Sin embargo, en muchos de los casos la presentación es muy formal y no hay dinamismo en las páginas. La que salta estos estándares es sin duda la de *Google Art Project*, donde la presentación de las imágenes es muy dinámica y moderna. Aunque más pesada, se mantiene la presentación de la información y su clasificación. De esta página se pueden tomar interesantes características para incorporar en el TEG, como puede ser la vista de las imágenes por autores. De igual manera se presenta las demás clasificaciones.

2.4 Herramientas para crear paseos virtuales y galería de imágenes

Street View, aunque es una característica de Google Maps y Google Earth que proporciona panorámicas a nivel de calle (360 grados de movimiento horizontal y 290 grados de movimiento

vertical)), permitiendo a los usuarios ver partes de las ciudades seleccionadas y sus áreas, es una aplicación que tiene muy poca documentación para el uso del API y no está disponible en todas las regiones, su implementación es reducida y la elaboración está supeditada a los desarrollos de Google.

Las siguientes están estrechamente ligadas y se destacan como librerías de uso libre, que se pueden modificar o adaptar al desarrollo de cualquier aplicación y están integradas en el proyecto.

- ✓ **API de Google Maps:** permite incrustar mapas en las páginas web mediante JavaScript, además de poder crear potentes aplicaciones de los mapas en sitio web, el API proporciona utilidades para manipular los mapas como añadir contenido mediante diversos servicios. Es una alternativa en la integración con mapas, en donde se pueden incluir las diferentes capas necesarias para la organización y ubicación de las obras de arte a nivel cartográfico.
- ✓ **JQuery Virtual Tour:** es una extensión para simple panorama *viewer*, programa que permite hacer presentaciones con JQuery añadiendo interactividad al transformar algunas vistas panorámicas en un paseo virtual. Al agregar esta librería nos puede permitir estructurar visitas virtuales por el recinto, mediante diferentes imágenes panorámicas conectadas según los enlaces.
- ✓ **JQuery UI:** es una biblioteca de componentes para el *framework* JQuery que le añaden un conjunto de librerías, para crear efectos visuales en las aplicaciones web. Se utiliza la función de autocompletar este permite encontrar de forma rápida y seleccionar de una lista previa que contiene los valores a medida que escriba.
- ✓ **Fancybox:** es una herramienta para mostrar las imágenes de forma moderna, contenido HTML y multimedia en un estilo similar al “*lightbox*” de MAC que flota encima de la propia página web incluyendo su carga y ampliación, así como también el comportamiento general del sitio. Esta librería es bastante sencillo de utilizar aunque también permite un alto grado de personalización utilizando técnicas algo más complejas.

- ✓ **JQuery NailThumb:** permite crear miniaturas de cualquier imagen con alta resolución, sin ningún tipo de distorsión, con una línea de código JQuery. Incluso puede decidir dimensiones miniatura directamente de CSS. Se integra con cualquier tipo de galería, muy útil en aplicaciones web dinámicas cuando se puede hacer frente a cualquier forma o tamaño.
- ✓ **Spritely:** es una librería de JQuery que permite crear fondos (*background*) en los sitios web animados, es muy ajustable a las necesidades de los usuarios, puedes agregar movimientos a imágenes haciendo del sitio algo más interactivo. Fue creado por *Artlogic* para la creación de animaciones de fondo en HTML puro y *JavaScript*. Es simple, ligero con algunos métodos sencillos para crear *sprites* animados.
- ✓ **Galleria:** Crear y publicar hermosas galerías de fotos. Utiliza un modelo de datos que contiene todos los datos de la imagen que se utilizará para la galería. Una característica muy solicitada en *Galleria* es la posibilidad de añadir HTML plano por encima de la imagen que también sigue una transición elegida.

2.5 Formato de almacenamiento para imágenes

De los formatos de almacenamiento revisados en el trabajo de seminario se recompilan en la siguiente tabla las características más significativas de los tres formatos de imagen recomendados para publicaciones en la web:

	JPG - JPEG	GIF	PNG
Cantidad de colores	24 bits color o 8 bits B/N	Hasta 256 colores	24 bits color
Grado de compresión	Muy alto grado de compresión	Formato de compresión	Mayor compresión que el formato GIF (+10%)
Carga progresiva	Admite	Admite	Admite
Transparencia en fondos	No admite	Admite	Admite fondos transparentes en 8-bits
Animación	No permite	Permite	No permite

Tabla 2. Características Significativas de los Formatos de Imagen

Fuente: (Formación en Red, 2008)

Análisis de los formatos de almacenamiento

Con respecto a los formatos que se utilizan en el desarrollo de la aplicación es JPG/JPEG para las fotografías, considerando que es el formato por defecto arrojado por la mayoría de las cámaras modernas, y GIF para los logos o animaciones simple, PNG tiene un amplio nivel de aceptación, bajo los niveles de compresión y ajuste adecuados, dado que permite obtener una buena relación calidad/peso, así como también el almacenamiento de imágenes con fondo transparente, que suelen ser de mucha utilidad en el diseño web.

2.6 Fotos panorámicas

Las fotos panorámicas son representaciones diversas. Constituyen un mundo virtual tridimensional, abiertamente dispuesto para ingresar y recorrer, facilitando la observación de los distintos puntos de vista escenográficos, mediante su distribución en un horizonte de 360°.

En fotos panorámicas suelen admitirse acercamientos en zonas específicas, lo que permite explorar detalles. Otra característica resaltante es su conexión mediante enlaces, para crear recorridos virtuales que elevan la interactividad con el entorno, pudiendo estar ambientado con fondos musicales, destellos de luz, mapas y más (Fotografia360, s.f).

2.6.1 Calidad de imagen, dimensiones verticales y horizontales

Las imágenes se desarrollan siempre en varias calidades. La calidad final que se desea obtener dependerá del medio de distribución (Internet, red local, CD). Para el caso de Internet, se escogen aquellas imágenes que optimizan el compromiso entre calidad y tiempo de descarga, mientras que para imágenes destinadas a distribuirse en un CD, se utilizan altas resoluciones, y por ende una alta calidad.

En las imágenes panorámicas de rotación tienen un tamaño máximo de 640 x 480 pixeles, y según el contenido de la fotografía y de lo que se quiere mostrar, se eligen las dimensiones que la hagan más atractiva, por lo que no es muy recomendable que las imágenes desarrolladas para ser vistas en Internet sean presentadas en tamaño muy grande, para así evitar que sea visible la

pérdida de calidad de la imagen. A continuación en la siguiente tabla se muestran los tipos de fotografías que se pueden utilizar en el portal web.

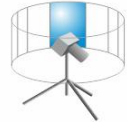
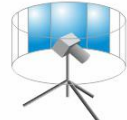
DEFINICIÓN		IMAGEN
Fotos clásicas o de papel	Es la que todos conocemos, requiere de la habilidad del fotógrafo para mostrar lo que se desea, ya que solo permiten ver una pequeña parte de la escena.	
Fotos panorámicas (panorámica plana o parcial)	Son el montaje o empalme de varias fotos clásicas, que muestran áreas más amplias, pero que no tienen los 360° completos.	
Fotos cilíndricas (panorámica 360 a 90 grados)	Es el montaje de varias fotos, las cuales se unen en sus extremos y generan una vista en 360°. Este tipo de fotos no permiten ver en su totalidad áreas del piso y techo, pero permiten rotar la cámara en su propio eje, en un solo ángulo.	
Fotos esféricas (panorámica 360 a 180 grados)	Permiten observar en cualquier dirección, ofreciendo así una vista completa de pisos y techos, son tomadas con lentes especiales que logran rotar la cámara sobre su propio eje en diversos ángulos.	

Figura 8. Foto Clásica o de Papel

Figura 9. Foto Panorámica

Figura 10. Foto Cilíndrica

Figura 11. Foto Esférica

Tabla 3. Tipos de Fotografías

Fuente: (Fotografia360, s.f.)

2.6.2 Técnica de montaje para crear fotos panorámicas

La técnica más utilizada en panorámicas consiste en tomar múltiples instantáneas, una a continuación de otra en el espacio, y posteriormente montarlas en el PC, de modo que las zonas comunes coincidan y dé la sensación de formar parte de una captura única. Una vez montadas las imágenes, estos programas nos dan la oportunidad de recortar la panorámica resultante para igualar los bordes; algunas de las aplicaciones añaden opciones y herramientas de mejora de la foto, tales como ajustes del contraste, variación del color, etc. (Rodríguez, 2007).

A continuación se presenta un cuadro de comparación donde se muestran algunos programas que se pueden utilizar en el tratamiento de fotos panorámicas.

	HUGIN	AUTOSTITCH	PTGUI	AUTOPANO PRO
Tipo de Software	Libre	Propietario	Propietario	Propietario
Sencillez	Diversidad de opciones	Muy sencillo	Sencillo	Intuitivo
Precisión	Puntos de control manuales. Parámetros modificables.	Automático	Puntos de control manuales. Parámetros modificables.	Automático. Parámetros modificables.
Proyecciones	Rectilínea (plana) Cilíndrica Equirrectangular Ojo de Pez Estereográfica Mercator Sinusoidal	Cilíndrica	Rectilínea (plana) Cilíndrica Equirrectangular (esférica) Circular Fullframe Mercator	Plana Cilíndrica Esférica
Formato de Salida	JPG TIFF	JPG	JPG TIFF PSD MOV	JPG TIFF PNG PSD
HDR	Si	No	Si	Si
Sistema Operativo	Linux Windows Mac	Windows	Windows Mac	Linux Windows Mac

Tabla 4. Comparación entre Programas para el Tratamiento de Fotos

Fuente: (Hugin, 2008)

2.7 Herramientas para el diseño gráfico digital

El uso y conjugación de elementos gráficos en la web se conoce como diseño web. El diseño web es el arte de hacer un contenido atractivo para los visitantes de un portal o pagina web, por lo que los buenos diseñadores obtienen el beneficio de un mayor tráfico. En ese sentido, los diseñadores colaboran con las empresas de diseño web profesionales con el fin de ofrecer

resultados de calidad (TechNoticias, 2011). Con respecto a las herramientas más útiles para el diseño gráfico, estas son algunas de ellas:

Paint: Es una herramienta profesional de dibujo, pintura y edición de fotografía. Es posible descargar el software y utilizarlo como una aplicación de escritorio. Cuenta con un interfaz amigable, sin embargo, sólo puede ser instalada en versiones avanzadas de Windows.

Blender: Programa gratuito de contenido 3D multimedia y suite de creación. Está disponible para Windows, Mac y Linux. Funciona muy bien como un software de desarrollo 3D, en él se pueden hacer animaciones frescas y atractivas para insertar en un sitio web.

Phoenix: Permite el retoque básico de imágenes con posibilidad de insertar efectos complejos. Se trata de una web completa que funciona como herramienta de edición de imágenes. Cuenta con diversos paneles de edición de imágenes y útiles opciones para la adición de capas y efectos como la transparencia.

GIMP (*GNU Image Manipulation Program*): Es un programa libre y gratuito de edición de imágenes digitales en forma de mapa de bits, tanto dibujos como fotografías. Está englobado en el proyecto GNU, se encuentra disponible bajo la Licencia pública general de GNU. GIMP lee y escribe la mayoría de los formatos de ficheros gráficos, entre ellos jpg, gif, png, pcx, tiff, y los de Photoshop, además de poseer su propio formato de almacenamiento de ficheros: xcf (Gimp.es, s.f.).

RIOT (*Radical Imagen Optimization Tool*): Es una herramienta de optimización de imagen radical gratuita que le permitirá ajustar visualmente los parámetros de compresión, manteniendo el tamaño mínimo. Es ligero, rápido y fácil de usar, pero potente. Es capaz de controlar la compresión, número de colores, configuración de metadatos y seleccionar el formato de imagen (JPG, GIF o PNG) para el archivo de salida.

Photoshop: Es una aplicación para la creación, edición y retoque de imágenes, desarrollada por la compañía *Adobe Systems*. Se lanzó originalmente para computadoras Apple, pero luego saltó

a la plataforma Windows. Los formatos propios de Photoshop son PSD y PDD, que guardan capas, canales, guías y cualquier modo de color, también soporta otros formatos como PostScript, EPS, DCS, BMP, GIF, JPEG, PICT, PIFF, PNG, PDF, IFF, PCX, RAW, TGA, Scitex CT, Filmstrip, FlashPix (Alegsa, 1998).

Análisis de las herramientas para el diseño gráfico digital

De las herramientas mencionadas anteriormente para el diseño gráfico y tratamiento de imagen y logo, se recomienda el uso de GIMP además que se puede integrar con RIOT para mejor compresión y optimización de la imagen. La motivación de esta selección radica principalmente en las características del programa y aceptando la amplia gama de herramientas que ofrece GIMP, además de la libertad de uso que fomentan las licencias GNU.

2.8 Patrón de arquitectura

El patrón de arquitectura propuesto para desarrollar el portal será el Modelo Vista Controlador, el cual se definirá a continuación:

2.8.1 Modelo - Vista - Controlador (MVC)

El modelo vista controlador es un patrón de arquitectura de software, encargado de separar la lógica de negocio de la interfaz del usuario, es el más utilizado en aplicaciones web, ya que facilita la funcionalidad, mantenimiento y escalabilidad del sistema, de forma simple y sencilla, a la vez que permite no mezclar lenguajes de programación en el mismo código (Ecured.cu, s.f).

MVC divide las aplicaciones en tres niveles de abstracción:

- ✓ **Modelo:** Es la representación de la información que maneja la aplicación. El modelo en si son los datos puros, que puestos en contexto del sistema proveen la información al usuario o a la aplicación misma.
- ✓ **Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso del portal, la vista es una página HTML con contenido dinámico sobre la cual el usuario puede realizar operaciones.

- ✓ **Controlador:** Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información requerida y modificando el modelo en caso de ser necesario.

2.8.2 Ventajas y desventajas de MVC

Las principales ventajas de hacer uso del patrón MVC son:

- ✓ Permite la separación de los datos de la representación visual de los mismos.
- ✓ Es mucho más sencillo agregar múltiples representaciones de los mismos datos o información.
- ✓ Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- ✓ Ofrece maneras más sencillas para probar el correcto funcionamiento del sistema.

Las desventajas de seguir el planteamiento de MVC son:

- ✓ La separación de conceptos en capas agrega complejidad al sistema.
- ✓ La cantidad de archivos a mantener y desarrollar se incrementa considerablemente.
- ✓ La curva de aprendizaje del patrón de diseño es más alta que usando otros modelos más sencillos.

El uso de MVC como patrón de arquitectura se fundamenta totalmente en sus ventajas. MVC permite construir aplicaciones y sitios web bajo desarrollos limpios, cuyas partes se encuentran certeramente distinguidas.

2.9 Tecnologías para el desarrollo del portal

Dadas las exigencias presentadas y esbozadas en la investigación del seminario, se optó por seleccionar del grupo de herramientas las siguientes tecnologías PHP, MySQL y *JavaScript* por las siguientes razones:

PHP es un lenguaje de programación de amplia utilización para el desarrollo de páginas web dinámicas. A diferencia de lo que sucede en una página estática, donde el contenido que se muestra es siempre predeterminado, en las páginas dinámicas el contenido se genera en el momento de la visualización, a partir de datos variables. Las posibilidades interactivas de PHP encuentran aplicación en foros, blogs, portales de comercio electrónico y cualquier sitio que

requiera interactuar con bases de datos, ya que ofrece una perfecta integración con MySQL y Oracle, entre otros sistemas.

PHP funciona sobre prácticamente todas las plataformas imaginables y garantiza una alta velocidad de ejecución, además de una excelente estabilidad. Su seguridad se ve reforzada por el hecho de que el código original permanece oculto al usuario: el navegador lo ejecuta y lo muestra “traducido” a HTML. A diferencia de otros lenguajes (como ASP, de Microsoft) es gratuito y *Open Source*. Y gracias a una enorme comunidad de desarrolladores a nivel mundial, cuenta con una amplia documentación.

Por otro lado el sistema de administración de bases de datos es la parte más importante de un sistema de base de datos ya que se encarga de ejecutar todas las operaciones sobre la base de datos además de llevar a cabo funciones como: control de concurrencia, seguridad, integridad, respaldo y recuperación de datos. Para el desarrollo del portal se utilizará MySQL, es un sistema de administración de bases de datos relacionales rápido sólido, flexible y permite manejar con facilidad grandes volúmenes de información, posee gran velocidad a la hora de realizar operaciones ofreciendo así mayor rendimiento.

Las bases de datos relacionales están constituidas por una o más tablas que contienen los datos de una forma organizada, esto agrega velocidad y flexibilidad. Las tablas son enlazadas al definir relaciones que hacen posible combinar datos de varias tablas cuando se necesitan consultar datos. La parte SQL de "MySQL" significa "Lenguaje Estructurado de Consulta", y es el lenguaje más usado y estándar para trabajar con bases de datos relacionales.

Adicionalmente, JavaScript junto con JQuery, que permite procesamiento adicional de datos y, con especial énfasis, manipulación de los elementos gráficos que generan dinamismo en las visitas virtuales y galerías. También se evalúa el uso de AJAX, su principal objetivo consistirá en facilitar las tareas del usuario en búsquedas y soluciones de autocompletado, así como también aquellas que requieran de simulación de carga desde el servidor.

Además agregamos para el desarrollo del TEG *CakePHP* es un marco de desarrollo *framework* rápido para PHP, libre y de código abierto, creado sobre los conceptos de *Ruby on Rails*. Se trata de una estructura que sirve de base a los programadores para que éstos puedan crear aplicaciones Web. Su principal objetivo es que se pueda trabajar de forma estructurada y rápida, sin pérdida de flexibilidad.

Facilita la creación de administradores para el manejo de contenido estándar y operaciones CRUD (insertar, editar, actualizar y eliminar registros), facilita la validación de los campos de los formulario, previene de ataques por *SQL Injection*⁴, facilita el testeo de las aplicaciones, y la interacción con la base de datos mediante el uso de *ActiveRecord*⁵, también tiene un sistema de plantillas rápido y flexible. Además del patrón arquitectónico Modelo Vista Controlador.

2.10 Metodología utilizada

En el trabajo de investigación del seminario se seleccionó una metodología ágil porque se basa en el desarrollo iterativo e incremental, es cooperativo dado que el cliente y el desarrollador trabajan junto en una constante comunicación y responde a cambios. Varias metodologías encajan bajo el estandarte de ágil, todas ellas comparten muchas características y también hay algunas diferencias significativas. Entre las metodologías ágiles que podemos mencionar están:

- ✓ *Scrum*.
- ✓ *Crystal*.
- ✓ Programación Extrema (*Extreme Programming*, XP).
- ✓ Gestión de Proyectos Evolutiva (*Evolutionary Project Management*, Evo).
- ✓ Desarrollo Basado en Funcionalidades (*Feature Driven Development*, FDD).
- ✓ Desarrollo Adaptable de Software (*Adaptive Software Development*, ASD).

⁴ Es una técnica donde un atacante crea o altera comandos SQL existentes para exponer datos ocultos, sobreponer a los que son importantes.

⁵ Es un enfoque de acceso a datos en una base de datos. Consiste en envolver la tabla de una base de datos relacional en una clase.

- ✓ Desarrollo de Software Magra (Lean Software Development, LSD), entre otros.

En el desarrollo del Trabajo Especial de Grado se utilizará la Metodología de Desarrollo basado en Funcionalidades (FDD) ya que, se trata de un proceso ligero el cual posee la documentación necesaria para el desarrollo del mismo y permite el trabajo de los desarrolladores de forma separada, teniendo sesiones de trabajos cada cierto tiempo para compartir impresiones y apoyarse.

