



Universitat
Autònoma
de Barcelona



2814-2

GENERACIÓN DE VISITAS VIRTUALES A MUSEOS 3D PARA LA INCLUSIÓN EN PÁGINAS WEB

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Marta Laperal Martín
i dirigit per
Daniel Riera Terrén
Bellaterra, 12 de Septiembre de 2011

El sotasingnat, **Daniel Riera Terrén**

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per na **Marta Laperal Martín**.

I per tal que consti firma la present.



Signat: Daniel Riera Terrén

Bellaterra, 12 de Setembre de 2011

RESUMEN DE LA MEMORIA

ENGLISH

Web pages and 3D animation are two major attractions nowadays on Internet. An enjoyable and interactive web interface to navigate comfortably, with a two-dimensional easy-to-use tool to design and explore the result in three dimensions.

Those have been the basis for my application, which is a website dedicated to art, where anyone can register and draw, using a paddle, a museum to his taste. Visiting the resulting virtual museum to admire the paintings with its author and title.

CASTELLANO

Las páginas web junto a la animación 3D son dos grandes atractivos hoy en día en el mundo de Internet. Una interfaz web agradable e interactiva por la que navegar cómodamente; junto a una herramienta en dos dimensiones fácil de manejar, para diseñar y obtener un resultado en tres dimensiones.

Esas han sido las bases de mi aplicación, la cual consiste en una página web dedicada al arte, donde cualquier persona podrá registrarse y dibujar mediante una paleta como sería un museo a su gusto. Visitando el museo virtual resultante, donde admirar las obras junto a su autor y título.

CATALÀ

Les pàgines web, al costat de l'animació 3D són dos grans atractius avui dia en el món d'Internet. Una interfície web agradable i interactiva per la qual navegar còmodament, al costat d'una eina en dues dimensions fàcil de manegar, per dissenyar i obtenir un resultat en tres dimensions.

Aquestes han estat les bases del meva aplicació, la qual consisteix en una aplicació web dedicada a l'art, on qualsevol persona podrà registrar-se i dissenyar mitjançant una paleta com seria un museu al seu gust. Visitant el museu virtual resultant, es pot admirar les obres al costat del seu autor i títol.

ÍNDICE

Introducción.....	5
<i>Objetivos y motivaciones.....</i>	5
<i>Situación el a sociedad actual</i>	7
<i>Estudio de viabilidad.....</i>	9
Viabilidad técnica	9
Viabilidad económica	20
Viabilidad legal.....	21
<i>Planificación temporal del trabajo.....</i>	22
Descripción del proyecto.....	25
<i>Análisis de los requerimientos.....</i>	25
Requerimientos funcionales	25
Requerimientos no funcionales.....	26
Requerimientos del dominio	27
<i>Fundamentos teóricos.....</i>	28
Creación de una página web.....	28
Animación y gráficos	31
<i>Fases del sistema.....</i>	32
Primera fase: Construcción de una aplicación web	32
Segunda fase: Definición de una base de datos subyacente	33
Tercera fase: Dibujo de una animación en dos y tres dimensiones	33
<i>Diseño del sistema</i>	34
Actores y sus roles	34
Diagrama Caso de Uso.....	35
Ejemplo Caso de Uso.....	36
Ejemplo Diagrama de Secuencia	37
Diagrama de clases.....	38
<i>Implementación.....</i>	39
Aplicación web	39
Base de datos	42
Animación 2D y 3D	44
Comentario final.....	48
Bibliografía.....	49
Apuntes de la autora.....	49

INTRODUCCIÓN

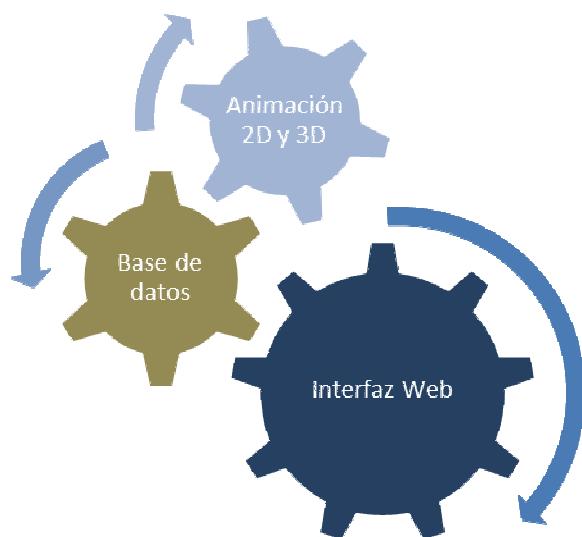
OBJETIVOS Y MOTIVACIONES

A lo largo de mi carrera como informática, he tenido un amplio aprendizaje y he estudiado múltiples áreas de aplicación. Al pasar los años me fui inclinando más por unas ramas que por otras y al final de mi licenciatura llegué a la conclusión de que el modelado 3D y la programación web era lo que más me gustaba y con lo que más disfrutaba como informática.