Una descripción general de la Metodología FDD puede encontrarse a continuación:

2.10.1 Desarrollo Basado en Funcionalidades (*Feature Driven Development*, FDD)

FDD es una Metodología Ágil para el desarrollo de sistemas, basado en la calidad del software, que incluye un monitoreo constante del proyecto. Fue desarrollado por Jeff De Luca y Peter Coad a mediados de los años 90. Se estructura alrededor de la definición de características (*features*), que representan la funcionalidad que debe contener el sistema, tienen un alcance lo suficientemente corto como para ser implementadas en pocas semanas y pueden ser agrupadas en diferentes jerarquías para representar aspectos en común de la aplicación (Herman, 2004)

A diferencia de otros procesos ágiles, no cubre todo el ciclo de vida, sólo las fases de diseño y construcción. No requiere un modelo específico de proceso y se complementa con otras metodologías, enfatiza cuestiones de calidad y define claramente entregas tangibles y formas de evaluación del progreso (Uamisoft.izt.uam.mx, s.f.).

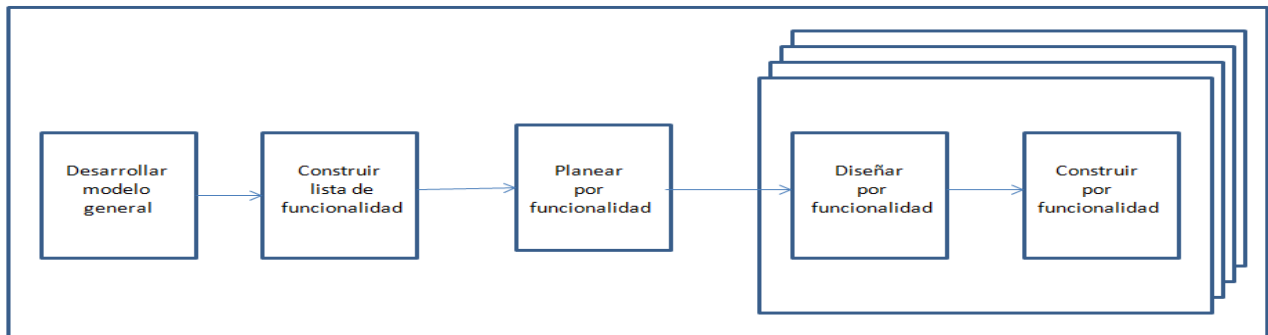


Figura 12. Proceso de Desarrollo Basado en Funcionalidades (FDD)

Fuente: (Valverde, 2007)

Un proyecto que sigue FDD se divide en cinco fases (Herman, 2004):

- ✓ Desarrollo de un modelo general (*Develop Overall Model*): El proyecto inicia con un alto nivel de *walkthrough*⁶ del ámbito y contexto del sistema
- ✓ Construcción de la lista de funcionalidades (*Built Feature List*): El conocimiento escogido en el modelo es usado para identificar una lista de características.
- ✓ Planificación en base a las funcionalidades a implementar (Plan By Feature).
- ✓ Diseñar en base a las funcionalidades (*Design By Feature*).
- ✓ Implementar en base a las funcionalidades (*Built By Feature*).

La jerarquía entre las funcionalidades está definida de la siguiente manera:

- ✓ *features*: son las funcionalidades específicas.
- ✓ *features sets*: son las que poseen características en común con el sistema.
- ✓ *major features sets*: es el más alto nivel de agrupación, abarca diferentes *feature sets*, posee un nivel de abstracción al cliente.

Esta jerarquía provee el siguiente formato:

- ✓ *feature*: <acción> el <resultado> <de|para|sobre|por> un <objeto>.
- ✓ *feature sets*: <acción> <-endo> un <objeto>.
- ✓ *major feature sets*: administración de <acción>.

⁶ Es una manera de revisar el software por pareja, se diferencia de una revisión técnica del software en la forma de inspección pues mencionamos que se interactúa directamente con el código.

CAPÍTULO III

MARCO APLICATIVO

El desarrollo de un proyecto está asociado a una serie de actividades y etapas en donde se analizan, diseñan y construyen estructuras, siguiendo las pautas de una metodología de desarrollo dada.

Con el fin de dar cumplimiento a los objetivos establecidos se seleccionó la metodología de Desarrollo Basada en Funcionalidades (*Feature Driven Development – FDD*), entre otras razones, por qué es una metodología adaptable en donde el diseño y el desarrollo se realizan en períodos cortos en base a funcionalidades. A continuación se indican las etapas de este desarrollo utilizando FDD:

3.1 Desarrollo de un modelo global

En base a la investigación realizada en el Trabajo de Seminario (Gutiérrez & Chacón, 2012), el levantamiento de los requisitos con los que debe contar el sistema se realizó aplicando la técnica de entrevista con el cliente. Además, se verificaron con éste las actividades que se deberían cumplir en el proceso, siendo las principales:

- ✓ Ofrecer un mapa que detalle la ubicación de las obras de arte en la Ciudad Universitaria de Caracas dentro de la Plaza Cubierta del Rectorado.
- ✓ Permitir a los usuarios realizar paseos virtuales por la zona de la Plaza Cubierta del Rectorado en la Ciudad Universitaria de Caracas, desplegando información adicional sobre las obras de arte.
- ✓ Presentar la información de los autores de las obras de arte.
- ✓ Centralizar la administración del portal a través de un módulo diseñado para tal fin, que integre todas las funcionalidades de gestión del contenido (inserción, modificación y eliminación).

- ✓ Restringir el acceso al panel administrativo a usuarios registrados.
- ✓ Permitir la administración de los perfiles de usuario.

Después de tener los requisitos capturados, fueron refinados y estructurados, para facilitar la elaboración de la lista de principales funcionalidades que será vista en la siguiente etapa, con apoyo en los diagramas de casos de uso (Anexo A). Como culminación de esta etapa se muestra el diagrama de clases que representa la estructura de la base de datos.

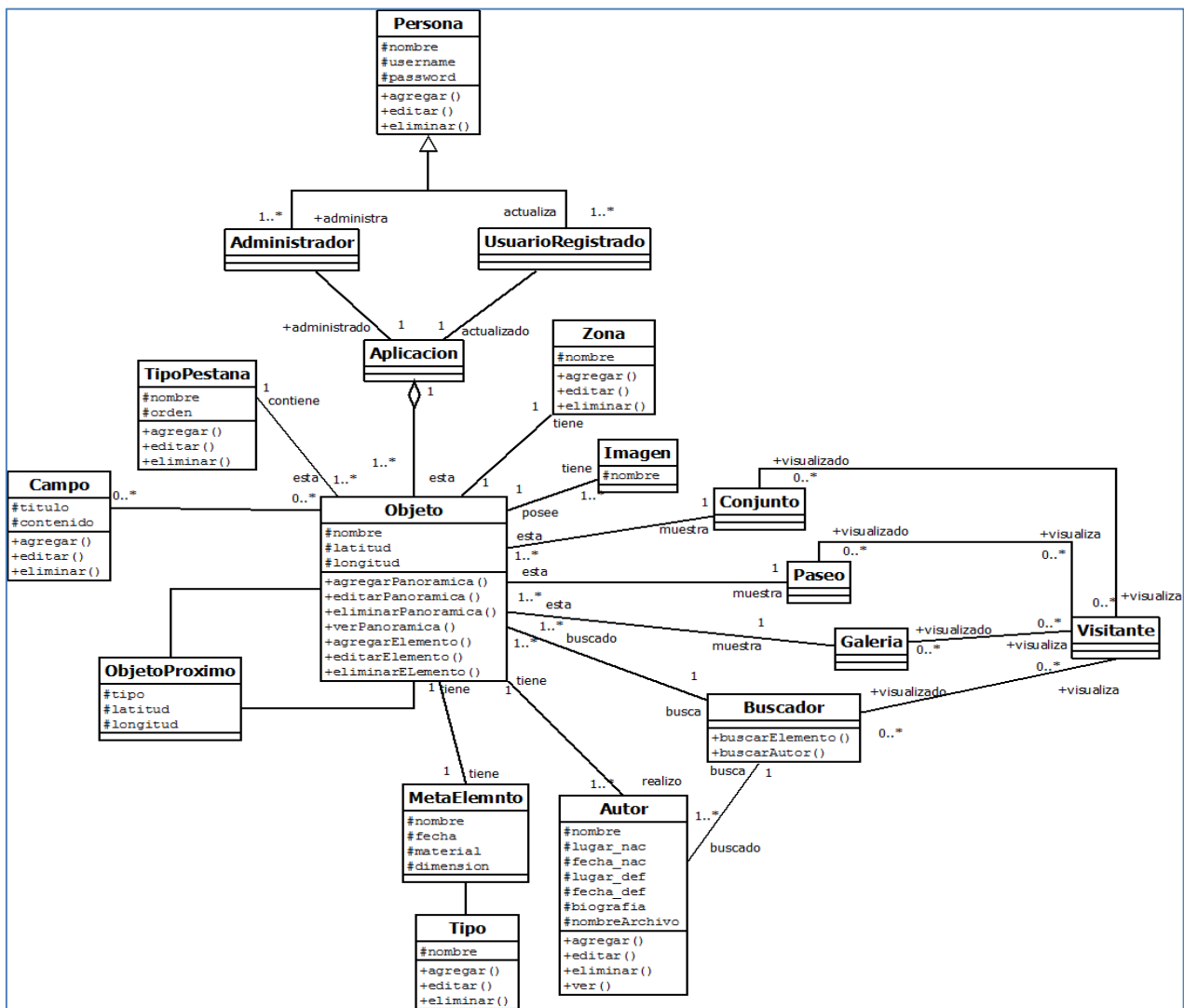


Figura 13. Diagrama de Clases del Portal

Fuente: Los Autores

3.1.1 Requerimientos no funcionales del sistema

Los requerimientos no funcionales son los que van a permitir la eficiencia y productividad de la aplicación, entre los más destacados dentro de la aplicación tenemos:

- ✓ **Fiabilidad:** El portal web deberá ser fiable, su desarrollo debe transcurrir bajo metodologías y procesos de ingeniería de software.
- ✓ **Disponibilidad:** El portal deberá estar disponible continuamente, se accederá a él por medio de Internet.
- ✓ **Seguridad:** El portal web deberá ser seguro, utilizando el control de acceso para gestionar los datos, permitiendo la asignación de privilegios según el perfil del usuario. Adicionalmente, las contraseñas deben ser guardadas en la base de datos encriptado.
- ✓ **Escalabilidad:** El portal web deberá ser escalable, estará diseñado bajo un esquema modular de acuerdo a los fundamentos de la ingeniería de software.

3.2 Lista de funcionalidades

Con base en el modelo global anterior, se elaboró una lista de funcionalidades, que agrupa tres conjuntos principales: las funcionalidades a nivel de usuario con perfil visitante y las funcionalidades a nivel de usuario con perfil administrador y registrado.

Funcionalidades a nivel de usuario con perfil visitante: Son todas aquellas funcionalidades asociadas al portal de visitantes. Se listan a continuación:

- 1) Agrupar la información sobre los elementos en fichas, con versión para imprimir, que contienen la siguiente información: Nombre, autor, año de creación, ubicación, tipo de obra, material, dimensiones, galería de imágenes integrada y otros campos adicionales que varían de acuerdo al tipo de obra.

- 2) Agrupar la información sobre los autores en fichas, con versión para imprimir, que contengan la siguiente información: Nombre, fecha y lugar de nacimiento, fecha y lugar de fallecimiento (Si aplica), una fotografía y una breve reseña biográfica.
- 3) Ejecutar paseos virtuales por la zona de la Plaza Cubierta del Rectorado en la Ciudad Universitaria de Caracas, utilizando vistas panorámicas en 360° con enlaces incluidos; dichos enlaces permiten:
 - a. Desplegar la ficha informativa de una obra de arte ubicada en el lugar. En este particular, el nombre del autor es un enlace que permite abrir su ficha.
 - b. Continuar la navegación hacia otra vista panorámica.
- 4) Construir un mapa con la ubicación de las obras de arte de la Plaza Cubierta del Rectorado en la Ciudad Universitaria de Caracas.
- 5) Agrupar las obras de la Plaza Cubierta del Rectorado en la Ciudad Universitaria de Caracas como parte del conjunto arquitectónico.
- 6) Presentar de una galería fotográfica con las imágenes de las obras, pudiendo agruparlas por: nombre, autor, zona, tipo de obra y fecha.

Funcionalidades a nivel de usuario con perfil administrador y registrado: Son todas aquellas funcionalidades asociadas al portal administrativo y que permitirán la gestión del contenido, se listan a continuación:

- 1) Gestionar los autores, que incluye:
 - a. Listar autores: Despliega la información en una tabla, para cada entrada muestra el nombre del autor, sus obras, su lugar y fecha de nacimiento y las acciones relacionadas (ver detalle, editar y eliminar).
 - b. Ver detalle: Permite mostrar la información completa de un autor.

- c. Editar autor: Permite la modificación de la información almacenada sobre cada autor.
 - d. Eliminar autor: Permite eliminar un autor de la base de datos.
 - e. Agregar autor: Permite la inserción de un autor en la base de datos.
- 2) Gestionar las zonas, que incluye:
- a. Listar zonas: Despliega la información en una tabla, cada entrada muestra el nombre de la zona, sus obras, sus panorámicas y acciones relacionadas (editar y eliminar).
 - b. Editar zona: Permite la modificación del nombre de cada zona.
 - c. Eliminar zona: Permite eliminar una zona de la base de datos.
 - d. Agregar zona: Permite la inserción de una zona en la base de datos.
- 3) Gestionar los tipos de elementos (son imágenes del conjunto las cuales puede ser una obras de arte, paisajismos, entre otros; así como la información respectiva o imágenes de la galería), que incluye:
- a. Listar tipos de elementos: Despliega la información en una tabla, cada entrada incluye el nombre del tipo y las acciones relacionadas (editar y eliminar).
 - b. Editar tipo: Permite la modificación del nombre para cada tipo.
 - c. Eliminar tipo: Permite eliminar un tipo de la base de datos.
 - d. Agregar tipo: Permitir la inserción de un tipo en la base de datos.
- 4) Gestionar los elementos, que incluye:
- a. Listar elementos: Despliega la información en una tabla, cada entrada incluye: nombre y año del elemento, autor, ubicación y acciones relacionadas (ver detalle, editar y eliminar).
 - b. Ver detalle: Permite mostrar la información completa de un elemento.

- c. Editar elemento: Permite la modificación de la información almacenada sobre cada elemento.
 - d. Eliminar elemento: Permite eliminar un elemento de la base de datos.
 - e. Agregar elemento: Permitir la inserción de un elemento en la base de datos.
- 5) Gestionar las panorámicas, que incluye:
- a. Listar panorámicas: Despliega la información en una tabla, cada entrada incluye: nombre del archivo, latitud y longitud, ubicación, imagen de la panorámica con enlaces incluidos y acciones relacionadas (ver detalle, editar y eliminar).
 - b. Ver detalle: Permite mostrar la información completa de una panorámica.
 - c. Editar panorámica: Permite la modificación de la información almacenada sobre cada panorámica.
 - d. Eliminar panorámica: Permite eliminar una panorámica de la base de datos.
 - e. Agregar panorámica: Permitir la inserción de una panorámica en la base de datos.
- 6) Gestionar los usuarios, que incluye:
- a. Listar usuarios: Despliega la información en una tabla, cada entrada incluye: nombre del usuario, fecha de creación, fecha de modificación y acciones relacionadas (editar y eliminar).
 - b. Editar usuario: Permite la modificación de la información almacenada sobre cada usuario (Sólo disponible cuando lo permiten los privilegios del usuario).
 - c. Eliminar usuario: Permite eliminar un usuario de la base de datos (Sólo disponible cuando lo permiten los privilegios del usuario).
 - d. Agregar usuario: Permite la inserción de un usuario en la base de datos (Sólo disponible cuando los privilegios del usuario lo permiten).

3.3 Perfiles de usuarios

Visitante: Son usuarios generales interesados en navegar por las diferentes secciones del portal para visitantes. No necesitan ningún tipo de credencial o autenticación para acceder, por lo que no guardan relación con la tabla users de la base de datos.

Administrador: Es el principal usuario de gestión del portal administrativo, tiene acceso a todas las funciones de administración y puede gestionar los demás perfiles de usuario, sus datos van almacenados en la tabla users.

Registrado: Son usuarios a los que se les permite acceder a todas las funciones administrativas, con la excepción de que no pueden gestionar perfiles de usuario diferentes del suyo o eliminar ningún tipo de información. Puede ser un pasante o aquella persona que va a estar en constante iteración con el panel de administración agregando la información necesaria, sus datos van almacenados en la tabla users.

A continuación se muestra en la tabla 7 las principales actividades que puede realizar cada tipo de usuario.

Módulo	Funcionalidad	Visitante	Administrador	Registrado
Administrar usuario	Registrar de usuario		X	
	Listar usuario		X	
	Modificar usuario		X	
	Eliminar Usuario		X	
Autor	Registrar autor		X	X
	Listar autor		X	X
	Modificar autor		X	X
	Eliminar autor		X	
Elemento, Zona o Tipo	Registrar elemento, zona o tipo		X	X
	Listar elemento, zona o tipo		X	X
	Modificar elemento, zona o tipo		X	X
	Eliminar elemento, zona o tipo		X	
Panorámica	Registrar panorámica		X	X
	Listar panorámica		X	X
	Modificar panorámica		X	X
	Eliminar panorámica		X	
Interfaz de Visitante	Visualizar el portal e imprimir información.	X	X	X
Base de Datos	Realizar respaldo de la base de datos.		X	

Tabla 5. Funcionalidades y Permisos de los Usuarios del Portal

Fuente: Los Autores

3.4 Planeación por funcionalidad

La metodología FDD recomienda que la construcción de cada grupo de funcionalidades dure como máximo 2 semanas.

Nro.	Descripción	Duración
1	Diseño lógico y físico de la base de datos	2 semanas
2	Diseño del portal para visitantes	2 semanas
3	Generación de la plantilla del portal para visitantes	2 semanas
4	Desarrollo de la página inicio	1 semana
5	Desarrollo de la página ¿Quiénes somos?	1 semana
6	Desarrollo de la página ¿Por qué somos patrimonio?	1 semana
7	Desarrollo de la página Historia	1 semana
8	Desarrollo de la página Conjunto	2 semanas
9	Desarrollo de la página Paseo Virtual	2 semanas
10	Desarrollo de la página Galería	2 semanas
11	Desarrollo de la página Recorridos	1 semana
12	Desarrollo de la página Documentos	1 semana
13	Adiestramiento de <i>CakePHP</i>	1 semana
14	Estructuración y diseño del módulo administrativo	1 semana
15	Desarrollo de las funcionalidades de administración de autor	2 semanas
16	Desarrollo de las funcionalidades de administración de zona	2 semanas
17	Desarrollo de las funcionalidades de administración de tipos de elementos	2 semanas
18	Desarrollo de las funcionalidades de administración de elementos	2 semanas
19	Desarrollo de las funcionalidades de administración de panorámicas	2 semanas
20	Desarrollo de las funcionalidades de administración de usuarios	1 semana
21	Pruebas	1 semana

Tabla 6. Lista de actividades

Fuente: Los Autores

3.5 Diseño de la base de datos

Dado el modelo global y luego de estudiar la lista de funcionalidades, se procede a modelar la base de datos, cuya construcción necesitó el estudio de las convenciones de *CakePHP*, requiriendo por ejemplo, el nombramiento de las tablas en plural.

El núcleo básico gira entorno a una tabla llamada *Objetos*, que almacenará tanto elementos (obras de arte, edificaciones, entre otros) como panorámicas (que integran el paseo virtual), el tipo de objeto se determina por las relaciones que guarda con el resto de las tablas, sabremos por ejemplo que un objeto relacionado con la tabla *Metaelementos* será un elemento, mientras que un objeto relacionado con la tabla *Objetos_proximos* representará una panorámica con enlaces.

La tabla *Objetos_proximos* modela una relación muchos a muchos (*hasAndBelongsToMany*) entre la tabla *Objetos* y una tabla virtual llamada *Proximos*, que es en realidad la misma tabla *Objetos*; la construcción de la tabla *Objetos_proximos* permite la relación circular que genera el paseo virtual al enlazar una panorámica con otras, permitiendo que *CakePHP* entienda la relación muchos-a-muchos de la forma correcta.

El resto de las tablas sigue los lineamientos de modelado y construcción de base de datos.

Tabla	Descripción	Relación
Objetos	Incluye cualquier tipo de objeto del portal, incluye cada uno de los elementos del conjunto, es decir, las obras de arte, paisajismos, entre otros. Y las fotos panorámicas.	Zonas, Tipopestanas
Metaelementos	Incluye información específica para los objetos que son elementos del conjunto, es decir, las obras de arte, paisajismos, entre otros.	Tipos, Objetos
Tipos	Incluye lo que habíamos llamado subtipos, es decir, los subtipos de elementos, por ejemplo dentro de las obras de arte están los murales, los mosaicos, las esculturas, entre otros.	
Tipopestanas	Incluye un tipo de clasificación pero superior, son las pestañas del conjunto, es decir, obras de arte, paisajismos, entre otros.	
Objetos_proximos	Son los enlaces que tiene cada panorámica, una tabla que incluye una entrada por cada enlace que tiene una panorámica, sean enlaces a las fichas del autor o a otra panorámica.	Objetos, objetos_proximos
Zonas	Incluye las diferentes zonas, por ejemplo plaza cubierta del rectorado.	
Autores	Incluye la lista de autores.	
Autores_objetos	Incluye las relaciones entre un autor y un objeto.	Objetos, Autores
Images	Incluye la información de las imágenes para cada objeto, incluidas las panorámicas y las múltiples imágenes que puede tener un elemento del conjunto.	Objetos
Campos	Incluye campos de información adicionales para cada elemento	objetos
Users	Incluye usuarios del sistema en representación de las credenciales de acceso	

Tabla 7. Descripción de la Base de Datos y las Relaciones

Fuente: Los Autores

En la figura 14, se define el modelo entidad / relación que permite visualizar la estructura de la base de datos que se utilizó en el desarrollo del módulo.

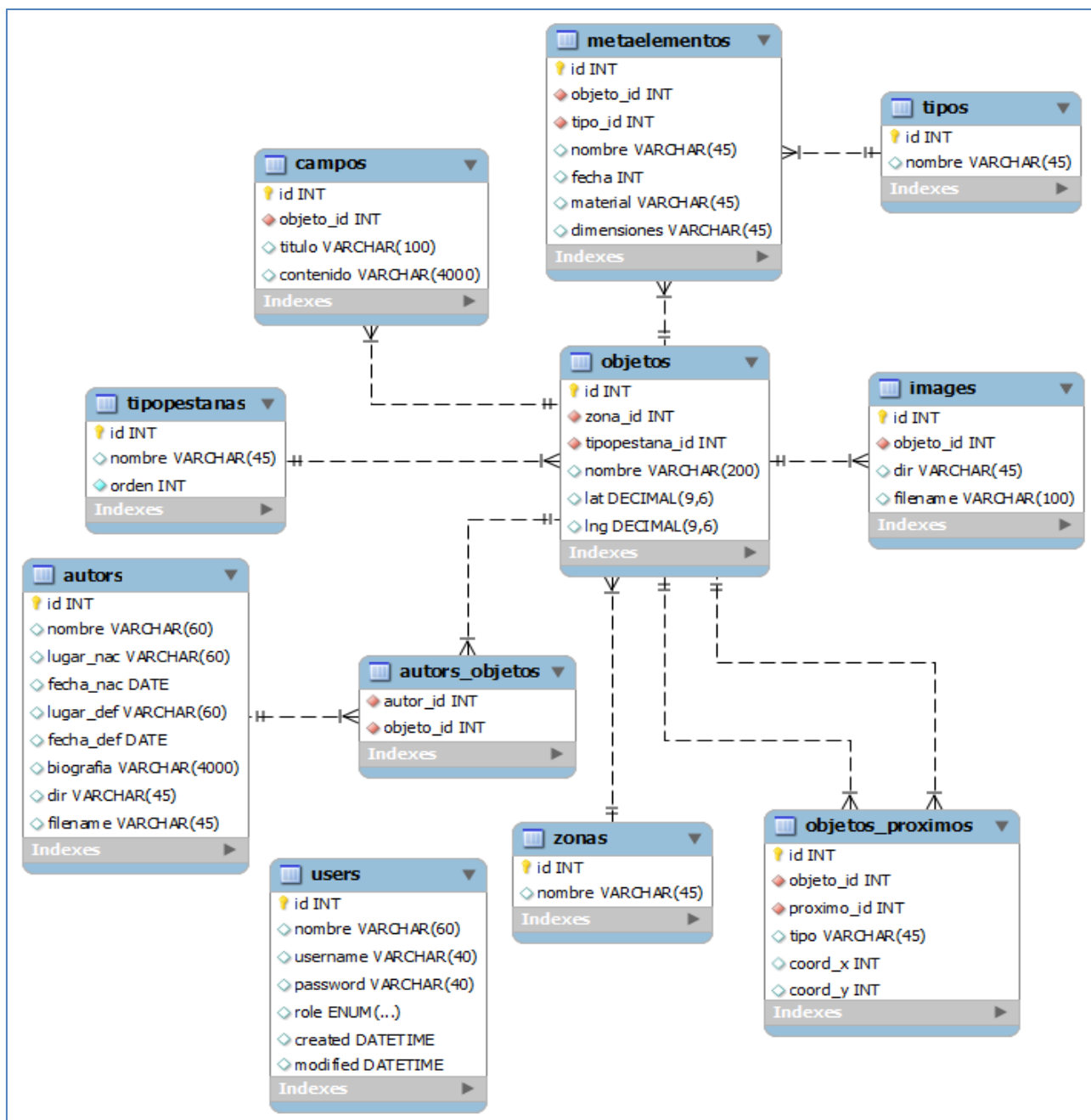


Figura 14. Modelo de Datos del Portal Web

Fuente: Los Autores

3.6 Diseño por funcionalidad

Una vez que han sido listadas las funcionalidades de forma detallada, podemos evaluar el diseño de cada una de ellas.

Como ya sabemos, FDD utiliza diagramas de secuencia y de clases para describir cada funcionalidad, para este desarrollo el diagrama de clases no sufre ninguna modificación, por lo que se estudia el diagrama de clases del modelo global junto con diagramas de secuencia ajustados a este contexto.

Partiendo de los conjuntos de funcionalidades, muchos de los diagramas se han agrupado de acuerdo a sus similitudes, sobre todo los correspondientes a funcionalidades de administración o gestión de contenido, veamos a continuación los diagramas de secuencia por funcionalidad:

Funcionalidades a nivel del usuario con perfil visitante:

- 1) Diseño sobre la funcionalidad de las “Ficha Informativa” de los elementos:

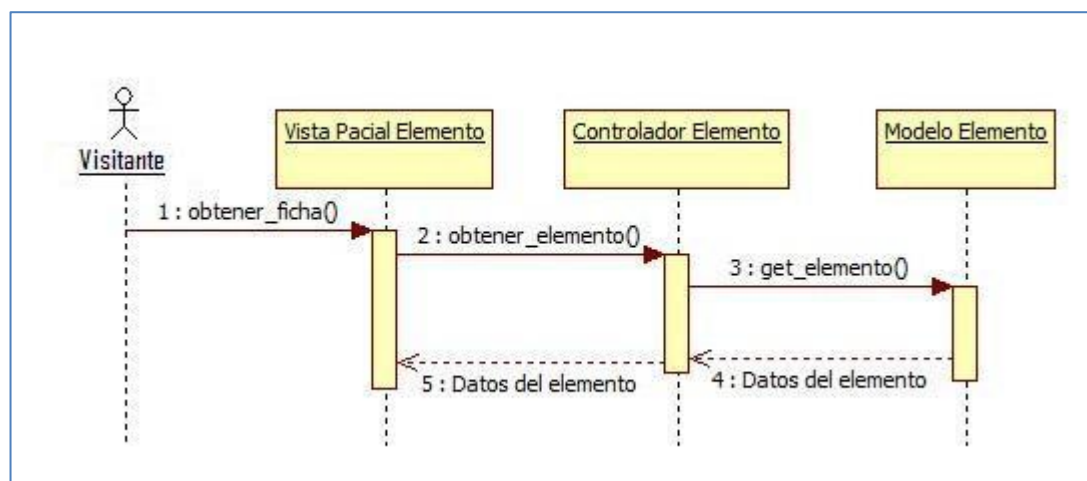


Figura 15. Diagrama de Secuencia: “Ficha Informativa” de los Elementos

Fuente: Los Autores

2) Diseño sobre la funcionalidad “Ficha de Autor”:

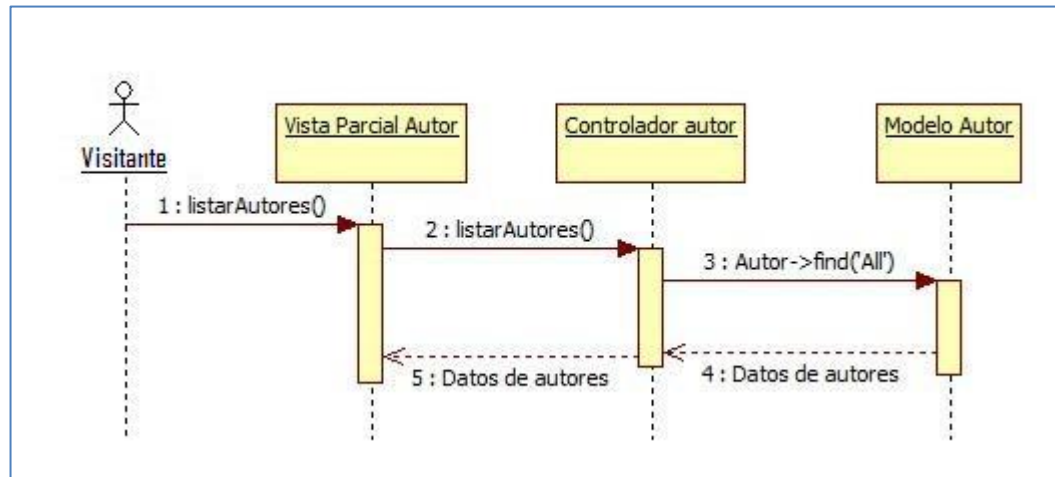


Figura 16. Diagrama de Secuencia: “Ficha de Autor”

Fuente: Los Autores

3) Diseño sobre la funcionalidad del “Paseo Virtual”:

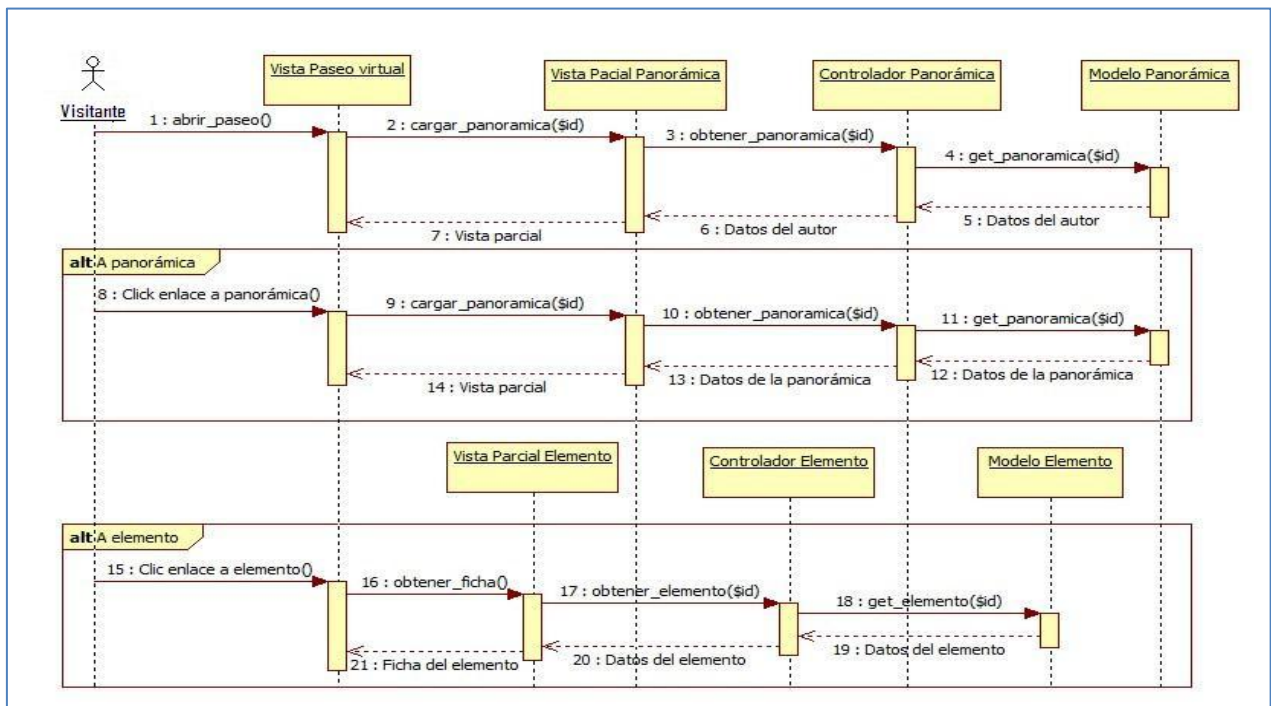


Figura 17. Diagrama de Secuencia: Paseo Virtual

Fuente: Los Autores

4) Diseño sobre la funcionalidad del “Mapa y sus elementos”:

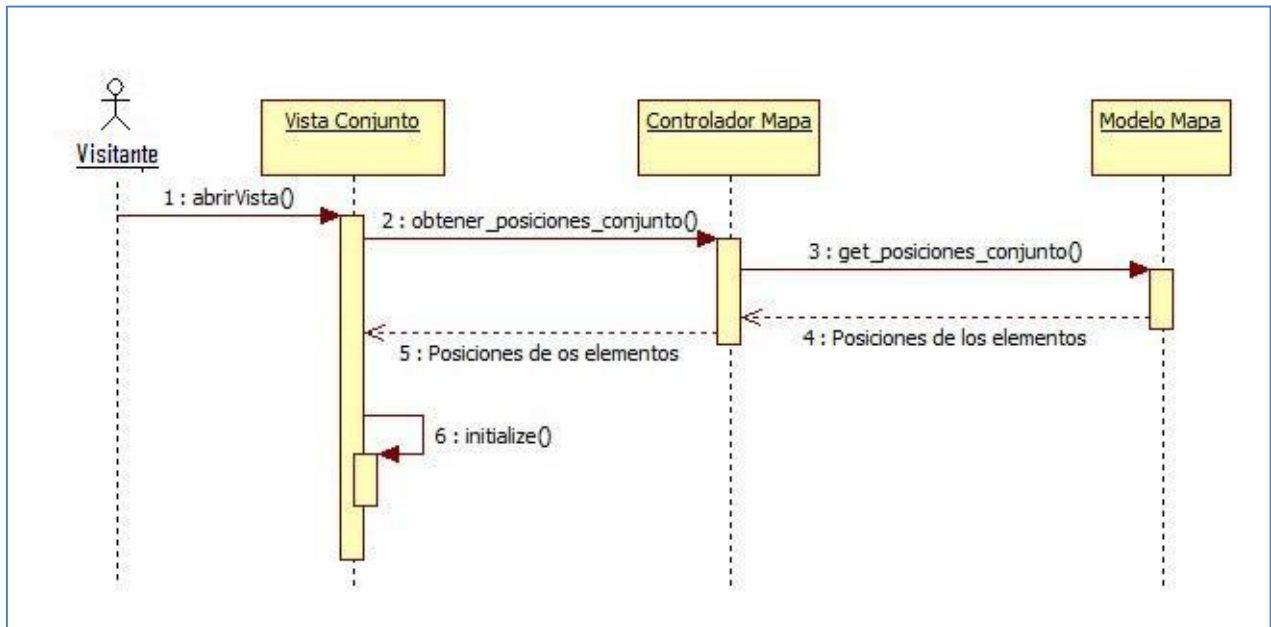


Figura 18. Diagrama de Secuencia: Mapa y sus elementos

Fuente: Los Autores

5) Diseño sobre la agrupación de los elementos como un “Conjunto Arquitectónico”:

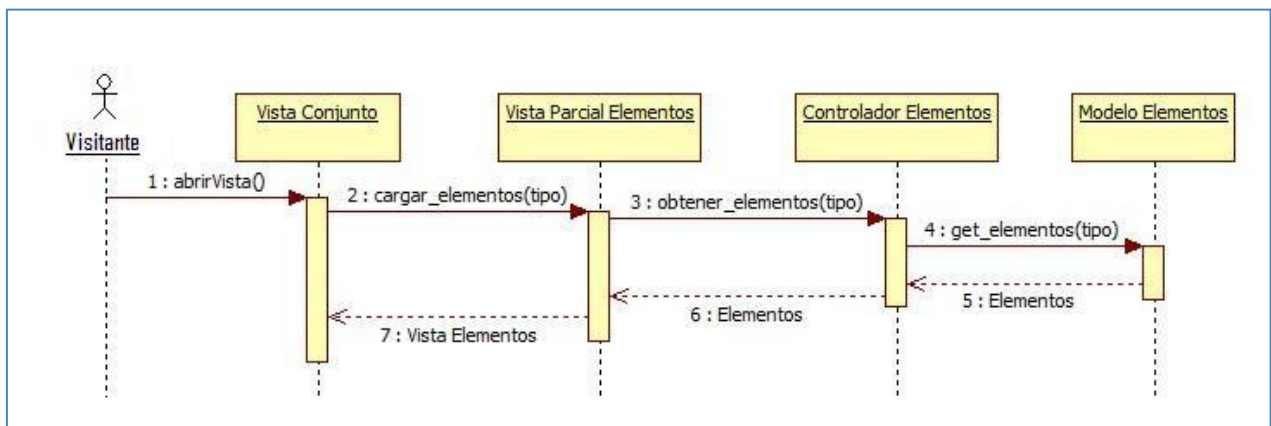


Figura 19. Diagrama de Secuencia: Conjunto Arquitectónico

Fuente: Los Autores

6) Diseño sobre la “galería de imágenes con ordenamiento”:

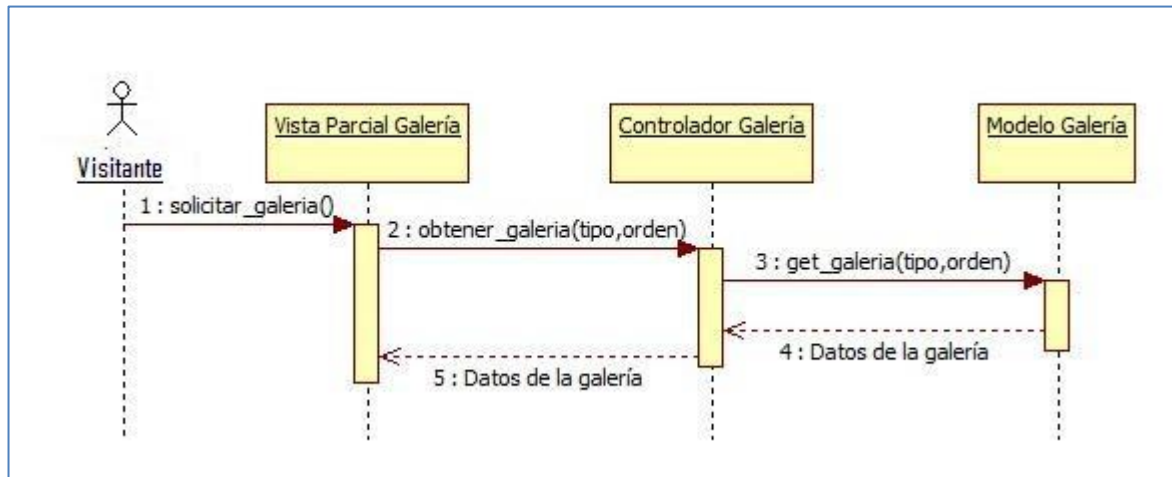


Figura 20. Diagrama de Secuencia: Galería de Imágenes con Ordenamiento

Fuente: Los Autores

Funcionalidades a nivel del usuario con perfil administrador o registrado: Son todas aquellas funcionalidades asociadas al portal de administración y que permite la gestión de contenido donde el perfil registrado está restringido en eliminar y gestionar usuarios, se lista a continuación:

1) Diseño sobre la funcionalidad de listar:

El diseño de esta funcionalidad se comporta similar en todos los casos listar, de modo que el siguiente diagrama de secuencia aplica para: autores, zonas, tipos de elementos, elementos, panorámicas y usuarios. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

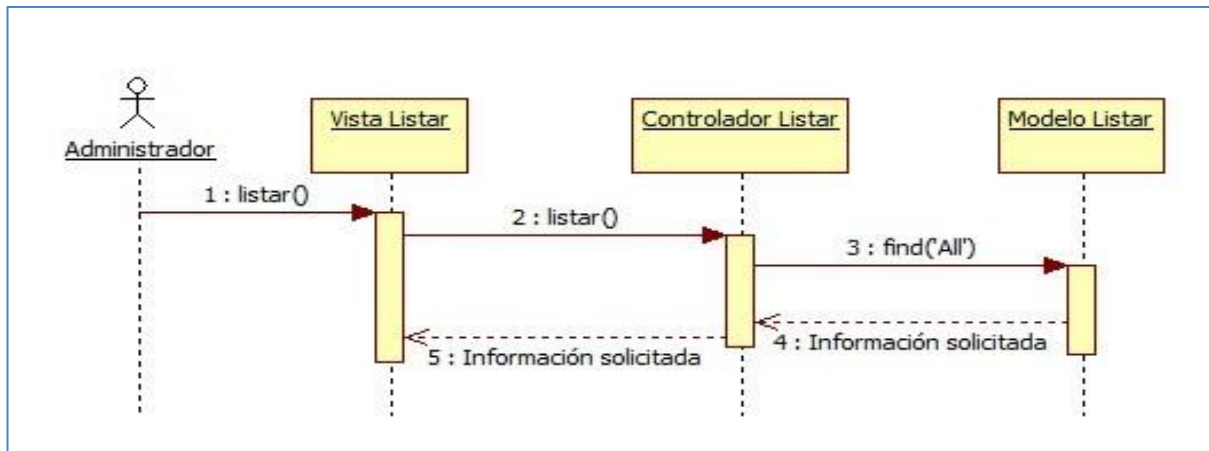


Figura 21. Diagrama de Secuencia: Listar

Fuente: Los Autores

2) Diseño sobre la funcionalidad de ver detalle:

Al igual que Listar, el diseño de esta funcionalidad se comporta similar en todos sus casos, el siguiente diagrama de secuencia aplica para: autores, zonas, tipos de elementos, elementos, panorámicas y usuarios. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

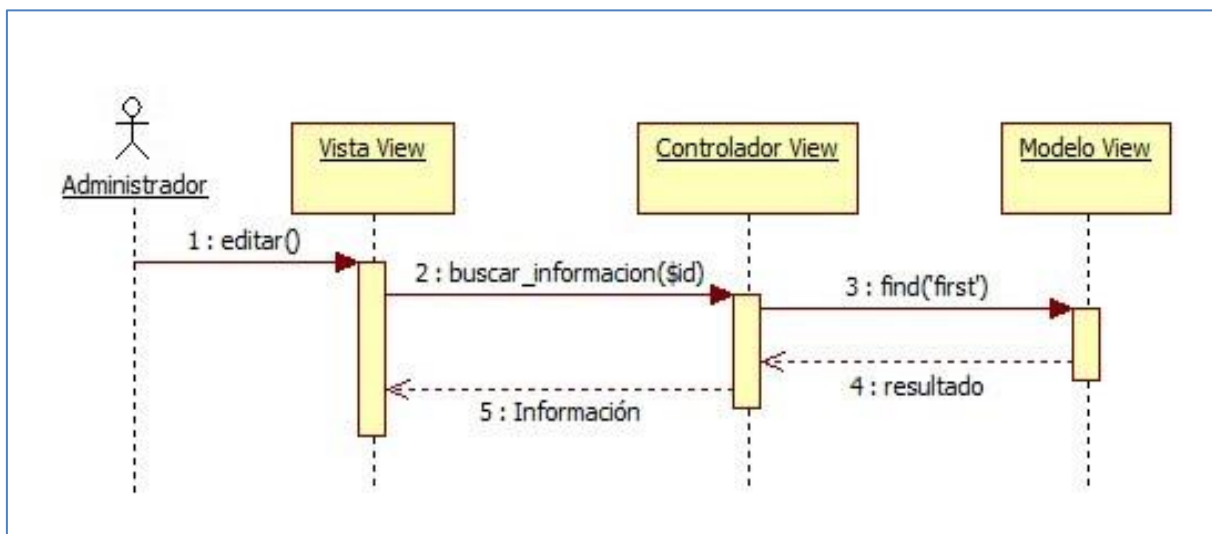


Figura 22. Diagrama de Secuencia: Ver Detalle

Fuente: Los Autores

3) Diseño sobre la funcionalidad de eliminar:

Nuevamente, dadas las características básicas de esta funcionalidad, es posible hacer el agrupamiento en un único diagrama, por lo que el siguiente diagrama expresa las funcionalidades de eliminar: autores, zonas, tipos de elementos, elementos, panorámicas y usuarios. Esta funcionalidad puede ser realizada por el perfil administrador.

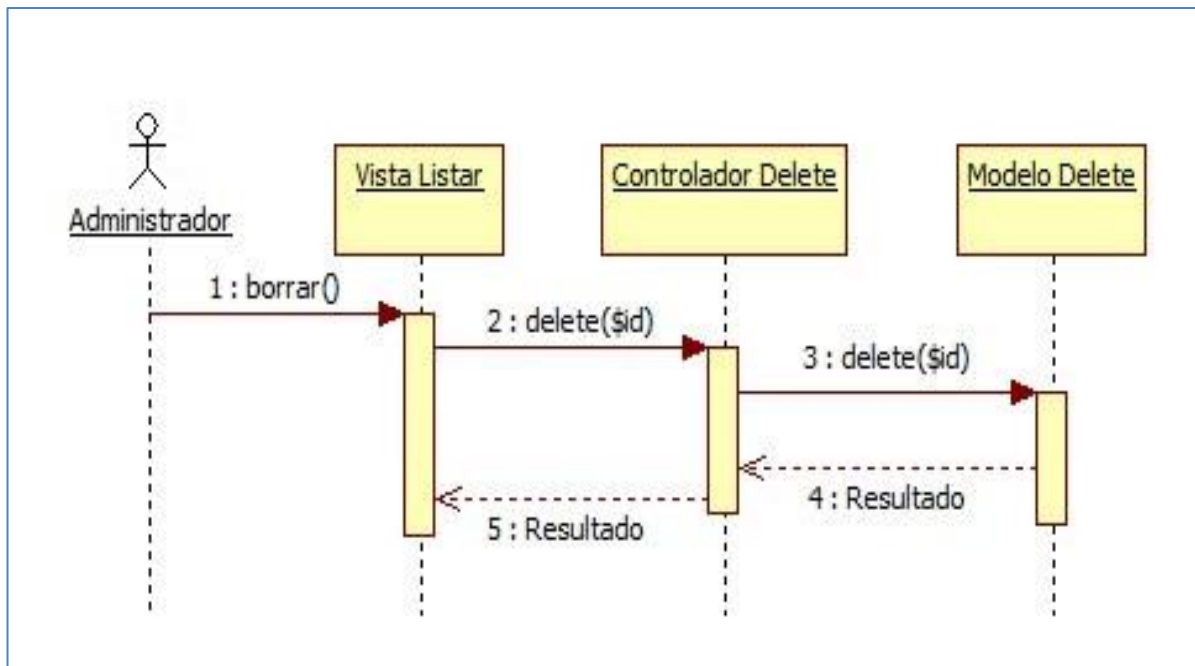


Figura 23. Diagrama de Secuencia: Eliminar

Fuente: Los Autores

4) Diseño sobre la funcionalidad de editar para los casos: autor, zona, tipo de elemento y usuario. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

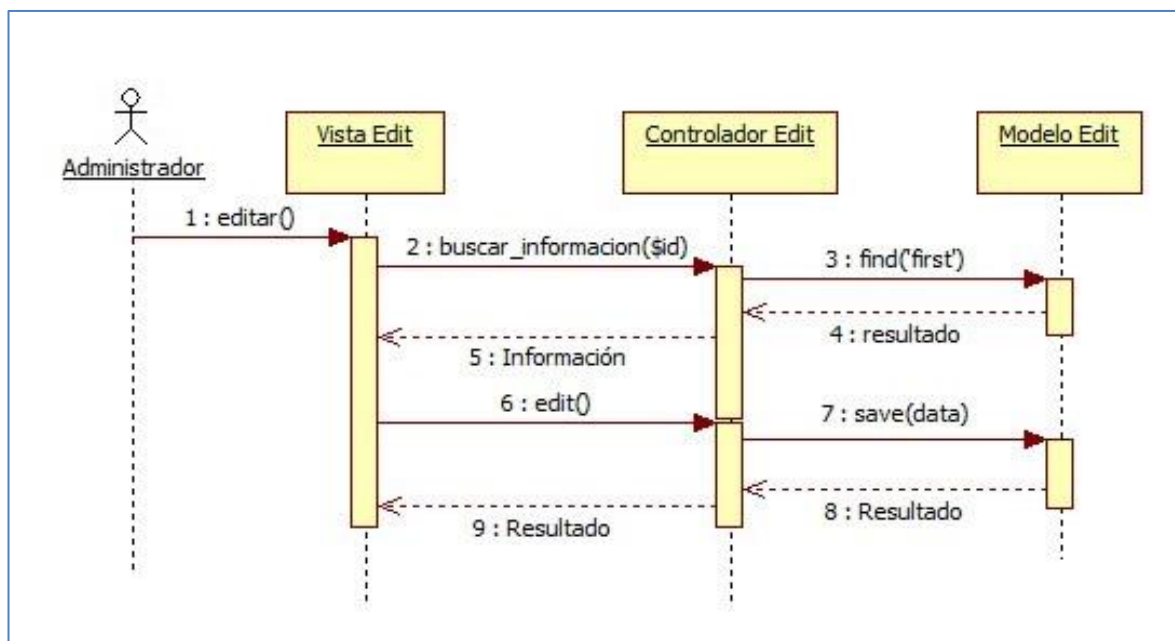


Figura 24. Diagrama de Secuencia Editar: Autor, zona, tipo de elemento y usuario

Fuente: Los Autores

- 5) Diseño sobre la funcionalidad de editar para los casos: panorámica y elemento. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

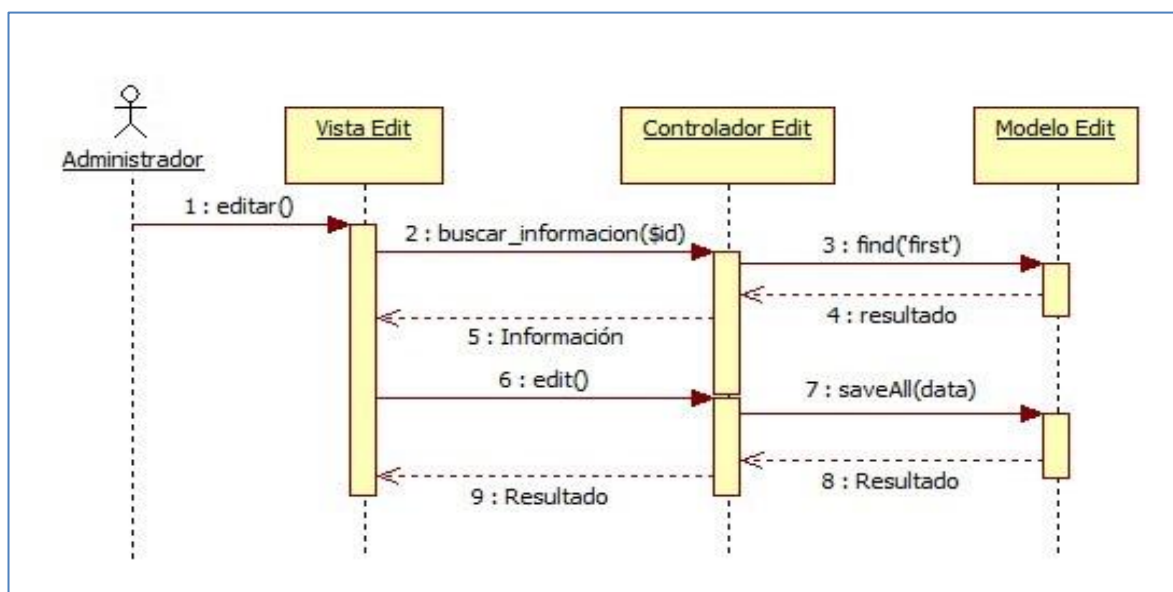


Figura 25. Diagrama de Secuencia Editar: Panorámica y elemento

Fuente: Los Autores

- 6) Diseño sobre la funcionalidad de agregar para los casos: Autor, zona, tipo de elemento y usuario. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

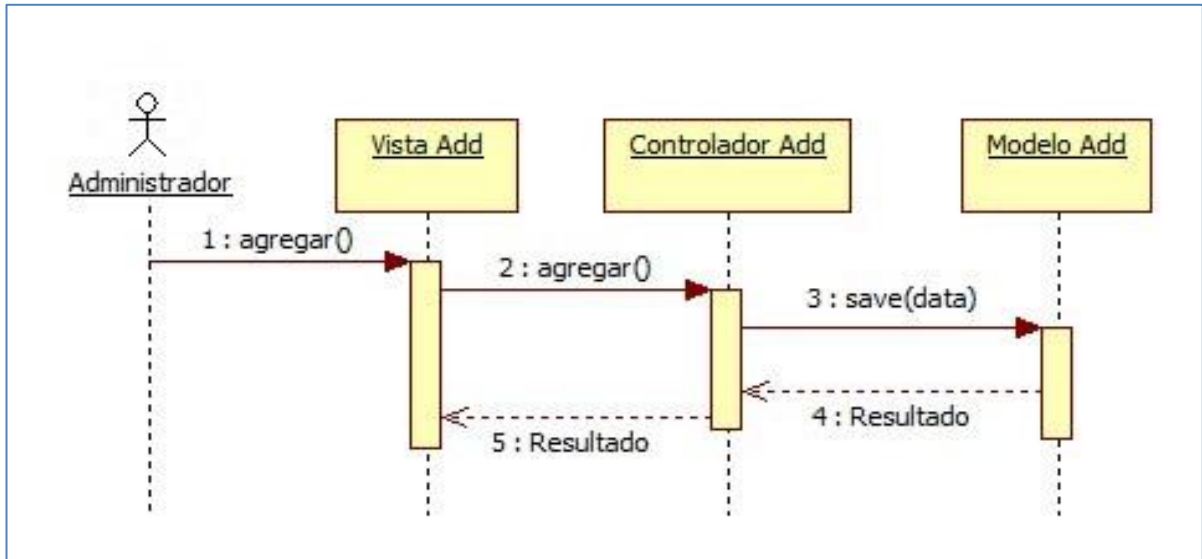


Figura 26. Diagrama de Secuencia Agregar: Autor, zona, tipo de elemento y usuario

Fuente: Los Autores

- 7) Diseño sobre la funcionalidad de agregar para los casos: Panorámica y elemento. Esta funcionalidad puede ser realizada tanto por perfil administrador como el perfil registrado.

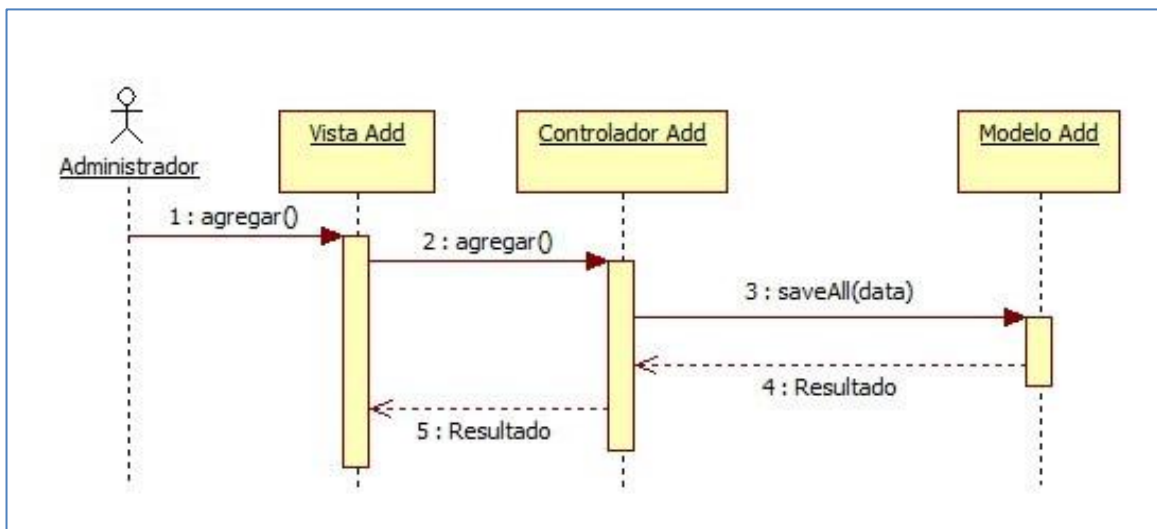


Figura 27. Diagrama de Secuencia Agregar: Panorámica y elemento

Fuente: Los Autores

3.7 Construcción por funcionalidad

Luego de estudiar detalladamente los diagramas de secuencia expuestos, podemos analizar algunos de los métodos involucrados en el cumplimiento exitoso de las funcionalidades listadas. De acuerdo al desarrollo, es posible afirmar que varias de las funcionalidades se llevan a cabo implementando secciones de código similares, pero trataremos de extraer características significativas de cada una.

Tal como lo indica la metodología FDD, hemos separado dos grandes conjuntos superiores (las funcionalidades para el perfil general de visitante y las funcionalidades para el perfil de usuario restringido que son utilizadas por el usuario registrado y el usuario administrador), cada una de ellas cuenta con subconjuntos que a su vez se fragmentan en funcionalidades básicas; la construcción de una funcionalidad básica se contempla que tiene una duración máxima de dos semanas. Ahora bien, considerando el esquema propuesto, veamos la construcción de las actividades de acuerdo a cada conjunto superior:

Funcionalidades a nivel del perfil visitante: La construcción del portal para visitantes se implementa básicamente en PHP y sigue los lineamientos del patrón MVC, garantizando modularidad en el desarrollo de las funcionalidades actuales y extensibilidad del código hacia posibles nuevas funcionalidades.

1) Sobre la construcción de las fichas informativas de los elementos:

La ficha informativa de un elemento se representa mediante una vista denominada “Vista parcial”, ésta es usualmente insertada de forma dinámica en una vista principal mediante AJAX, pero también puede reutilizarse en un contexto más amplio, como por ejemplo una ventana de *jQuery FancyBox*.

En síntesis, la ficha obtiene los datos del controlador, quien es el encargado de dialogar con el modelo para obtener la información requerida de la base de datos, esta información se almacena en un objeto del tipo elemento.

En esta funcionalidad se expresa la necesidad de almacenar algunos campos estáticos establecidos por el cliente, que permanecerán vigentes en todas las fichas, sin embargo, se destaca la exigencia de que la ficha tenga otros campos dinámicos, lo que significó la elaboración de la tabla Campos en la base de datos, dedicada para tal fin.