Motivo por el cual mi proyecto ha consistido en una aplicación con la que poder trabajar extensamente en estos dos sectores.

En concreto, mi tarea ha sido crear una herramienta con la que generar museos virtuales de una forma fácil y rápida. Mi objetivo principal ha sido “conseguir grandes resultados con pequeños útiles”, por lo que he diseñado tanto la página web como la paleta de dibujo de forma que resulten lo más intuitivas, cómodas e interactivas posibles.

Las distintas secciones de las que se compone la aplicación son:



El usuario interaccionará con éstas de la siguiente forma:

1. Una persona entra en el sitio web, apareciendo la página de Inicio (Home).
2. Navega por las distintas pestañas pudiendo consultar el manual de usuario y la información acerca de la creadora, entre otros.
3. Para acceder a la zona privada, se registra como nuevo usuario, facilitando sus datos bajo confidencialidad.
4. Una vez registrado entra en la zona de dibujo donde encuentra las herramientas necesarias para dibujar en una cuadrícula.
5. Inserta en las casillas de la rejilla distintos objetos como una puerta, una ventana, una pared, una baldosa o un cuadro (pintura).
6. Termina el diseño de su museo trazado y guarda su configuración.
7. Añade de su directorio las imágenes correspondientes a los cuadros que ha colgado en las paredes.
8. Accede entonces a la pre visualización del museo virtual equivalente al diseño anteriormente realizado.
9. Se encuentra con un entorno en 3 dimensiones donde se halla su museo, por el cual se mueve usando unas teclas específicas.
10. Recorre los pasillos, cruza las puertas y admira los cuadros expuestos en las paredes, acercándose, alejándose o desplazándose según su gusto.
11. Para terminar, visualiza el listado de cuadros compuesto por la imagen, el título y el autor de la obra.

SITUACIÓN EN LA SOCIEDAD ACTUAL

Hoy en día el arte es un sector muy extenso y demandado, son muchas las aplicaciones de éste y la gente busca cada vez más distintos medios que le acerquen y ofrezcan servicios relacionados con esta área. Es por eso que la existencia de museos en tres dimensiones es una posibilidad muy atractiva y solicitada por todos en general, ya que les permite por un lado a los expositores disponer de una galería de arte virtual en lugar de una física; y por otro lado se le permite a los clientes poder visitar online susodicho museo sin tener que desplazarse.

Esto supone un ahorro de tiempo, esfuerzo y dinero, ideología básica que persigue la sociedad moderna.

Con una aplicación para crear con sencillez una animación del museo en tres dimensiones, las empresas expositoras podrán ofrecer una visita virtual a sus galerías de arte, evitando los problemas que podrían surgir si se realizara de forma física la exposición; como son las fechas de finalización, la falta de recursos físicos, un boceto con buen aspecto en el plano pero con un resultado delusorio en la construcción física, la degradación de las pinturas con el tiempo o por manipulación, etc.

A la par, el cliente tendrá la posibilidad de evaluar el producto “sin salir de casa”, no teniendo la necesidad de ir a ver el museo en persona, lo que supone que la página web reciba visitas desde cualquier parte del mundo.

Además, actualmente la extensión de Internet en el mundo es inmensa y tener disponible un museo online supondrá aumentar considerablemente el número de clientes que visitarán la página, abriendo un abanico de posibilidades tanto a la empresa como al clientes, es aspectos como la compra y puesta en contacto entre ambos.

Este tipo de herramienta de creación de museos virtuales en 3 dimensiones ya existe en Internet y está creciendo cada día más. Ejemplos de empresas que ya proporcionan este servicio son:

(SIGUIENTE PÁGINA)

 THE REAL WORL ONLINE

*Compañía EveryScape.
Software "Scape Artist".
Massachusetts (EEUU).*

Enlace: <http://www.everscape.com/index.aspx> [Disponible: 12/09/2011]

 IMAGINARTE

*Instituto Tecnológico de Materiales.
Asturias (España).*

Enlace: <http://www.itma.es/esp/01/index.HTML> [Disponible: 12/09/2011]

 TRADKY

*Empresa Tradky Software.
Madrid (España).*

Enlace: <http://www.tradky.com> [Disponible: 12/09/2011]

 GOOGLE EARTH

Museo El Prado.