Por otra parte, la ficha también incluye una galería de imágenes, para ello se dispuso de la tabla Images, que almacena la ruta de cada imagen junto con su nombre. El uso de una galería de imágenes dentro de la ficha, acarrea la utilización de la librería *jQuery Galleria*, que debe ser inicializada cuando se carga cada ficha; en la figura 28 puede apreciarse un fragmento de la función AJAX que hace la carga dinámica del contenido, junto con la inicialización de la galería.

```
if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
    /* Actualización del contenido */
    document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
    /* Pre-configurar Galleria */
    Galleria.configure({
        wait: 'true',
        showInfo: false,
        _toggleInfo: false
    });
    /* Inicializar Galleria */
    Galleria.run('#galleria');
    /* Habilitar Fullscreen en Galleria */
    Galleria.ready(function() {
        var gallery = this;
        $('#fullscreen').click(function() {
            gallery.enterFullscreen();
        });
    });
    /* Hacer scroll hacia la sección actualizada */
    $('html, body').animate({
        scrollTop: $('#txtHint').offset().top
    }, 2000);
    ajustar_pantalla();
}
```

Figura 28. Función AJAX que Realiza la Carga Dinámica

Fuente: Los Autores

Ahora bien, el modelo obtiene los datos de los campos e imágenes y crea dos arreglos, que posteriormente forman parte del objeto elemento, tal detalle se puede apreciar en la figura 29.

```

/* Obtención de los campos para este elemento */
$result = mysql_query("SELECT titulo, contenido FROM campos WHERE objeto_id = '". $q. "'");
$campos = array();
$i = 0;
while($row = mysql_fetch_array($result))
{
    $campos[$i]['titulo'] = $row['titulo'];
    $campos[$i]['contenido'] = $row['contenido'];
    $i++;
}

/* Obtención de las rutas de imágenes para este elemento */
$result = mysql_query("SELECT dir, filename FROM images WHERE objeto_id = '". $q. "'");
$images = array();
$i = 0;
while($row = mysql_fetch_array($result))
{
    $images[$i]['ruta'] = str_replace("\\", "/", '..\admin_cuc\app\webroot'.'/'.$row['dir'].'/' . $row['filename']);
    $i++;
}

```

Figura 29. Código de obtención de campos y rutas del elemento en PHP

Fuente: Los Autores

2) Sobre la construcción de las fichas de los autores:

La construcción de las fichas biográficas de autores sigue un patrón similar al de las fichas informativas vistas descripta anteriormente. Nuevamente se representa mediante una vista llamada “Vista parcial” con las mismas características de la vista parcial para la ficha informativa de un elemento.

Su funcionamiento es el mismo, la ficha obtiene los datos del controlador, quien dialoga con el modelo para obtener la información requerida, esta información se almacena en un objeto del tipo autor. La particularidad del modelo radica en que no sólo se buscan los datos del autor como tal, sino que también se solicitan una lista de todos los elementos que pertenecen a este autor, solicitándolos a la tabla Objetos; el procedimiento que se observa en la figura 30.

```

/* Obtención de los datos propios de este elemento */
$result = mysql_query("SELECT nombre, lugar_nac, fecha_nac, lugar_def, fecha_def, biografia FROM autores WHERE id = '". $q.
"");
while($row = mysql_fetch_array($result))
{
    $autor_result = mysql_query("SELECT objetos.id, objetos.nombre FROM autores INNER JOIN autores_objetos ON autores.id
= autores_objetos.autor_id INNER JOIN objetos ON autores_objetos.objeto_id = objetos.id WHERE autores.id = '". $q. "");
    $sobras_autor = array();
    $i = 0;
    while($autor_row = mysql_fetch_array($autor_result))
    {
        $sobras_autor[$i]['id'] = $autor_row['id'];
        $sobras_autor[$i]['nombre'] = $autor_row['nombre'];
        $i++;
    }
    $autor_solicitado = new autor($row['nombre'], $row['lugar_nac'], $row['fecha_nac'], $row['lugar_def'], $row['fecha_def'],
    $row['biografia'], $sobras_autor);
}

```

Figure 30. Código de obtención de datos del elemento en PHP

Fuente: Los Autores

3) Sobre la construcción del paseo virtual:

Lo primero que cabe destacar en esta actividad es que involucra la inclusión de la librería *jQuery Virtual Tour*, que construye un *viewport* a partir de una imagen y una lista de enlaces. Un *viewport* no es más que un contenedor que simula la interfaz del paseo virtual, en donde es posible desplazarnos dentro de las panorámicas usando el ratón. El *jQuery Virtual Tour* está compuesto por dos funciones principales, la función `panorama()` y la función `advancedPanorama()`.

La función `panorama()` es la primera en ser llamada, en ella se procesa la imagen y se construye el *viewport* con la interfaz de visualización del panorama, es decir, es la que empotra la imagen y nos permite mover el campo de visión, limitado por el cuadro contenedor. Para garantizar la sensación de desplazamiento circular en las panorámicas y lograr continuidad de movimiento, esta función duplica la imagen y la coloca en los extremos de la original, lo que significa que si por ejemplo, movemos el campo de visión hacia la derecha, hasta un punto en el que la panorámica termina, el panorama lo soluciona colocando inmediatamente después la copia de la imagen. La figura 31 muestra las instrucciones que utiliza `panorama()` en los eventos del ratón para mover el campo de visión de la panorámica.

```

panoramaViewport.mousedown(function(e){
    if (!bMouseMove) {
        bMouseMove = true;
        mouseMoveStart = e.clientX;
    }
    return false;
}).mouseup(function(){
    bMouseMove = false;
    mouseMoveStart = 0;
    return false;
}).mousemove(function(e){
    if (bMouseMove){
        var delta = parseInt((mouseMoveStart - e.clientX)/30);
        if ((delta>10) || (delta<-10)) {
            var newMarginLeft = parseInt(panoramaContainer.css('marginLeft')) + (delta);
            if (settings.mode_360) {
                if (newMarginLeft > 0) {newMarginLeft = -elemWidth;}
                if (newMarginLeft < -elemWidth) {newMarginLeft = 0;}
            } else {
                if (newMarginLeft > 0) {newMarginLeft = 0;}
                if (newMarginLeft < -elemWidth) {newMarginLeft = -elemWidth;}
            }
            panoramaContainer.css('marginLeft', newMarginLeft+'px');
        }
    }
}).bind('contextmenu',function(){return false;});

```

Figure 31. Instrucción que utiliza panorama()

Fuente: Los Autores

Por su parte, la función `advancedPanorama()` es la segunda en ser llamada y permite colocar en el *viewport* los enlaces, ver figura 32 dichos enlaces permiten:

- ✓ Abrir la ficha de un elemento que se encuentra en ese lugar y se aprecia en la panorámica. La carga se realiza reutilizando la función AJAX mencionada anteriormente.
- ✓ Cargar una nueva panorámica con posibles nuevos enlaces incluidos.

La carga de las panorámicas, bien sea a través de un enlace en el *viewport* o algún otro activador, se realiza mediante una función AJAX y puede ser reutilizada tantas veces como se requiera. Es importante destacar que el procesamiento de la imagen hecho por la librería *jQuery Virtual Tour* se repite cada vez que cargamos una imagen, de modo que necesitamos llamar a la función de inicialización cada vez que hacemos una carga. Un fragmento del código de la función de carga puede verse en la figura 33.

```

/* Se agrega el enlace segun su tipo */
if (areaalt == "Ver detalle"){

    panoContainer.append("<a class='panorama-area area'+areaId+areaclass+'\" style='position: absolute; left: "+areacoordArray[0]+"px; top: "+areacoordArray[1]
]+\"px; width: "+(areacoordArray[2]-areacoordArray[0])+"px; height: "+(areacoordArray[3]-areacoordArray[1])+"px;' onmouseover='javascript:area_hover(\"+
areaId+\")' onmouseout='javascript:area_out(\"+areaId+\")' href='"+areahref+"' title='"+areaalt+"'+>"+areatext+"</a>");

    panoContainer.append("<a class='panorama-area area'+areaId+areaclass+'\" style='position: absolute; left: +(panoImgWidth+parseInt(areacoordArray[0]))+
\"px; top: "+areacoordArray[1]+"px; width: "+(areacoordArray[2]-areacoordArray[0])+"px; height: "+(areacoordArray[3]-areacoordArray[1])+"px;'
onmouseover='javascript:area_hover(\"+areaId+\")' onmouseout='javascript:area_out(\"+areaId+\")' href='"+areahref+"' title='"+areaalt+"'+>"+areatext+"</a>");
}else{

    panoContainer.append("<a class='panorama-area area'+areaId+areaclass+'\" style='position: absolute; left: "+areacoordArray[0]+"px; top: "+areacoordArray[1]
]+\"px; width: 75px; height: 40px;' onmouseover='javascript:area_hover(\"+areaId+\")' onmouseout='javascript:area_out(\"+areaId+\")' href='"+areahref+"'
title='"+areaalt+"'+>"+areatext+"</a>");

    panoContainer.append("<a class='panorama-area area'+areaId+areaclass+'\" style='position: absolute; left: +(panoImgWidth+parseInt(areacoordArray[0]))+
\"px; top: "+areacoordArray[1]+"px; width: 100px; height: 60px;' onmouseover='javascript:area_hover(\"+areaId+\")' onmouseout='javascript:area_out(\"+areaId+
\")' href='"+areahref+"' title='"+areaalt+"'+>"+areatext+"</a>");

}

```

Figura 32. Función advancedPanorama()

Fuente: Los Autores

```

if (xmlhttp.readyState==4 && xmlhttp.status==200)
{
    /* Actualización del contenido */
    document.getElementById("page").innerHTML=xmlhttp.responseText;
    /* Inicialización del virtual tour básico (Solo la panorámica y desplazamiento) */
    $("img.advancedpanorama").panorama({
        auto_start: 0,
        start_position: 1400
    });
    /* Inicialización del virtual tour (Enlaces)*/
    $(".panorama-viewport").advancedpanorama();
    /* Limpieza de elementos que pudieran estar abiertos */
    jQuery("#txtHint").html('');
    ajustar_pantalla();
}
}

```

Figure 33. Fragmento de función de carga

Fuente: Los Autores

4) Sobre la construcción del mapa con sus elementos:

La presentación del mapa se hace mediante *Google Maps*, lo que facilita ampliamente la tarea de colocar los puntos en el plano utilizando las primitivas que ofrece su API. El funcionamiento se estructura en un patrón MVC similar al visto anteriormente, con la diferencia de que los datos

no son cargados en una vista parcial y luego solicitados por una función AJAX, sino que son traídos desde un primer momento a la vista.

El procedimiento comienza con la llamada al controlador para que contacte con el modelo y busque la lista de elementos presentes en la base de datos, de allí se obtiene un arreglo de objetos de tipo posición, en donde cada objeto representa la abstracción de un elemento con su ubicación GPS (Latitud y longitud). Una vez que se obtiene el arreglo de objetos posición, se apilan los datos en un arreglo de *JavaScript* que utilizarán las funciones de *Google Maps*; para ver el detalle puede revisar la figura 34.

```

/*--- Inicio de manipulación del mapa ---*/
<?php
$pos = obtener_posiciones_conjunto(); /* Obtiene los elementos con sus posiciones para ponerlas en el mapa (llamado al controlador) */
?>
var data = [];
<?php
foreach ($pos as $elem) { ?>
    data.push({id: '<?php echo $elem->id; ?>', nombre: '<?php echo $elem->nombre; ?>', tipo: '<?php echo $elem->super_tipo; ?>', ruta: '<?php echo $elem->ruta; ?>', lat: '<?php echo $elem->lat; ?>', lng: '<?php echo $elem->lng; ?>'});
<?php }?>
/*--- Fin de manipulación del mapa ---*/

```

Figure 34. Obtención de datos que utilizará las funciones de *Google Maps*

Fuente: Los Autores

Posteriormente se llama a la función `initialize()`, que se encarga de inicializar el mapa con los marcadores (elementos) que se almacenaron en el arreglo `data`. Por otro lado, pensando en la distinción de los diferentes tipos de elementos que son guardados en la base de datos (en la tabla `Tipopestanas`), se elaboró una función llamada `doFilter()` que permite filtrar los marcadores según los elementos que queramos visualizar.

5) Sobre la agrupación de los elementos como un conjunto arquitectónico:

En la sección Conjunto arquitectónico se obtienen las agrupaciones de elementos de acuerdo a su clasificación, para ello se almacenan en la base de datos (Tabla `Tipopestanas`) cada uno de los

tipos posibles, aunque cabe aclarar que el ámbito de este trabajo se limita a la presentación de las obras de arte de la colección Síntesis de las Artes Mayores.

La búsqueda de los grupos de elementos se realiza de forma similar a las estudiadas anteriormente, siguiendo el patrón MVC y creando una vista parcial con las imágenes de los elementos de la clasificación deseada.

Para representar la información utilizamos un collage con las imágenes de los elementos, en este caso el resultado del modelo es un arreglo de elementos que contiene la ruta de la imagen, el nombre y el id. El id asociado a cada imagen nos permite el enlace con la ficha del elemento, pudiendo reutilizar la función AJAX detallada anteriormente, que se encarga de mostrar la ficha de un elemento.

6) Sobre la construcción de la galería de imágenes con ordenamiento:

La galería de imágenes se desarrolló en una sección específicamente destinada para ello, en este particular se destaca el uso de la librería *jQuery FancyBox* para presentar las imágenes al hacer clic sobre ellas. El funcionamiento de la sección sigue patrón MVC tal como se ha utilizado en las páginas anteriores, existe una vista parcial, en donde se presentan las fotografías, que es posteriormente añadida a la sección Galería.

En este particular, destaca la posibilidad de ordenar las imágenes, dicha posibilidad viene prediseñada desde el query de búsqueda en el modelo, una vez que seleccionamos el ordenamiento deseado, la vista procede a llamar al controlador, que a su vez pedirá al modelo los datos de la tabla Images ordenados según el parámetro solicitado en la vista. Una versión resumida de los queries en el modelo puede observarse en la figura 35.

```

switch ($q1) {
    case 'obras':
        switch ($q2)
        {
            case 'nombre':
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY objetos.nombre ASC");
                break;
            case 'autor':
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY autor ASC");
                break;
            case 'zona':
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY zona ASC");
                break;
            case 'tipo':
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY tipo ASC");
                break;
            case 'fecha':
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY fecha ASC");
                break;
            default:
                $result = mysql_query("SELECT images.dir, images.filename, objetos.nombre, ... WHERE tipopestanas.id = 1 OR tipopestanas.id = 2 ORDER BY objetos.nombre ASC");
        }
        $i = 1;
        while($row = mysql_fetch_array($result, MYSQL_ASSOC))
        {
            $ruta = str_replace("\\", "/", '..\admin_cuc\app\webroot\'.'/.$row['dir'].'/'.$row['filename']);
            $galeria[$i] = new elemento_galeria($row['nombre'], $row['tipo'], $row['autor'], $row['zona'], $row['fecha'], $ruta);
            $i++;
        }
        break;
}

```

Figura 35. Querys utilizados en la consulta de base de datos

Fuente: Los Autores

Funcionalidades a nivel de administrador: La construcción del portal de administración de contenido se implementa en el *framework CakePHP* y sigue sus lineamientos y convenciones. Como rasgos generales podemos decir que *CakePHP* implementa también el patrón MVC, de modo que tenemos vistas, modelos y controladores, aunque en situaciones determinadas puede carecer de una vista o de un modelo. Por lo general el controlador incluye una función por cada vista y el modelo agrupa definiciones básicas sobre los datos que están en la base de datos, tales como validaciones y demás.

Ahora bien, para comprender mejor las siguientes actividades, vamos a identificar primero algunas de las funciones básicas de *CakePHP* que son usadas en el código implementado:

- ✓ ***find(all)***: devolverá un resultado.
- ✓ ***find(first)***: devuelve una matriz (posiblemente multiples) resultados
- ✓ ***find(list)***: devuelve una matriz indexada, útil para cualquier lugar que se necesite una lista como para rellenar las cajas de selección de entrada.

- ✓ **save():** primitiva que efectúa el guardado de los datos, retorna *true* en caso de éxito y *false* en caso contrario.
- ✓ **saveall():** usado para guardar múltiples registros individuales para un único modelo.
- ✓ **set():** se utiliza para pasar variables del controlador a la vista.

1) Sobre la construcción de gestionar autores (Operaciones de listar, ver detalle, editar, eliminar y agregar):

Listar: Vista (index) - controlador (index()) – modelo (Autor). La función index() del controlador utiliza la función find('all'), de manera que obtiene todos los autores de la tabla Authors y los pasa a la vista a través de la función set(). Luego la vista index construye una tabla con la información de los autores. En esta operación destacan los enlaces las obras del autor que son enlaces a la función viewElemento() que veremos más adelante.

Ver detalle: Vista (view) - controlador (view()) – modelo (Autor). La función view() del controlador utiliza la función find('first') con la condición de que el id sea igual al requerido. Luego la vista view presenta la información normalmente en una vista sencilla.

Editar: Vista (edit) - controlador (edit()) – modelo (Autor). La función edit() del controlador utiliza la función find('first') con la condición de que el id sea igual al requerido, los datos son pasados a la vista mediante la función set(). Luego la vista presenta un formulario con los campos llenos que es enviado nuevamente al controlador, en este punto se utiliza la función save() para actualizar el registro en la base de datos.

Eliminar: Vista (No tiene) - controlador (delete()) – modelo (Autor). La función edit() es casi tan sencilla como la función index(), el controlador utiliza la función delete() para eliminar el registro de la base de datos.

Agregar: Vista (add) - controlador (add()) – modelo (Autor). La vista muestra un formulario que el usuario debe llenar y enviar, una vez que lo envía, la función add() del controlador se encarga de utilizar la función save() para almacenar el nuevo registro en la base de datos.

2) Sobre la construcción de gestionar zonas (Operaciones de listar, editar, eliminar y agregar):

Listar: Vista (index) - controlador (index()) – modelo (Zona). Funciona de la misma forma que el listar de la iteración 7. La función index() del controlador utiliza la función find('all'). Luego la vista index construye una tabla con la lista de las zonas. En esta operación destacan los enlaces a obras y panorámicas en la zona, que son enlaces a la función viewElemento() y viewPanoramica() respectivamente.

Editar: Vista (edit) - controlador (edit()) – modelo (Zona). Funciona de la misma forma que el editar de la iteración 7. Edit() Utiliza la función find('first') con la condición de que el id sea igual al requerido, se usa la función set() para pasar los datos. Luego la vista presenta el formulario lleno que es enviado nuevamente al controlador, finalmente se efectúa la función save().

Eliminar: Vista (No tiene) - controlador (delete()) – modelo (Zona). El controlador utiliza la función delete() para eliminar el registro de la base de datos.

Agregar: Vista (add) - controlador (add()) – modelo (Zona). La vista muestra un formulario que el usuario debe llenar y enviar. La función add() del controlador se encarga de ejecutar la función save() una vez que tiene los datos.

3) Sobre la construcción de gestionar zonas (Operaciones de listar, editar, eliminar y agregar):

Listar: Vista (index) - controlador (index()) – modelo (Tipo). Funciona de la misma forma que el listar de la funcionalidad anterior. La función index() utiliza la función find('all'). Luego la vista index construye una tabla con la lista de los tipos.

Editar: Vista (edit) - controlador (edit()) – modelo (Tipo). Es muy similar al editar zonas, edit() utiliza la función find('first') con la condición de que el id sea igual al requerido, se usa la función

set() para pasar los datos. Luego la vista presenta el formulario lleno que es enviado nuevamente al controlador, finalmente se efectúa la función save().

Eliminar: Vista (No tiene) - controlador (delete()) – modelo (Tipo). Inicialmente se chequea que no hayan elementos asociados a este tipo, esto se hace utilizando la función find('all') en la tabla Metaelementos, con la condición de que tengan el tipo_id actual. En caso de no tener elementos asociados, el controlador utiliza la función delete() para eliminar el registro de la base de datos.