Enlace: <http://www.gooGLE.es/intl/es/landing/prado/> [Disponible: 12/09/2011]

 ANIMOGRAPH, NEW YORK TIMES

*Museos virtuales. Director Peter Kossoy.
Rusia.*

Enlace: <http://www.xocas.com/blog/?p=39> [Disponible: 12/09/2011]

 THE NEW YORK TIMES

*Sector de gráficos del periódico.
Nueva York (EEUU).*

Enlace: <http://www.xocas.com/blog/?p=32> [Disponible: 12/09/2011]

 SECOND STORY

*Gráficos interactivos.
Portland (Orlando).*

Enlace: <http://secondstory.com/> [Disponible: 12/09/2011]

ESTUDIO DE VIABILIDAD

Para desarrollar todo lo que comprende el proyecto explicado en las secciones anteriores, es necesario previamente estudiar todos los recursos disponibles y hacer una investigación acerca de qué herramientas utilizar, los costes, la legalidad subyacente, etc.

Esto permitirá preparar el entorno en el que se desarrollará la aplicación. En concreto, vamos a describir aspectos como la viabilidad técnica, la viabilidad económica y la viabilidad legal que suponen para llevar a cabo este proyecto.

VIABILIDAD TÉCNICA

Comprende todas las herramientas con las que se pueda crear la aplicación, en este caso al ser un proyecto informático, se refiere a los programas (de edición, dibujo, análisis y planificación) y a los lenguajes de programación (para la base de datos, la página web y las animaciones en dos y tres dimensiones, junto a la interacción dinámica con el usuario que sea necesaria).

Mostramos a continuación una tabla de los recursos candidatos para cada parte:

PROGRAMAS SOFTWARE

Sección	Software candidato	Software elegido
Planificación: • Diagrama de Gantt	✓ OpenProject 1.4. ✓ GanttProject 2.0.9.	✓ GanttProject 2.0.9.
Análisis: • Rol de los actores • Requisitos funcionales • Requisitos no funcionales • Casos de uso • Diagrama de clases	✓ Dia 0.97.1. ✓ ArgoUML 0.32.2. ✓ Umbrello 1.5.8. ✓ Enterprise Architect 9.1.	✓ Dia 0.97.1.

Implementación: <ul style="list-style-type: none"> • Aplicación web • Base de datos • Lienzo 2D • Espacio 3D 	<ul style="list-style-type: none"> ✓ Eclipse for Java Developers. ✓ Visual Studio 2010. ✓ gedit 2.30.2. ✓ Komodo Edit 6. ✓ NetBeans 7.0.1. ✓ Oracle. ✓ Xampp 1.7.4. 	<ul style="list-style-type: none"> ✓ Netbeans 7.0.1. ✓ Komodo Edit 6. ✓ Xampp
Redacción: <ul style="list-style-type: none"> • Memoria e informe previo 	<ul style="list-style-type: none"> ✓ Open Office Writer 3.2.1. ✓ Microsoft Office Word 2010. 	<ul style="list-style-type: none"> ✓ Microsoft Office Word 2010.
Diseño: <ul style="list-style-type: none"> • Fondos y dibujos (página web y redacción) 	<ul style="list-style-type: none"> ✓ Paint 6.1 (Windows 7). ✓ Gimp 2.6.5. ✓ Adobe Photoshop CS5. 	<ul style="list-style-type: none"> ✓ Paint 6.1. ✓ Gimp 2.6.5.

PROS Y CONTRAS DE LOS PROGRAMAS LISTADOS

PLANIFICACIÓN

 **OPEN PROJECT:** Es un programa muy completo con el que se pueden realizar distintos diagramas y vistas para la planificación de un proyecto. No obstante mi búsqueda era la de un programa que ofreciera una interfaz y unos resultados muy claros e intuitivos, de una forma rápida y sin complicaciones, motivo por el que lo he descartado.

 **GANTT PROJECT:** Programa muy parecido al anterior, con la diferencia de ofrecer una interfaz simple, atractiva y de fácil uso. Motivo por el que elegí emplearlo para crear el Diagrama de Gantt.