Agregar: Vista (add) - controlador (add()) – modelo (Zona). La vista muestra un formulario que el usuario debe llenar y enviar. La función add() del controlador utiliza la función save() para efectuar el guardado.

4) Sobre la construcción de gestionar elementos (Operaciones de listar, ver detalle, editar, eliminar y agregar):

Listar: Vista (listar_elementos) - controlador (listarElementos()) – modelo (Objeto). La función listarElementos() del controlador utiliza la función find('all') con la condición de que el tipopestanas_id sea diferente de panorámica. Utiliza la función set() para pasar los datos. Luego la vista construye una tabla con la lista de los elementos.

Ver detalle: Vista (view_elemento) - controlador (viewElemento()) – modelo (Objeto). La función viewElemento() del controlador primero utiliza la función find('first') con la condición de que el id sea igual al del objeto buscado, posteriormente utiliza la función find('first') para encontrar el tipo asociado al Metaelemento de ese objeto.

Editar: Vista (editelemento) - controlador (editelemento()) – modelo (Objeto). La función editelemento() del controlador utiliza la función find('first') con la condición de que el id sea igual al requerido. Luego la vista presenta el formulario lleno que es enviado nuevamente al controlador, este se encarga de gestionar esos datos: inicialmente se construye el nombre del objeto concatenando el nombre del elemento, su año de creación y el nombre del autor; la figura 36 muestra el fragmento de código asociado; luego, se pasa a validar que imágenes se

conservarán de las que ya estaban asociadas, para esto es necesario comparar las enviadas por el usuario con las que estaban almacenadas; la figura 37 muestra el código asociado; finalmente se efectúa la función saveall().

```
/* Construccion del nombre del objeto (Tabla Objetos) que es el que aparece en la mayoria de las listas de seleccion */
$principalAutor = $this->Objeto->Autor->findById($this->request->data['Objeto']['Autor'][0]);
$this->request->data['Objeto']['nombre'] = $this->request->data['Metaelemento']['nombre']. " (".$this->request->data[
'Metaelemento']['fecha'].") "- " ".$principalAutor['Autor']['nombre'];
/* ----- */
```

Figura 36. Construcción del nombre del objeto

Fuente: Los Autores

```
$imagenes = $this->Objeto->Image->find('all',array('fields' => array('Image.id','Image.dir','Image.filename'),'conditions' =>
array('Image.Objeto_id' => $id)));
//echo debug($imagenes);

$ NUEVASIMAGES = $this->data['Image']; /* Lista de Imagenes que proporcione el usuario (Anteriores que no quiere eliminar y
nuevas para insertar) */
$eliminable = true;
$cnt = 0; /* Contador simple */
$imgs['Image'] = array();
/* Recorremos los registros de la tabla Images relacionados con este elemento, para compararlos con los suministrados por el
usuario. Los registros que el usuario no incluyo, seran eliminados de la BD (Despues de guardar para preservar la
consistencia) */
foreach($imagenes as $image):
    foreach($nuevasimagenes as $nuevaimage):
        if ($nuevaimage['nuevo'] == 'false'){ /* Evitar comparar $nuevaimage['id'] si no existe el elemento */
            if ($nuevaimage['id'] == $image['Image']['id']){
                $eliminable = false; /* En principio todos las imagenes se eliminaran (esta variable es true por defecto). Si
el usuario no las elimino en el formulario, entonces no se eliminaran */
            }
        }
    endforeach;
    /* Se llena el array imgs con las que se van a eliminar */
    if ($eliminable == true){
        $imgs['Image'][$cnt]['id'] = $image['Image']['id'];
        $imgs['Image'][$cnt]['ruta'] = str_replace("\\", "/", $this->webroot.$image['Image']['dir'].'/'.$image['Image']['
filename']);
        $cnt++;
    }
    /* ----- */
    $eliminable = true;
endforeach;

/* ----- */

/* Limpiamos el arreglo de los datos a guardar, quitando todas las imagenes que ya estan en la BD y no necesitan ser guardadas
nuevamente. No es posible actualizar una imagen tal como lo hacemos con otros registros de texto, ya que la informacion de
ubicacion en el origen y demas no esta presente. */
foreach($imagenes as $image):
    foreach($this->request->data['Image'] as $key => $nuevaimage):
        if ($nuevaimage['nuevo'] == 'false'){ /* Evitar comparar $nuevaimage['id'] si no existe el elemento */
            if ($nuevaimage['id'] == $image['Image']['id']){
                unset($this->request->data['Image'][$key]); /* Se limpia el registro del array con los datos a guardar, lo que
quiere decir que el registro en la BD y la imagen quedan intactos */
            }
        }
    endforeach;
endforeach;
```

Figura 37. Validar asociación de imágenes

Fuente: Los Autores

Eliminar: Vista (No tiene) - controlador (deleteelemento()) – modelo (Objeto). El controlador utiliza la función deleteelemento() para eliminar el registro de la base de datos.

Agregar: Vista (add) - controlador (add()) – modelo (Objeto). Inicialmente son pasados a la vista las listas de las zonas, los tipopestanas, los autores y los tipos, mediante la función set(), luego de consultas con find('list'). Posteriormente la vista muestra un formulario que el usuario debe llenar y enviar. La función add() del controlador se encarga de ejecutar la función saveall() una vez que tiene todos los datos.

5) Sobre la construcción de gestionar panorámicas (Operaciones de listar, ver detalle, editar, eliminar y agregar):

El desarrollo de este punto es un tanto diferente a los demás, sobretodo porque su modelado demandó un comportamiento más complejo; cabe destacar la relación circular en los enlaces panorámica-panorámica, cuyo comportamiento fue modelado de acuerdo a las convenciones y normas de *CakePHP*. La solución surge de crear la tabla Objetos_proximos, que modela una relación muchos a muchos (hasAndBelongsToMany) entre la tabla Objetos y una tabla virtual llamada Proximos, que es en realidad la misma tabla Objetos, de modo que *CakePHP* entiende la relación muchos-a-muchos como cualquier otra.

Listar: Vista (listar_panoramicas) - controlador (listarPanoramicas()) – modelo (Objeto). La función listarPanoramicas() del controlador utiliza la función find('all') con la condición de que el tipopestanas_id sea igual a panorámica. Utiliza la función set() para pasar los datos. Luego la vista construye una tabla con la lista de las panorámicas.

La función listarPanoramicas() del controlador utiliza la función find('all') con la condición de que el tipopestanas_id sea igual a panorámica. Utiliza la función set() para pasar los datos. Luego la vista construye una tabla con la lista de las panorámicas, de ella destaca la carga de la imagen con los enlaces embebidos en cada entrada de la tabla, el procedimiento sucede colocando la imagen y posteriormente los enlaces dentro de un contenedor div que lo solapa, los enlaces se obtienen en forma de arreglo por el controlador, lo que permite la impresión mediante un recorrido a través de un ciclo foreach(); esta sección de código puede verse en la figura 38.

```


<div style="position:relative; border:none; margin:0px; padding:0px;">
    <?php
        $enlRow = -1;
        foreach($objeto['Proximo'] as $enlProximo):
            $enlRow++;
        ?>
        <div class="enlaceProximo"<?php echo $enlRow; ?>" style="background: #3399FF; width:18px;
height:18px; border:1px solid #EEEEEE; z-index:3; position:absolute; color:#FFFFFF;
line-height:18px; text-align:center; left:<?php echo $enlProximo['ObjetosProximo']['coord_x'];
?>px; top:<?php echo $enlProximo['ObjetosProximo']['coord_y']; ?>px;" title="Hacia: <?php
echo $enlProximo['nombre']; ?>"><?php echo $enlRow; ?></div>
        <?php endforeach; ?>
    </div>

```

Figura 38. Impresión Mediante un Recorrido

Fuente: Los Autores

Ver detalle: Vista (view_panoramica) - controlador (viewPanoramica()) – modelo (Objeto). La descripción de esta funcionalidad es similar a la de listar panorámicas, siendo la diferencia más importante que sólo se limita a mostrar el detalle de una sola panorámica, por lo tanto el controlador utiliza la función find('first') en lugar de find('all').

Editar: Vista (editpano) - controlador (editepano()) – modelo (Objeto). La función editpano() del controlador utiliza la función find('first') con la condición de que el id sea igual al requerido, luego, la vista presenta el formulario lleno. El procedimiento para mostrar la panorámica junto con los enlaces sigue el mismo al efectuado en el punto previo, sin embargo, para estructurar el funcionamiento de esta vista, fue necesario realizar cierto procesamiento adicional con JavaScript, de manera que podamos gestionar la eliminación e inserción de nuevos enlaces y que estos se posicionen correctamente en la imagen. Las posiciones de los enlaces con respecto a la imagen son almacenadas en un arreglo de dos dimensiones (posición x, posición y), que es manipulado a medida que se editan dichos enlaces. Básicamente se utilizan seis funciones generales:

- ✓ **Initarray(i):** Inicializa el arreglo de posiciones de enlaces en la posición i.
- ✓ **SetValor(i,x,y):** Asigna valores a una dupla del arreglo (x,y).

- ✓ **Point_it(e,actual):** Se encarga de detectar las coordenadas (x,y) del punto en donde el usuario clicó, comprueba que las coordenadas no estén ocupadas por otro enlace, en caso afirmativo las dibuja en el contenedor sobre la imagen y guarda las posiciones en el arreglo. La figura 39 muestra el código de la función.

```
function point_it(e,actual){
    var pos_x = 0;
    var pos_y = 0;

    var offset = $('#pointer_div'+actual).offset();
    //alert(e.pageY+' '+offset.top);

    pos_x = Math.round(e.pageX-offset.left);
    pos_y = Math.round(e.pageY-offset.top);
    if (pos_x>825){ pos_x = 825 }
    if (pos_y>104){ pos_y = 104 }

    if (haySolapamiento(actual,pos_x,pos_y)){
        alert('ERROR: Esta posición se solapa con alguna otra');
    }
    else{
        /*- Se actualizan los valores que van a la BD -*/
        $('#Proximo'+actual+'ObjetosProximoCoordX').val(pos_x);
        $('#Proximo'+actual+'ObjetosProximoCoordY').val(pos_y);

        /*- Se actualiza el cuadro en la imagen -*/
        $('#enlaceProximo'+actual).css('left',pos_x);
        $('#enlaceProximo'+actual).css('top',pos_y);
        $('#enlaceProximo'+actual).css('visibility','visible');

        /*- Se actualiza el mensaje de alerta debajo de la imagen -*/
        $('#alerta'+actual).css('display','none');

        /*- Se agrega esta posición al arreglo que almacena las posiciones para comparar si una nueva
        posición choca con algún otro enlace -*/
        setValor(actual,pos_x,pos_y);
    }
}
```

Figura 39. Código de Detección de Coordenadas

Fuente: Los Autores

- ✓ **HaySolapamiento(act,x,y):** Esta función se encarga de verificar si la posición pasada por parámetro se solapa con alguna de las presentes en el arreglo de posiciones.
- ✓ **AddEnlace():** Se encarga de añadir filas a la tabla de enlaces de panorámica, esta función se ejecuta cada vez que se hace clic sobre el botón de añadir panorámica, lo cual agrega una nueva fila con una imagen sin elementos a la que el usuario debe clicar para agregar enlaces. Puede verse el código de esta función en la figura 40.

```

function addEnlace() {
    lastRow++;
    $("#tabla_enlace div:#enlace-1").clone(true).attr('id','enlace'+lastRow).removeAttr('style').insertBefore("#tabla_enlace div:#divEnlaceAdd");
    $("#enlace"+lastRow+" button").attr('onclick','removeEnlace('+lastRow+')');
    $("#enlace"+lastRow+" input:first").attr('name','data[Proximo]['+lastRow+'][ObjetosProximo][coord_x]').attr('id','Proximo'+lastRow+'ObjetosProximoCoordX');
    $("#enlace"+lastRow+" input:eq(1)").attr('name','data[Proximo]['+lastRow+'][ObjetosProximo][coord_y]').attr('id','Proximo'+lastRow+'ObjetosProximoCoordY');
    $("#enlace"+lastRow+" select:first").attr('name','data[Proximo]['+lastRow+'][ObjetosProximo][proximo_id]').attr('id','Proximo'+lastRow+'ObjetosProximoProximoId');
    $("#enlace"+lastRow+" div:eq(2)").attr('id','pointer_div'+lastRow).attr('onclick','point_it(event,'+lastRow+')');
    $("#enlace"+lastRow+" span:first").attr('id','alerta'+lastRow);
    $(".enlaceProximo-1").each(function(){ $(this).clone().attr('class','enlaceProximo'+lastRow).html(lastRow).insertAfter(this)});
    tam++;
    initArray(lastRow);
}

```

Figura 40. Función de Agregar Enlace

Fuente: Los Autores

- ✓ **RemoveEnlace(i):** Así como addEnlace() añade nuevas filas a la tabla de enlaces, esta función elimina filas existentes y actualiza los valores en el arreglo de posiciones. Puede verse el código de esta función en la figura 41.

```

function removeEnlace(x) {
    $("#enlace"+x).remove();
    $(".enlaceProximo"+x).remove();
    setValor(x,-100,-100);
}

```

Figura 41. Función de Remover Enlace

Fuente: Los Autores

Finalmente, el registro es almacenado en la base de datos utilizando la función saveAll(), aunque posteriormente se comprueban los enlaces que deberán eliminarse, la comparación se efectúa después del guardado para evitar problemas surgidos en el almacenamiento de los datos y un posible rollback. El proceso de comparación de los enlaces a eliminar, es similar al realizado en la comparación de campos e imágenes en la edición de elementos, para ello se utilizan dos arreglos, uno con la información que quiere ser almacenada y otro con la información original, se comparan y se determina si hay enlaces para eliminar.

Eliminar: Vista (No tiene) - controlador (deletepanoramica()) – modelo (Objeto). El controlador utiliza la función deletepanoramica() para eliminar el registro de la base de datos.

Agregar: Vista (add) - controlador (add()) – modelo (Objeto). La función agregar se estructura de la misma forma que la de editar, la única diferencia en el proceso es que no hay carga previa de imagen o enlaces, lo que permite que el procedimiento sea más sencillo. Aun así, las seis funciones utilizadas en la vista de editarpanorámica, son reutilizadas en la implementación de esta funcionalidad, ya que se necesita insertar nuevos enlaces para la nueva panorámica. El proceso de guardado es más simple, se utiliza `saveAll()` para guardar los datos, sin comparaciones adicionales de ningún tipo como en el caso de la edición.

6) Sobre la construcción de gestionar usuarios (Operaciones de listar, ver detalle, editar, eliminar y agregar):

En este particular destacan las modificaciones realizadas al controlador general de la aplicación (AppController.php), en donde se implementaron algunas funciones de control sobre la autenticación. La implementación de esta funcionalidad implicó el uso del componente Auth, al que le fueron configuradas algunas variables en la función `beforeFilter()`. Se puede apreciar mayor detalle de este código en la figura 42.

```
public $components = array(
    'Session',
    'Auth' => array(
        'loginRedirect' => array('controller'=>'users','actions'=>'index'),
        'logoutRedirect' => array('controller'=>'users','actions'=>'index'),
        'authError' => 'Por favor, inicie sesión para acceder a esta página.',
        'authorize' => array('Controller')
    )
);

public function isAuthorized($usuario){
    return true;
}

public function beforeFilter(){
    //$this->Auth->allow('index','view');
    $this->set('logged_in',$this->Auth->loggedIn());
    $this->set('current_user',$this->Auth->user());
}
```

Figura 42. Funciones de Autenticación

Fuente: Los Autores

Además de las modificaciones al AppController.php, también se procedió a crear un controlador propio de los usuarios (UsersController.php), que almacena las funciones de gestión básicas,

pero que también contiene algunas funciones adicionales necesarias para el control de acceso, estas son:

- ✓ **Login:** Una vez se chequea si la petición es de tipo POST, permite el acceso del usuario utilizando la función `Auth->login()`.
- ✓ **Logout:** Cierra la sesión del usuario utilizando la función `Auth->logout()`.
- ✓ **IsAuthorized:** Verifica si el usuario está autorizado para acceder a determinadas vistas.

Ahora bien, en cuanto a las funciones básicas de gestión tenemos:

Listar: Vista (`index`) - controlador (`index()`) – modelo (`User`). Funciona de la misma forma que la mayoría de los listar anteriores. La función `index()` del controlador utiliza la función `find('all')`, cuyos datos son pasados a la vista, quien finalmente construye una tabla con la lista de usuarios. En esta función se aplica cierto control sobre las opciones visibles para cada usuario, si el usuario es administrador se habilitan todas las funciones de gestión por cada usuario, en caso contrario, sólo se habilita la edición del perfil propio.

Editar: Vista (`edit`) - controlador (`edit()`) – modelo (`User`). La función editar usuarios sigue un patrón similar al de otras funciones editar vistas previamente, con la diferencia fundamental de que adicionalmente se valida que el tipo de petición sea del tipo POST. Por demás el controlador utiliza el mismo procedimiento, verifica si el usuario solicitado existe y los datos son guardados utilizando la función `save()`.

Eliminar: Vista (No tiene) - controlador (`delete()`) – modelo (`User`). En la función eliminar se realizó una validación adicional permitiendo que no se acepten solicitudes tipo GET. El controlador utiliza la función `delete()` para eliminar el registro de la base de datos. Este código puede observarse en detalle a través de la figura 43.

```
public function delete($id = null){
    if ($this->request->is('get')){
        throw new MethodNotAllowedException();
    }
    if(!$id){
        $this->Session->setFlash('Usuario invalido','flash_error');
        $this->redirect(array('action' => 'index'));
    }
    if($this->User->delete($id){
        $this->Session->setFlash('Usuario eliminado exitosamente', 'flash_success');
        $this->redirect(array('action' => 'index'));
    }
    $this->Session->setFlash('Usuario no eliminado','flash_error');
    $this->redirect(array('action' => 'index'));
}
```

Figura 43. Funciones de Eliminar

Fuente: Los Autores

Agregar: Vista (add) - controlador (add()) – modelo (User). La función agregar es también bastante básica, se valida que la petición sea tipo POST y posteriormente el guardado sucede mediante la función save().

3.8 Interfaz del portal web

En esta sección se muestran dos tipos de interfaz la del usuario con perfil de visitante general y la del módulo que permite a usuarios con perfil administrativo o registrado tener acceso a las funcionalidades de gestión del contenido del portal.

3.8.1 Interfaz del usuario con perfil de visitante

A continuación se colocarán las interfaces del portal, con las cuales los visitantes interactúan. La figura 44 muestra la página de inicio del arte en la CUC. En esta página da la bienvenida al sitio dando una breve reseña.



Figura 44. Interfaz de Página de Inicio

Fuente: Los Autores

Las principales secciones que ofrece la página de inicio de los visitantes podemos mencionar:

- ✓ ¿Quiénes somos?: muestra información
- ✓ ¿Por qué somos patrimonio?: muestra información de la declaración de patrimonio de la CUC y por cuales criterios.
- ✓ Historia: muestra información de las etapas y cuál fue el desarrollo a lo largo del tiempo.

- ✓ Conjunto Arquitectónico: muestra información e imágenes de los diferentes campos, ejemplos: paisajismo, jardín botánico, entre otros.
- ✓ Paseo Virtual: muestra mediante vistas panorámicas los lugares de interés en nuestro caso la Plaza Cubierta del Rectorado, posibilitando la exploración de las obras de arte deseadas.
- ✓ Galería: muestra información e imágenes clasificadas por categoría y ordenadas por nombre autor, fecha, entre otros.
- ✓ Documentos: muestra documentos de interés vinculados con la Ciudad Universitaria de Caracas.
- ✓ Logos: se encuentran logos de instituciones involucradas en el desarrollo. Al seleccionar estos logos se pueden acceder a la página de cada una de las instituciones o dependencias de la UCV.
- ✓ Pie de página: se encuentra enlaces a información importante y funcional.