ANÁLISIS

- **DIA:** Programa creado para la notación UML, los diagramas de flujo y de redes, etc. Se creó como parte del proyecto Gnome. Ha sido el programa que he elegido para crear la parte del diseño ya que es muy intuitivo y fácil de usar.
- **ARGO UML:** Es el programa de código abierto, escrito en Java y con la funcionalidad de diagramado en UML. Es más completo que el programa Dia más lo descarté al no verlo necesario.
- **UMBRELLA:** Es un programa modelador de UML pero creado para usarse en la plataforma de Linux, motivo por el que lo descarté al estar trabajando desde Windows.
- **ENTREPRISE ARCHITECT:** Es una herramienta perteneciente a Sparx Systems, muy potente, con una interfaz muy atractiva, y que permite analizar y diseñar mediante UML, entre otros. Aun siendo una plataforma fuerte, la descarté por falta de necesidad.

REDACCIÓN

- **OPEN OFFICE WRITER:** Programa para editar textos de libre uso (código abierto). Es una herramienta completa, con los útiles necesarios para hacer una buena redacción. No obstante, no posee algunas de las comodidades propias de Microsoft Word como son los estilos rápidos, objetos prediseñados, etc. Motivo por el que finalmente lo descarté.
- **MICROSOFT OFFICE WORD:** Programa similar al anterior pero con el inconveniente de no ser open-source y por lo tanto necesitar licencia, o bien utilizarlo con una versión de prueba temporal.
A pesar de ello fue el programa elegido para redactar mi informe, al tener una interfaz muy variada, con estilos y gamas de colores atractivos, junto a un conjunto de herramientas muy completo y variado que facilitan bastante el trabajo. Además de ser un programa con el que estoy familiarizada desde hace muchos años.

DISEÑO

- ✚ **PAINT:** Programa de dibujo de Microsoft y creado para el sistema operativo Windows. Es de muy fácil manejo para tratar imágenes o realizar dibujos, desde el más simple al más complejo. Razón por la que elegí utilizarlo para retocar o crear diseños.
- ✚ **GIMP:** Programa open-source de manipulación de imágenes de GNU, similar al anterior pero mucho más avanzado y de uso más complejo. Es una herramienta muy útil que ofrece múltiples posibilidades para dibujar y aplicar efectos sobre el lienzo, proporcionando buenos resultados. Por ello ha sido el programa que más he utilizado para la parte del diseño.
- ✚ **ADOBE PHOTOSHOP:** Programa para editar imágenes especializado en el campo de la fotografía. Es un software muy amplio y potente de uso profesional y comercial. Más es de licencia y no lo he visto necesario para el diseño de mi aplicación, quedando entonces descartado.

IMPLEMENTACIÓN

- ✚ **ECLIPSE:** Software dedicado a la programación, con las librerías necesarias para múltiples lenguaje. Ofrece sugerencias para autocompletar a lo largo de la codificación de cada lenguaje en concreto. Sin embargo, lo descarté frente a NetBeans ya que gran parte de la interacción del usuario con la página web la he realizado en JavaScript y este segundo es más especializado.
- ✚ **VISUAL STUDIO:** Comprende un conjunto de herramientas muy avanzado para crear aplicaciones complejas de todo tipo. Es de licencia (propiedad de Microsoft) y aunque he trabajado anteriormente con él, me resulta menos práctico a la hora de programar que el resto, motivo por el que descarté utilizarlo.
- ✚ **GEDIT:** Editor de textos disponible para los sistemas operativos principales, como son Gnu/Linux, Mac Os X y Microsoft Windows. Contiene características como el resaltado de texto en distintos colores para las clases u objetos predefinidos de cada lenguaje, junto a una interfaz muy simple y fácil de usar.

Es un programa con el que he trabajado muchos años a lo largo de la carrera por su gran utilidad, más lo he descartado pues buscaba un software más completo.