En la figura 45 se muestra la interfaz de ¿Por qué somos patrimonio? Al seleccionar la imagen se sobrepone a la página, también los criterios son enlaces a imágenes que representa esos criterios de aceptación de patrimonio.



Figura 45. Interfaz de ¿Por Qué Somos Patrimonio?

Fuente: Los Autores

En la figura 46 se muestra la interfaz de Conjunto. Esta página contiene imágenes las cuales al ser seleccionada se despliegue la pagina y muestra la información al final. En el mapa de Google Map se actualizar con información de ubicación.

UNIVERSIDAD CENTRAL DE VENEZUELA
EL ARTE EN LA CIUDAD UNIVERSITARIA DE CARACAS

INICIO ¿POR QUÉ SOMOS PATRIMONIO? HISTORIA **CONJUNTO ARQUITECTÓNICO** PASEO VIRTUAL GALERÍA RECORRIDOS DOCUMENTOS

MAPA DE LA CUC

CONJUNTO ARQUITECTÓNICO

BUSCAR

OBRAS DE ARTE EDIFICACIONES URBANISMO PAISAJISMO JARDÍN BOTÁNICO OTRAS COLECCIONES

El Campus Ciudad Universitaria de Caracas fue construido entre los primeros años de la década de los 40 hasta entrados los 60, bajo la notoria influencia del movimiento moderno en la arquitectura y el arte. Este discurso de la modernidad se pone de manifiesto en el diseño del Maestro Carlos Raúl Villanueva por medio de dos postulados clave: la noción de integración y el principio de la nueva monumentalidad. Principios que quedan claramente ilustrados a través de la propuesta del arquitecto Villanueva al invitar a un grupo de veinticinco artistas de vanguardia de distintas nacionalidades para establecer por medio del lenguaje plástico un diálogo entre el arte, el espacio, los elementos arquitectónicos, el urbanismo, el paisajismo y las condiciones naturales del ambiente. El ideal de incorporación de un todo para poner al servicio del espectador común los valores del espíritu a través del arte.

25 Artistas invitados: Jean Arp, Miguel Arroyo, Armando Barrios, André Bloc, Carlos González Bogen, Alexander Calder, Omar Carreño, Pedro León Castro, Baltasar Lobo, Wifredo Lam, Henry Laurens, Fernand Leger, Mateo Manaure, Pascual Navarro, Francisco Narváez, Alirio Oramas, Alejandro Otero, Antoine Pevsner, Héctor Poleo, Braulio Salazar, Jesús Soto, Sophie Tauber-Arp, Víctor Vasarely, Víctor Valera y Oswaldo Vigas

FICHA DE LA OBRA

SIN TÍTULO (BIMURAL)

Autor: [\[+\] Fernand Léger](#)

Año de creación: 1954

Ubicación: Plaza Cubierta del Rectorado

Tipo: Mural

Material: Concreto / Mosaico artesanal y vitreos indust

Dimensiones: 303.5 x 1239.5 x 21.8 cm

[\[+\] Versión para imprimir](#)

Lectura: Detrás del Amphion está ubicado un bimural del artista francés Fernand Léger. Esta obra se encuentra soportada en una estructura curva, elevado del piso por cuatro elementos de concreto.

[FUNDACIÓN VILLANUEVA] | [CENTENARIO VILLANUEVA] | [CONTACTENOS] | [CRÉDITOS]

© 2013 UNIVERSIDAD CENTRAL DE VENEZUELA. DERECHOS RESERVADOS

Figura 46. Interfaz de Conjunto

Fuente: Los Autores

En la figura 47 se muestra la interfaz de Paseo virtual. Esta página tiene una fecha de adelante y atrás en ellas se muestra los puntos que hacer referencia a obras dentro de ese paseo. En el mapa de Google Map se actualizar con información de ubicación.



Figura 47. Interfaz de Paseo Virtual

Fuente: Los Autores

En la figura 48 se muestra la interfaz de Galería. Al seleccionar la imagen se sobrepone a la página.

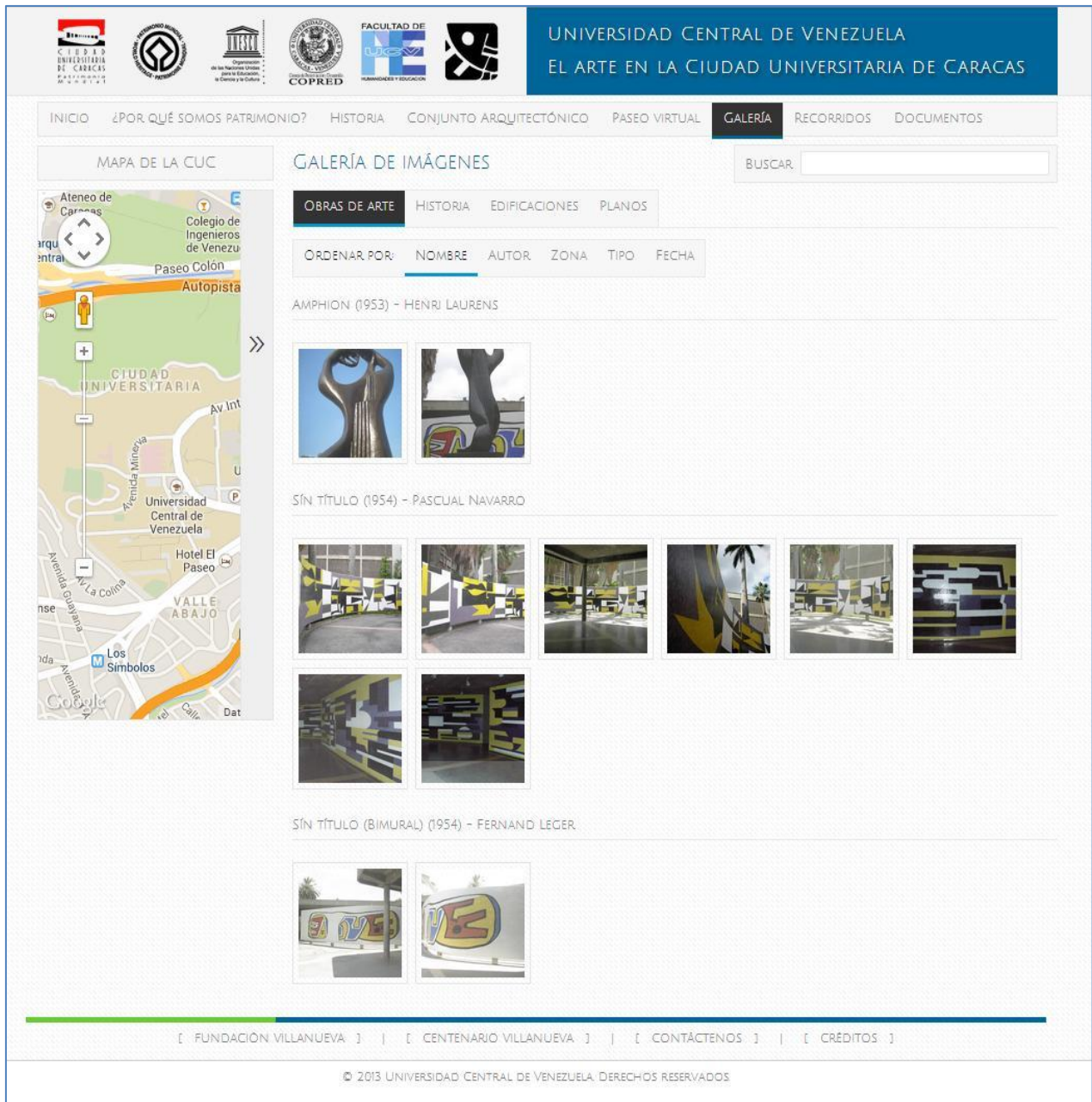


Figura 48. Interfaz de Galería

Fuente: Los Autores

3.8.2 Interfaz para el módulo de Administración o Gestión de contenido del usuario con perfil administrador o registrado

En la figura 49 se presenta la interfaz para la administración del contenido la primera pagina que muestra esta interfaz es la de autenticación de usuario.

UNIVERSIDAD CENTRAL DE VENEZUELA
EL ARTE EN LA CIUDAD UNIVERSITARIA DE CARACAS

PANEL ADMINISTRATIVO PARA LA GESTIÓN DEL CONTENIDO

INICIO AUTORES ZONAS PASEO VIRTUAL ELEMENTOS INICIAR SESIÓN

ACCESO A USUARIOS

Nombre de usuario

Contraseña

Acceder

Figura 49. Interfaz de Administración

Fuente: Los Autores

Las principales secciones que ofrece la página de inicio para la administración podemos mencionar:

- ✓ Inicio sesión: formulario para autenticación de usuario.
- ✓ Autores: se crea y gestiona los usuarios.
- ✓ Zonas: se crea y gestiona las zonas.
- ✓ Panorámicas: se crea y gestiona las panorámicas.
- ✓ Elementos: se crea y gestiona los elementos y los tipos.
- ✓ Cerrar sesión: nos permite desconectarnos de la interfaz de administración.

En la figura 50 se presenta la interfaz para listar autores aquí se gestiona a los autores después que son creados.



UNIVERSIDAD CENTRAL DE VENEZUELA
EL ARTE EN LA CIUDAD UNIVERSITARIA DE CARACAS

PANEL ADMINISTRATIVO PARA LA GESTIÓN DEL CONTENIDO

INICIO AUTORES ZONAS PASEO VIRTUAL ELEMENTOS CERRAR SESIÓN

LISTAR AUTORES

NOMBRE	OBRAS	LUGAR Y FECHA DE NACIMIENTO	ACCIONES
Henri Laurens	Amphion (1953)	Paris, Francia. 18/02/1885	  
Fernand Leger	Sin título (Bimural) (1954)	Cdad. del Cabo. 04/02/1881	  
Pascual Navarro	Sin título (1954)	Caracas, Venezuela. 17/05/1923	  
	Sin título (1954)		

Figura 50. Interfaz de Listar Autores

Fuente: Los Autores

3.9 Pruebas y resultados

Para realizar las pruebas, se hizo una adaptación de las Pruebas de Usabilidad, donde se seleccionaron preguntas al público en general. Como instrumento de recolección de datos, se utilizó la encuesta, con el propósito de conocer opiniones de los usuarios con respecto al sitio Web desarrollado (ver Anexo B).

La muestra fue restringida a personas del tipo administrador como son la Profa. Mariantonia Palacios y la Profa. Yusneyi Carballo y a cinco usuarios visitantes desde inexpertos a expertos en el área de computación (Solo respondieron tres usuarios).

Las pruebas se aplicaron de la siguiente manera:

1. Se le envió por Internet la encuesta con el link de la página para que realice el recorrido por todas las secciones del sitio.
2. Una vez que el encuestado visualizó el recorrido del sitio, aplicaría la encuesta en forma digital para obtener su opinión y sugerencias.

En base a las respuestas obtenidas, fue posible evidenciar que el portal fue aceptado positivamente por los encuestados, sin embargo no podemos dar un indicio que puede ser aceptado por el público en general.

Por otra parte, se logró cumplir el objetivo principal de este trabajo que es el de construir un portal que permita divulgar información y realizar un paseo virtual por la Plaza Cubierta del Rectorado presentando las obras más emblemáticas que se encuentran allí, el sitio creado tuvo buena aceptación por parte de las personas encuestadas.

Como resultado final de este Trabajo Especial de Grado se obtuvo una aplicación que permite:

1. Visualizar las galerías de imágenes.
2. Realizar búsqueda sencilla en la base de datos de la página.
3. Ver información de los diversos autores y sus obras.
4. Realizar paseos por la Plaza Cubierta del Rectorado.
5. Ofrecer un mapa del sitio.
6. Ofrecer un espacio informativo para: quiénes somos, por qué somos patrimonio, historia.
7. Al administrador, actualizar el contenido de la aplicación.

Durante el proceso de desarrollo fueron de gran importancia los siguientes aspectos:

- ✓ El uso de FDD como metodología ágil, permitió organizar, dividir y estructurar las tareas de desarrollo.

- ✓ La aplicación de modelos de diseño Web, estándares, patrones y lineamientos de usabilidad, porque con ello se logró un diseño claro, limpio y acorde en el desarrollo del módulo.
- ✓ El diseño de la base de datos adaptado a los lineamientos del *framework CakePHP*, ya que, permitió usar al máximo las ventajas del *framework*.

CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Después de haber aplicado la metodología FDD al TEG, se pudo constatar que la misma facilitó la etapa de diseño y desarrollo. La entrevista con el cliente y el análisis de requerimientos, permitieron conocer los requisitos funcionales, precisar un modelo de estudio y establecer los módulos necesarios para el desarrollo del sistema con la creación de la lista de funcionalidades; esto gracias a que FDD le presta más importancia a la comunicación y participación activa con el usuario que a la documentación excesiva.

El resultado de una buena planificación se ve reflejado en el tiempo de implementación de cada funcionalidad, ninguna con duración mayor a dos semanas, ya que la metodología establece ese límite de tiempo, que de no ser cumplido, demanda la división de la funcionalidad en fragmentos menos generales y más pequeños.

Con el diseño de la base de datos basada en el modelo relacional, así como también el diseño de la interfaz de usuario y el módulo administrador como parte fundamental de la solución, se pudieron cumplir satisfactoriamente los requisitos del sistema. Las interfaces cumplen un papel importante a la hora de elaborar un sistema, porque a través de ellas los usuarios pueden interactuar con él, obteniendo mayor provecho al realizar las operaciones en un menor tiempo.

Se desarrolló el portal web empleando herramientas de software libre, como son el lenguaje de programación PHP y MySQL como manejador de base de datos, además de varias librerías de JQuery que ayudaron en el procesamiento de la data, así como también en la mejora de la presentación; luego se realizaron las pruebas de aceptación, verificando que el software cumple con los requerimientos mínimos, siguiendo las directrices de las buenas prácticas en ingeniería de software y programación. Este desarrollo se llevó a cabo utilizando el patrón MVC, que permitió

construir un esquema escalable y modular, cuyas partes se pueden mejorar y extender hacia nuevas funcionalidades de forma fácil y coordinada.

4.2 Recomendaciones y Trabajos Futuros

Como recomendaciones para futuros trabajos de investigación que bien podrían ser tomadas en cuenta con el fin de optimizar y extender las funcionalidades del portal desarrollado, resulta necesario realizar la integración de nuevos componentes o herramientas funcionales teniendo en cuenta las posibles innovaciones que puedan ser incluidas en el portal para dar continuidad al desarrollo de este proyecto; son recomendaciones que están fuera del alcance de este Trabajo Especial de Grado, pero que agregarían funcionalidades y un mayor alcance al portal son:

- ✓ Implementar la funcionalidad de permitir el recorrido virtual, para crear recorridos por los diferentes lugares de la Ciudad Universitaria de Caracas.
- ✓ Implementar la funcionalidad de registrar usuarios en el portal para visitantes, de manera que puedan guardar y generar recorridos por las diversas obras y lugares.
- ✓ Mejorar la carga de la galería, almacenando la información de las imágenes en un arreglo, de manera que el ordenamiento se haga sobre el arreglo y no a través de la consulta a la base de datos.
- ✓ Implementar una vista en pantalla completa del paseo virtual.
- ✓ Implementar la funcionalidad de gestión de otras secciones de la galería no alimentadas por las imágenes de los elementos, tal es el caso de las secciones historia y planos.

BIBLIOGRAFÍA

1P-Estilosdevida. (2009). Recuperado el 18 de abril de 2012, de <http://www.1p-estilosdevida.com/museologia-arte-estilo-y-patrimonio-cultural/>

Alegsa. (1998). Recuperado el 20 de abril de 2012, de <http://www.alegsa.com.ar/Dic/photoshop.php>

Art, N. G. (2013). *National Gallery of Art*. Retrieved from National Gallery of Art: www.nga.gov/ngaweb/Collection

Art, T. M. (2013). *The Metropolitan Museum of Art*. Retrieved from The Metropolitan Museum of Art: www.metmuseum.org/collections

Basilica de San Pablo. (s.f.). Recuperado el 23 de mayo de 2012, de http://www.vatican.va/various/basiliche/san_paolo/vr_tour/index-it.html

Basilica de San Pablo. (s.f.). Recuperado el 23 de mayo de 2012, de http://www.vatican.va/various/basiliche/san_paolo/index_sp.html

Cavsi. (s.f.). Recuperado el 12 de mayo de 2012, de <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>

COPRED. (2002-2003). *Exposicion itinerante Ciudad Universitaria de Caracas, Patrimonio Mundial*. Caracas: Cuadernos del Patrimonio Mundial.

Desarrolloweb.com. (14 de octubre de 2009). Recuperado el 01 de mayo de 2012, de <http://www.desarrolloweb.com/articulos/que-es-html5.html>

Desarrolloweb.com. (s.f.). Recuperado el 16 de abril de 2012, de <http://www.desarrolloweb.com/articulos/que-es-html5.html>

Developers.google.com. (19 de mayo de 2010). Recuperado el 16 de abril de 2012, de documentacion: <http://developers.google.com/maps/documentation/javascript/v2/basics?hl=es>

Easypano.com. (2001). Recuperado el 16 de abril de 2012, de <http://www.easypano.com/es/productsTw.htm>

EcuRed. (20 de enero de 2012). Recuperado el 20 de mayo de 2012, de http://www.ecured.cu/index.php/Metodolog%C3%ADa_FDD

Ecured.cu. (s.f.). *EcuRed*. Recuperado el 08 de abril de 2012, de Visita virtual: http://www.ecured.cu/index.php/Visita_virtual

Ecured.cu. (s.f). Recuperado el 23 de mayo de 2012, de http://www.ecured.cu/index.php/Patrones_de_dise%C3%B1o_y_arquitectura

Fancybox. (11 de noviembre de 2010). Recuperado el 20 de abril de 2012, de <http://fancyapps.com/fancybox/>

Formacion en Red. (2008). Recuperado el 16 de abril de 2012, de <http://www.ite.educacion.es/formacion/materiales/107/cd/audio/audio0102.html>

Formacion en Red. (2008). Recuperado el 16 de abril de 2012, de <http://www.ite.educacion.es/formacion/materiales/107/cd/video/video0102.html>

Formación en Red. (2008). Recuperado el 16 de abril de 2012, de <http://www.ite.educacion.es/formacion/materiales/107/cd/imagen/imagen0105.html>

Fotografia360. (s.f). Recuperado el 16 de abril de 2012, de <http://www.fotografia360.co/>

Gimp.es. (s.f). Recuperado el 20 de abril de 2012, de <http://gimp.es/>

Gimp.es. (s.f.). Recuperado el 20 de abril de 2012, de <http://gimp.es/>

Gutiérrez, J., & Chacón, B. (2012). *Análisis de requerimientos para la creación de una aplicación web que permita realizar un paseo virtual por el campus de la Ciudad Universitaria de Caracas. Caso de Estudio: Plaza Cubierta del Rectorado*.

Herman, S. M. (2004). *Diseño de una Metodología Ágil de Desarrollo de Software*. Retrieved from [Diseño de una Metodología Ágil de Desarrollo de Software: materias.fi.uba.ar/schenone-tesisdegradoingenieriainformatica.pdf](http://Diseño%20de%20una%20Metodolog%C3%ADa%20%C3%81gil%20de%20Desarrollo%20de%20Software:materias.fi.uba.ar/schenone-tesisdegradoingenieriainformatica.pdf)

Hucreative.com. (s.f). Recuperado el 12 de mayo de 2012, de <http://www.hucreative.com/google/go1.html>

Hugin. (10 de diciembre de 2008). Recuperado el 20 de abril de 2012, de http://citywiki.ugr.es/wiki/Hugin._Tutorial_de_uso

Inah. (s.f). Recuperado el 23 de mayo de 2012, de <http://www.inah.gob.mx/paseos/Calakmul/tour.html>

Inah. (s.f.). Recuperado el 23 de mayo de 2012, de <http://www.inah.gob.mx/paseos/Calakmul/tour.html>

Inah.gob.mx. (2010). Recuperado el 26 de mayo de 2012, de <http://www.inah.gob.mx/index.php/iquienes-somos>

(2009). *Inteco*. España: Instituto Nacional de Tecnologías de la Comunicación.