- + **KOMODO EDIT:** Es un editor de texto multiplataforma y gratuito, con una interfaz cómoda y sencilla. Ofrece sugerencias, resaltado y coloreado de texto, pre visualización del resultado para aplicaciones web, etc. Está especializado en lenguajes de programación web, los cuales han sido el centro de mi aplicación. Debido a esto ha sido el elegido para realizar la implementación.
- + **NETBEANS:** Programa muy similar a Eclipse pero con una IDE integrada mejor, sobre todo si se va a desarrollar una aplicación web con el Servidor/JavaScript, el cual ha sido mi caso. Por ello fue el programa elegido para desarrollar la parte del código en JavaScript.
- + **ORACLE:** Arquitectura multiplataforma, que proporciona sistemas integrados hardware y software para negocios. Permite trabajar con bases de datos mediante el lenguaje SQL entre otros. Es muy completa y aunque la he usado en la carrera, no he visto necesario su uso, por lo que la he descartado.
- + **XAMPP:** Es un servidor web Apache que permite trabajar con MySQL, PHP y Perl. Permite simular el resultado de la aplicación web que se esté desarrollando. Lo elegí para trabajar ya que es fácil de usar y muy completo.

LENGUAJES DE PROGRAMACIÓN Y LIBRERÍAS

Sección	Software candidato	Software elegido
Aplicación web	<ul style="list-style-type: none"> ✓ HTML 5. ✓ CSS 3. ✓ JavaScript. ✓ PHP. ✓ JQuery. ✓ JSON. ✓ AJAX. ✓ XML. ✓ Scriptaculous. ✓ Yui 2 (Yahoo). 	<ul style="list-style-type: none"> ✓ HTML 5. ✓ CSS 3. ✓ JavaScript. ✓ PHP. ✓ JQuery. ✓ JSON.
Base de datos	<ul style="list-style-type: none"> ✓ MySQL. ✓ SQLServer. ✓ PosrtgreSQL. 	<ul style="list-style-type: none"> ✓ MySQL.
Lienzo dos dimensiones	<ul style="list-style-type: none"> ✓ JavaScript. ✓ Canvas (HTML 5). ✓ Flash. 	<ul style="list-style-type: none"> ✓ JavaScript. ✓ Canvas (HTML 5).
Visualización tres dimensiones	<ul style="list-style-type: none"> ✓ Canvas (HTML 5). ✓ WebGL. ✓ JavaScript. ✓ O3D. ✓ Array3D. ✓ Five 3D. ✓ Flash. 	<ul style="list-style-type: none"> ✓ Canvas (HTML5). ✓ WebGL.

PROS Y CONTRAS DE LOS LENGUAJES/LIBRERÍAS LISTADAS

APLICACIÓN WEB

 **HTML 5:** Es la última versión sacada al mercado de HTML (HyperText Markup Language), es decir, un lenguaje de programación web con el que definir la estructura básica de una página web. Entre sus desarrolladores se encuentra la W3C (World Wide Web Consortium), comunidad de desarrollo de estándares muy presente actualmente en el mundo de las páginas web.

Esta nueva versión del lenguaje ofrece muchas mejoras y es muy útil para programar este tipo de aplicaciones, por ello ha sido la base utilizada para mi interfaz web.

 **CSS 3:** Lenguaje de programación básico para una página web con el que darle estilo. Comprende una familia de hojas de estilo muy completa y de fácil aprendizaje. La versión 3 trae nuevas características como son los bordes redondeados con distintos ángulos de curvatura, exteriores o interiores; sombreado de los textos y objetos, y más efectos varios. Me ha resultado un lenguaje muy interesante para crear una interfaz suave y atractiva a la vez, por lo que lo elegí.

 **JAVASCRIPT:** Es un lenguaje orientado a objetos, influido en gran parte por el lenguaje Java, y diseñado por Netscape Communications y Mozilla Fundation. Gran parte de mi aplicación web trabaja en la interacción del cliente con el servidor, y este lenguaje está tanto del lado del cliente como del servidor, según el uso que se le esté dando. Además permite la interacción mediante la interfaz DOM (Document Object Model) para programar aplicaciones. Debido a todo ello elegí este lenguaje para la parte dinámica de la página web.

 **PHP**

Lenguaje de código libre y de alto nivel para desarrollar la interacción existente entre las distintas partes de una página web. Mediante éste se puede cambiar la estructura de las páginas, intercambiar datos y pasar variables entre ellas, enlazar el lenguaje utilizado para la base de datos con el resto de la web, etc.

Se trata de un lenguaje muy práctico, fácil y sobretodo útil ya que permite realizar muchas operaciones internas, por lo que fue elegido para desarrollar mi web.

 **JQUERY:** Es una librería de JavaScript cuyo lema es “write less, do more”, es decir, conseguir más resultados con menos código. Comprende un conjunto de funciones y tipo de lenguaje con el que realizar las operaciones básicas de toda web de una manera mucho más rápida y eficiente. Por ello elegí para emplearlo en distintas partes de mi aplicación como los envíos Post, la carga de códigos en distintos lenguajes, etc.

 **JSON:** JSON es el acrónimo de JavaScript Object Notation. Comprende un tipo de objeto caracterizado por ser sencillo y muy intuitivo, dedicado al intercambio de datos. Se creó para ofrecer una alternativa a XML para tratar los datos obtenidos en operaciones.

Es un formato muy cómodo y más práctico a mi parecer que XML, motivo por el que lo elegí para transmitir datos de unas páginas a otras.

 **AJAX:** Es la unión de los lenguajes JavaScript y XML. Una tecnología asíncrona con la que implementar aplicaciones interactivas con el cliente, de una forma dinámica y sin tener que recargar las páginas.

Es una apuesta muy interesante con buenos resultados, más fue descartada al final por usar JQuery, con el cual estaba más familiarizada.

 **XML:** De las siGLas Extensive Markup Language, es un metalenguaje desarrollado por la anteriormente mencionada corporación W3C, con el cual etiquetar y crear objetos abstractos para el conjunto de datos a intercambiar en las distintas operaciones.

Es una propuesta muy original y útil, no obstante lo descarté ante JSON pues este último me resultó más intuitivo y rápido a la hora de trabajar.

 **SCRIPTACULOUS:** Es un conjunto de librerías de JavaScript para la interfaz del usuario con el servidor. Está compuesta por pequeñas aplicaciones ya predefinidas y listas para usar, como son las animaciones, las búsquedas de autocompletado con sugerencias, efectos para la página web, etc. Su propósito es facilitar la creación de la página web con trabajo ya hecho para el programador.

Es una herramienta muy interesante más la descarté al querer realizar las distintas secciones de la web de forma manual para crearlo todo a mi gusto.

 **YUI 2:** Es una librería de CSS y JavaScript con múltiples y muy variados componentes a alto y bajo nivel para poder construir una interfaz de usuario original y atractiva al público. Sigue la misma filosofía del lenguaje anterior, es decir, facilitar el trabajo.

De todas formas lo descarté finalmente por el mismo motivo que descarté Scriptaculous.

BASE DE DATOS

 **MySQL:** Lenguaje dedicado a la gestión de bases de datos creado por la corporación MySQL Ab, propiedad de Oracle Corporation. De fácil uso y muy parecido a Oracle, lenguaje con el que estoy acostumbrada a trabajar, por eso lo he elegido.

 **SQL SERVER:** Herramienta de programación de bases de datos propiedad de Microsoft, que permite el análisis y administración de datos. Garantiza un tiempo de respuesta óptimo para las consultas y un uso de la memoria y de la CPU bien aprovechado.

Decidí descartarlo ante MySQL por ser de licencia y más complicado.

 **POSTGRESQL:** Es un recurso de libre uso para sistemas de bases de datos orientados a objetos. Lleva más de 15 años en el mercado y proporciona por lo tanto una fuerte arquitectura basada en la experiencia.

Lo he descartado por el mismo motivo que SQLServer.

LIENZO DOS DIMENSIONES

 **JAVASCRIPT Y CANVAS (HTML 5):** Canvas no es más que una etiqueta perteneciente al nuevo HTML5, creada para poder visualizar e interaccionar con dibujos realizados tanto en dos como en tres dimensiones. Para establecer la comunicación lienzo-usuario es necesario un lenguaje mediador, es decir, JavaScript.

Ambos son dos herramientas fuertes para la animación 2/3D en el mundo de Internet, motivo por el que elegí utilizarlos.

 **FLASH:** Adobe Flash es un lenguaje para realizar animaciones, popular actualmente por sus aplicaciones en los móviles, como son el sistema Android y la tableta BlackBerry.

Sus inconvenientes son que es de pago y requiere un tiempo de aprendizaje previo considerable, además de la necesidad de instalar el reproductor Adobe Flash Player en el ordenador. Estas son las razones por las que lo descarté.

VISUALIZACIÓN TRES DIMENSIONES

 **JAVASCRIPT Y CANVAS (HTML 5):** Anteriormente explicados [3](#).

 **FLASH:** Anteriormente explicado [4](#).

 **WEBGL:** Es la especificación estándar basada en el lenguaje OpenGL, que permite su uso mediante el lenguaje JavaScript y reproduce los resultados en la página web mediante la etiqueta Canvas de HTML anteriormente explicada.