Juaa Maria Fernanda, M. A. (2000). *La Cuidad Universitaria de Caracas*. Caracas: Instituto del Patrimonio de la Cultura.

Libroweb. (2009). Recuperado el 16 de abril de 2012, de <http://www.librosweb.es/ajax/capitulo1.html>

Libroweb.com. (2009). Recuperado el 16 de abril de 2012, de http://www.librosweb.es/css/pdf/introduccion_css.pdf

Libroweb.es. (s.f). Recuperado el 01 de mayo de 2012, de <http://www.librosweb.es/javascript/capitulo1.html>

linperial.com. (2006). Recuperado el 12 de mayo de 2012, de <http://www.linperial.com/communities/forums/administrators/?q=node/16>

Maps.google.com. (s.f). Recuperado el 12 de abril de 2012, de <http://maps.google.com/help/maps/streetview/learn/using-street-view.html>

Maps.google.com. (s.f). Recuperado el 12 de abril de 2012, de <http://maps.google.com/help/maps/streetview/learn/where-is-street-view.html>

Marina, G. (1991). *Obras de arte de la Ciudad Universitaria de Caracas*. Caracas: Ediciones Universidad Central de Venezuela.

Museo Nacional Centro de Arte Reina Sofia. (2013). Retrieved from Museo Nacional Centro de Arte Reina Sofia: www.museoreinasofia.es/coleccion/obra/sonia-klamery-echada

Museo Thyssen. (2009). Recuperado el 24 de mayo de 2012, de <http://www.museothyssen.org/>

Museo Virtual. (2010). Recuperado el 23 de mayo de 2012, de <http://www.museovirtualdeamericalatinayelcaribe.org/>

Museos Virtuales. (4 de febrero de 2009). Recuperado el 8 de abril de 2012, de ¿Qué es un museo virtual?: <http://museosvirtuales.wordpress.com/2009/02/04/museovirtual/>

Museum Box. (s.f). Recuperado el 17 de abril de 2012, de <http://museumbox.e2bn.org/creator/>

Orienta.org. (s.f). Recuperado el 16 de abril de 2012, de <http://www.orienta.org.mx/biblioteca/catalogacion.html>

Prado, M. N. (2013). *Museo Nacional del Prado*. Retrieved from Museo Nacional del Prado: www.museodelprado.es/coleccion/dibujos-y-estampas/

Project, G. A. (2011, Febrero 11). *Google Art Project*. Retrieved from Google Art Project: www.google.com/culturalinstitute/collections?projectId=art-project&hi=es

Puertas, J. P. (2007). PHP. En J. P. Puertas, *Creacion de un portal con PHP y MySQL* (págs. 1-2). Mexico: 2007.

Rodriguez, J. L. (12 de marzo de 2007). *dzoom*. Recuperado el 20 de abril de 2012, de <http://www.dzoom.org.es/noticia-1606.html>

TechNoticias. (1 de marzo de 2011). Recuperado el 28 de abril de 2012, de <http://technoticias1.wordpress.com/2011/03/01/algunas-herramientas-para-el-diseno-grafico/>

Tecnologias de Realidad Virtual. (2001). Recuperado el 12 de mayo de 2012, de <http://telematica.cicese.mx/computo/super/cicese2000/realvirtual/Part2.html>

Telos. (s.f). Recuperado el 24 de abril de 2012, de <http://sociedadinformacion.fundacion.telefonica.com/telos/noticiasgrabar.asp?idarticulo=10&ev=51.htm>

Teoman. (13 de septiembre de 2011). *tecnologiaid.com*. Recuperado el 17 de abril de 2012, de <http://www.tecnologiaid.com/museumbox-crear-un-museo-online-para-hacer-exposiciones-virtuales.html>

Torregrosa et al. (s.f). Recuperado el 12 de mayo de 2012, de <http://es.scribd.com/doc/84325178/Metodologias-Agiles>

Uamisoft.izt.uam.mx. (s.f.). Recuperado el 12 de mayo de 2012, de http://uamisoft.izt.uam.mx/uamisoft/lib/exe/fetch.php?media=amm:cursos:psu:07-o:procesos_agiles.pdf

UCV, D. d. (2003). *El Aula Magna y la sinstesis de las artes 50 años*. Caracas: Editorial Universidad Central de Venezuela.

Valdes, A. G. (1974). *Obras de Arte de la Ciudad Universitaria de Caracas*. Caracas: 1974.

Valverde, A. &. (2007). Recuperado el 12 de mayo de 2012, de <http://www.seccperu.org/files/Metodologias%20Agiles.pdf>

Visual Beta. (13 de marzo de 2008). Recuperado el 20 de abril de 2012, de <http://www.visualbeta.es/1816/aplicaciones-web/google-incluye-google-maps-street-view-en-la-api-de-mapas/>

XcitingArt.com. (18 de abril de 2008). Recuperado el 20 de abril de 2012, de <http://blog.xcitingart.com/arte-y-diseno/jquery-virtual-tour/>

ANEXOS

Anexo A: Casos de Usos

Mediante los diagramas de casos de uso se desea esbozar la aplicación a desarrollar, con las funcionalidades y las interacciones que esta va a presentar. En este sentido, son identificados dos actores principales:

- ✓ Visitante: Son aquellos usuarios que acceden al sitio con el objetivo de recorrer los espacios ofrecidos.
- ✓ Administrador: Son aquellos usuarios que se encargan de evaluar y gestionar el contenido del sitio.

Lo dicho anteriormente confluye en la siguiente tabla, que agrupa un breve resumen de los casos de uso planteados:

Código	Caso de uso	Actor
1.1	Describe el ingreso del usuario a la sección de "¿Quiénes somos?"	Visitante
1.2	Describe el ingreso del usuario a la sección de "¿Por qué somos patrimonio?"	Visitante
1.3	Abarca el ingreso del usuario a la sección de "Historia"	Visitante
1.4	Permite el ingreso del usuario a la sección de Conjunto"	Visitante
1.5	Describe el ingreso del usuario a la sección de "Paseo Virtual"	Visitante
1.6	Describe el ingreso del usuario a la sección de "Galería"	Visitante
1.7	Describe el ingreso del usuario a la sección de "Documento"	Visitante
2.1	Permite la inserción de obras dentro de la aplicación	Administrador
2.2	Cubre la gestión de obras previamente insertadas en la aplicación	Administrador
2.3	Permite la inserción de la panorámicas dentro de la aplicación	Administrador
2.4	Describe la gestión de la panorámicas previamente insertados en la aplicación	Administrador
2.5	Permite la inserción de zonas y tipos dentro de la aplicación	Administrador
2.6	Cubre la gestión de zonas y tipos previamente insertadas en la aplicación	Administrador
2.7	Permite la inserción de los autores de las obras dentro de la aplicación	Administrador
2.8	Abarca la gestión de los autores previamente insertados en la aplicación	Administrador
2.9	Permite la inserción de los elementos de las obras dentro de la aplicación	Administrador
2.10	Abarca la gestión de los elementos previamente insertados en la aplicación	Administrador

Tabla 8. Casos de usos: Resumen y actores

FUENTE: Los Autores

Se construyeron tres(3) diagramas de casos de uso, los cuales se muestran a continuación:

✓ Diagrama de caso de uso para el usuario con perfil visitante:



Figure 51. Diagrama de caso de uso para el usuario con perfil visitante

Fuente: Los Autores

✓ Diagrama de caso de uso para el usuario con perfil administrador:

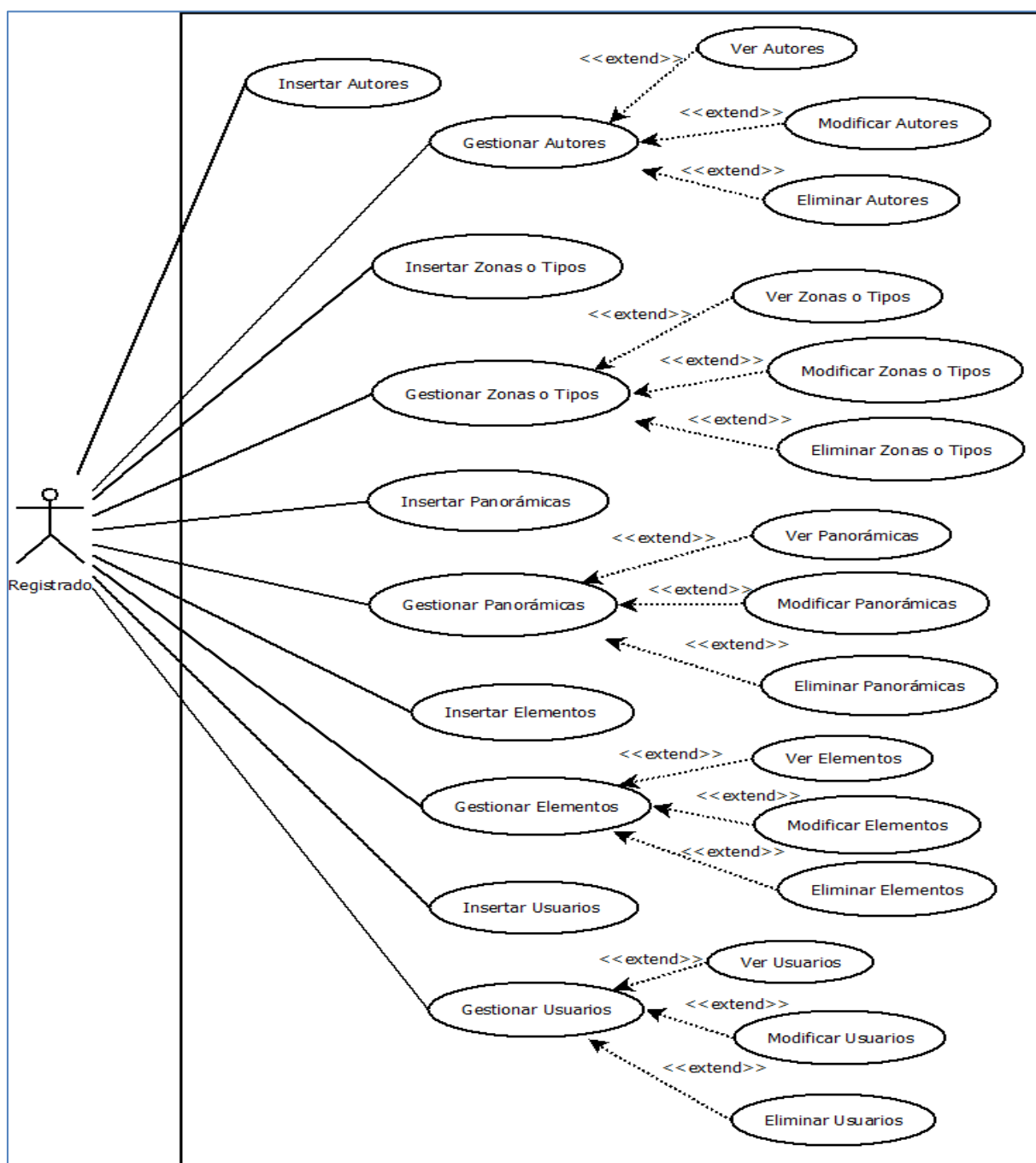


Figura 52. Diagrama de caso de uso para el usuario con perfil administrador

Fuente: Los Autores

✓ Diagrama de caso de uso para el usuario con perfil registrado:

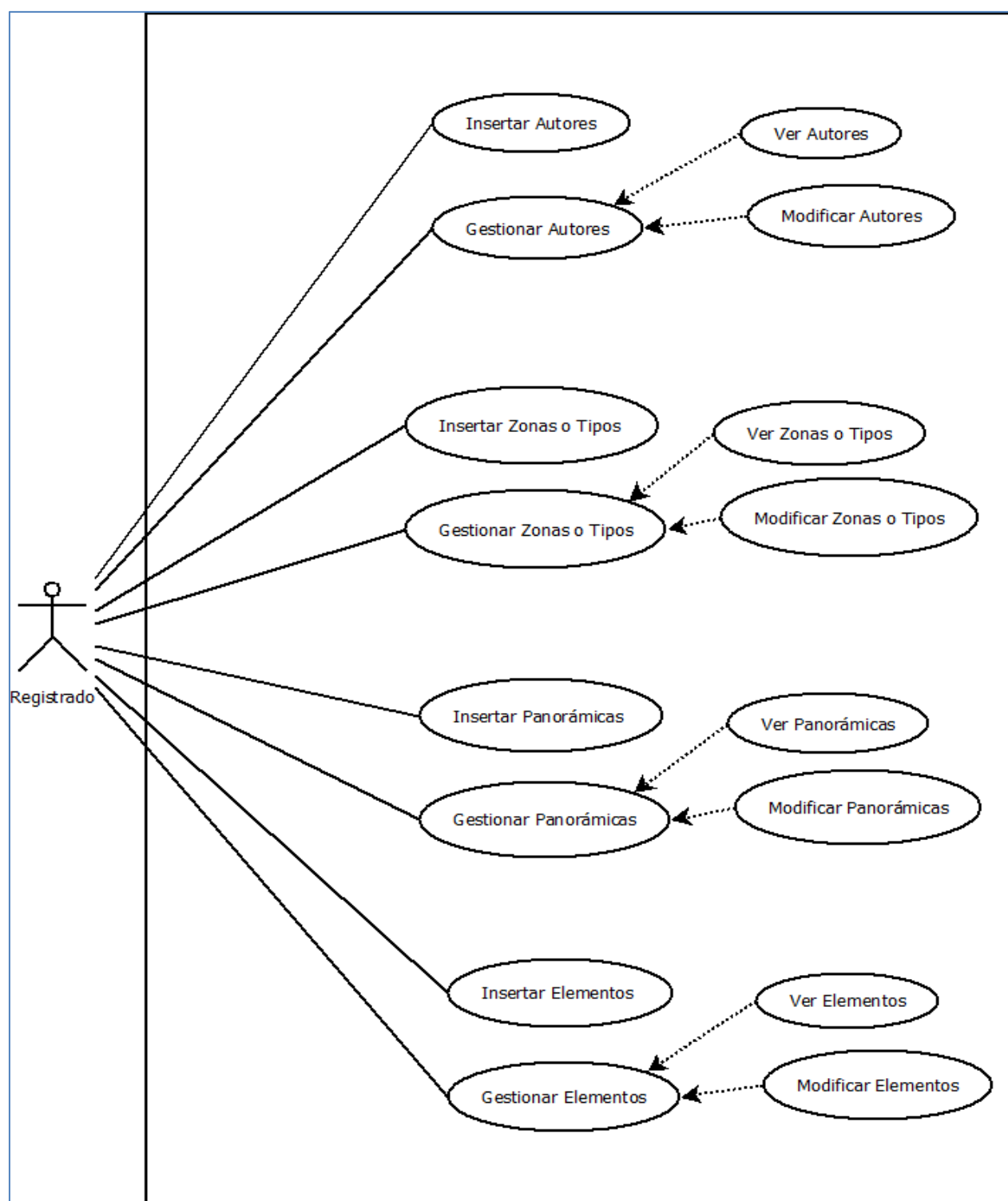


Figura 53. Diagrama de caso de uso para el usuario con perfil registrado

Fuente: Los Autores

Anexo B: Test de aceptación

Test de aceptación para los usuarios con perfil de visitante

1.- ¿Con la información que se ofrece en pantalla, es posible saber a qué institución o empresa corresponde el sitio?

- ☐ Si
- ☐ No tengo forma de saberlo

1.1.- ¿Cómo lo sabe?

2.- De los elementos que muestra esta pantalla, ¿hay algo que usted crea que está fuera de lugar?

- ☐ Si
- ☐ No

2.1.- En caso afirmativo. Lo puede indicar

3.- Tras una primera mirada, ¿le queda claro cuál es el objetivo del sitio y el contenido que ofrece?

- ☐ Si
- ☐ No

4.- ¿Los textos usados en los enlaces identifican correctamente el contenido de las páginas a las que se accede a través de ellos?

- ☐ Si
- ☐ No

5.- En caso de haber información relacionada con la que estaba viendo, ¿se le ofreció de manera simple?
¿O tuvo que volver a navegar para encontrarla?

- ☐ Esta de manera simple
- ☐ Tuve que navegar para encontrarla

6.- ¿Existen elementos dentro de las páginas, que le permitan saber exactamente en dónde se encuentra dentro de este sitio y cómo volver atrás sin usar los botones del programa navegador?

- ☐ Si
- ☐ No
- ☐ Otro:

7.- ¿La información que se le ofrece en pantalla le parece adecuada para entender dónde está ubicado en cualquier momento?

- ☐ Si
☐ No

8.- ¿Se ha perdido dentro del sitio?

- ☐ Si
☐ No

8.1.- ¿Recuerda en qué área fue?

9.- ¿Le pareció adecuada la forma en que se muestran las imágenes en el sitio web?

- ☐ Son nítidas y adecuadas para representar el contenido
☐ Son nítidas pero no adecuadas para representar el contenido
☐ No son nítidas pero si adecuadas para representar el contenido
☐ No son nítidas ni adecuadas para representar el contenido

10.- ¿Las imágenes grandes se demoraron más de lo esperado?

- ☐ Si
☐ No

11.- ¿Tuvo que seguir navegando sin que llegaran a mostrarse completamente las imágenes?

- ☐ Si
☐ No

12.- ¿Considera que gráficamente el sitio está equilibrado, muy simple o recargado?

- ☐ Muy Simple
☐ Muy Recargado
☐ Equilibrado

13.- Observación general o sugerencia

Test de aceptación para los usuarios con perfil de administrador

1.- ¿El sistema le permite agregar los siguientes elementos?

- ☐ Autores
- ☐ Zonas
- ☐ Panorámicas
- ☐ Enlaces
- ☐ Tipos

1.1.- En caso en que falle alguna de las anteriores, especifique el problema:

2.- ¿El sistema le permite editar los siguientes elementos?

- ☐ Autores
- ☐ Zonas
- ☐ Panorámicas
- ☐ Enlaces
- ☐ Tipos

2.1.- En caso en que falle alguna de las anteriores, especifique el problema:

3.- ¿El sistema le permite eliminar los siguientes elementos?

- ☐ Autores
- ☐ Zonas
- ☐ Panorámicas
- ☐ Enlaces
- ☐ Tipos

3.1.- En caso en que falle alguna de las anteriores, especifique el problema:

4.- ¿El sistema le permite listar los siguientes elementos?

- ☐ Autores
- ☐ Zonas
- ☐ Panorámicas
- ☐ Enlaces

☐ Tipos

4.1.- En caso en que falle alguna de las anteriores, especifique el problema:

5.- El módulo de administración es intuitivo y de fácil uso

☐ Si

☐ No

6.- ¿Al mandar datos mediante un formulario, el web le avisa si los recibió correctamente?

☐ Si

☐ No

7.- Los campos en los formularios contienen ayudas, ejemplos o modelos de respuestas para demostrar el dato que se debe introducir.

☐ Si

☐ No

8.- Existe una clara distinción entre campos “requeridos” y “opcionales” en los formularios.

☐ Si

☐ No

9.- Las cajas de texto indican la cantidad y el formato de los datos que deben ser introducidos.

☐ Si

☐ No

10.- Observación general o sugerencia respecto al módulo de administración

Anexo C: Manual de usuario

Manual de Usuario del Panel Administrativo para la Gestión del Contenido

Desarrollado por:
Br. Belkis Z. Chacón Pacheco
Br. Jorge E. Gutiérrez Roa