Es una herramienta nueva para la creación de gráficos 3D en navegadores web y resulta muy fácil de usar tras un rápido aprendizaje. Decidí apostar por esta nueva tecnología así que fue la elegida para dibujar en tres dimensiones.

 **O3D:** Es una API lanzada por GooGLE y Mozilla hace pocos años para el desarrollo de aplicaciones interactivas 3D. Es una plataforma en construcción que ofrece muy buenos resultados.

Aunque sea una herramienta potente y nueva, decidí descartarla ante WebGL al estar muy familiarizada con el lenguaje OpenGL.

Conclusión

Como se ha podido ver tras la explicación de la motivación para usar un programa u otro, en un lenguaje determinado, me he inclinado por un lado por aquellos programas que me permitieran trabajar con licencias más abiertas o libres, ya que opino que existen muchos programas de calidad que son de código libre y que ofrecen muchas herramientas.

Por otro lado he elegido aquellos lenguajes de programación que me han resultado más interesantes, en concreto he apostado por todos aquellos que son nuevos o con nuevas versiones. La principal motivación de mi proyecto ha sido realizar una investigación y aprender lenguajes actuales, presentes y utilizados en el mercado moderno para el mundo del desarrollo web.

Por último y para fundamentar mi investigación incluyo una gráfica (Figura 1) del mes de Septiembre de 2011 de la comunidad Tiobe¹, con la situación actual del uso de los lenguajes en la programación web. En ésta sólo se muestran algunos de los muchos existentes, más se puede ver que tanto JavaScript como PHP son dos de las grandes bases para Internet.

Vemos esto a continuación:

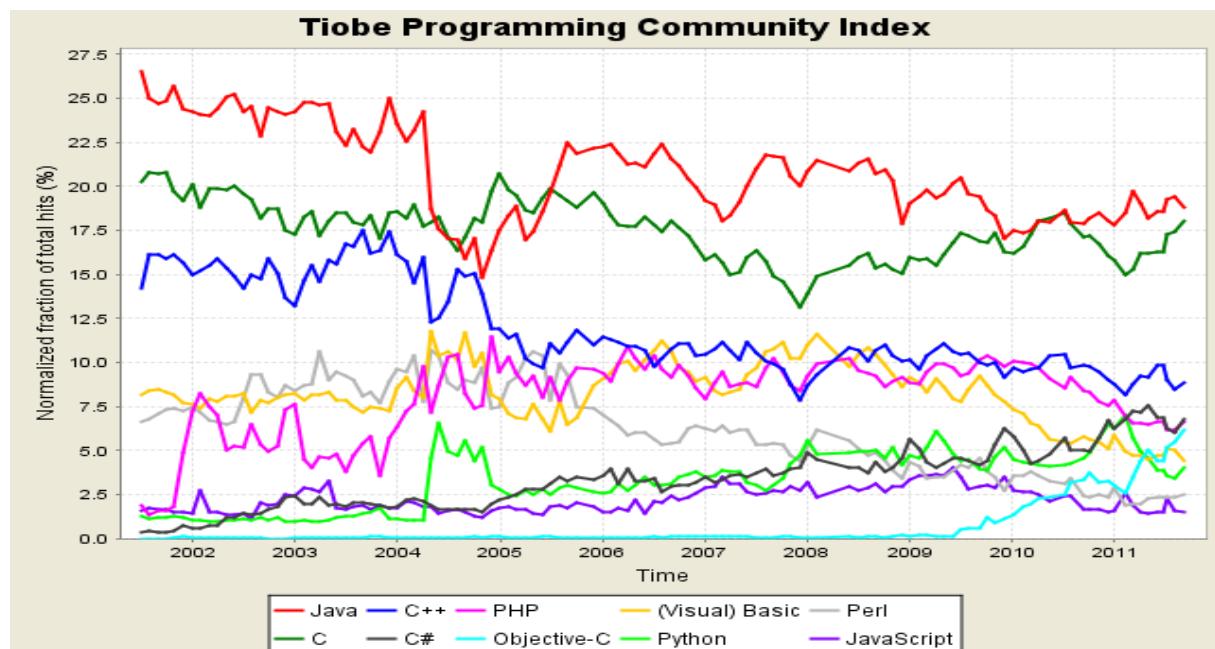


Figura 1. Tendencias a largo plazo en el Top 10 de los lenguajes de programación.

¹ TIOBE Programming Community Index, enlace:
<http://www.tiobe.com/index.PHP/content/paperinfo/tpci/index.HTML>

VIABILIDAD ECONÓMICA

Respecto a los costes que conlleva realizar este tipo de proyecto, podemos determinarlos en función de distintos aspectos, como son:

MEDIOS HARDWARE

He empleado únicamente mi portátil para trabajar, el cual es de la marca **Acer** y está valorado en 900€, con 5 años de vida estimados. Han sido aproximadamente 11 meses de trabajo, más adelante se puede consultar la planificación final que se ha seguido mediante un Diagrama de Gantt [2](#).

HERRAMIENTAS SOFTWARE

Como se ha indicado anteriormente, he optado por los programas de libre uso o con versiones de evaluación, por lo que el coste de las licencias no añadirá al coste final.

TOTAL DE HORAS POR PERSONA

He desempeñado el papel de diseñadora, analista, programadora y redactora; facetas propias como informática más necesarias para la vida real.

En definitiva, el coste total quedaría de la siguiente manera:

Sección	Composición	Coste
Hardware	11 meses de 5 años de vida estimados	~18% de 900€ → 162 €
Software	Open-source	0€
Horas/persona	Proyecto de Fin de Carrera ↓ Asignatura de 15 créditos ↓ Si 1 crédito son 10 horas ↓	<ul style="list-style-type: none"> Diseñadora: ~20€/h, 20% de 150h = 30 horas Total: 20 € · 30h = 600 € Analista: ~30€/h, 30% de 150h = 45 horas

Total: $15 \text{ cr} \cdot 10 \text{ h} = 150 \text{ horas.}$	Total: $30 \cdot 45 \text{h} = 1350 \text{ €}$ <ul style="list-style-type: none"> • Programadora (aprendizaje e implementación): $\sim 40\text{€}/\text{h}, 40\% \text{ de } 150\text{h} = 60 \text{ horas}$ • Redactora: $\sim 10\text{€}/\text{h}, 10\% \text{ de } 150\text{h} = 15 \text{ horas}$
TOTAL	4.662 €

Se puede observar cómo con respecto al anterior estudio económico realizado en el Informe Previo, han aumentado el número de meses de trabajo, por lo que el **coste final** se ha visto **incrementado**.

VIABILIDAD LEGAL

Todo producto a ofrecer al público, que tenga pues contacto con las personas en un futuro, deberá garantizar seguridad, confidencialidad y privacidad a la hora de ser usado. Estos son los principales requisitos que toda aplicación en Internet debe tener. Respecto al trato con el cliente, se ha implementado la página web de modo que proporcione la protección de datos de los usuarios registrados que brindan tecnologías como **PHP** y **MySQL**.

Los únicos conocedores de la información privada de contacto serán el cliente y el proyectista. El acceso sólo será posible con un usuario y una contraseña, la cual se proteje en todo momento con el algoritmo de codificación **Md5**, y éstos sólo los conocerá el cliente, por lo que sólo él podrá acceder a sus datos, al boceto de su museo en 2D y al resultado final de éste en 3D. De este modo se pretende asegurar que no se distribuirán los datos privados a cualquier empresa o compañía publicitaria. Evitando accesos indeseados o ataques de bots, proporcionando así un entorno seguro y fiable.

PLANIFICACIÓN TEMPORAL DEL TRABAJO

Mostramos a continuación un diagrama (Figura 2) acerca de la distribución del trabajo que ha comprendido llevar a cabo el proyecto, donde se indica la duración de cada una de las actividades realizadas, ordenadas por meses cronológicamente.

Para ello se ha realizado un diagrama de Gantt de todo el curso (desde Octubre de 2010 hasta Septiembre de 2011), repartido en investigación, modelado, desarrollo y redacción de la memoria. Las fechas indicadas han sido las definitivas tras terminar el trabajo, por lo que han cambiado respecto a la predicción inicial del Informe Previo.

En este diagrama se puede observar a la izquierda, un listado con las distintas tareas desarrolladas a lo largo del tiempo, junto al desGlose de sub tareas de las que han estado compuestas. A la derecha le sigue un gráfico donde cada barra representa a una tarea distinta, en la que está representado el nombre de la tarea, la fecha de inicio y de finalización, y la duración de ésta.

Veamos el resultado final:

(SIGUIENTE PÁGINA)

DIAGRAMA DE GANTT (I/II)

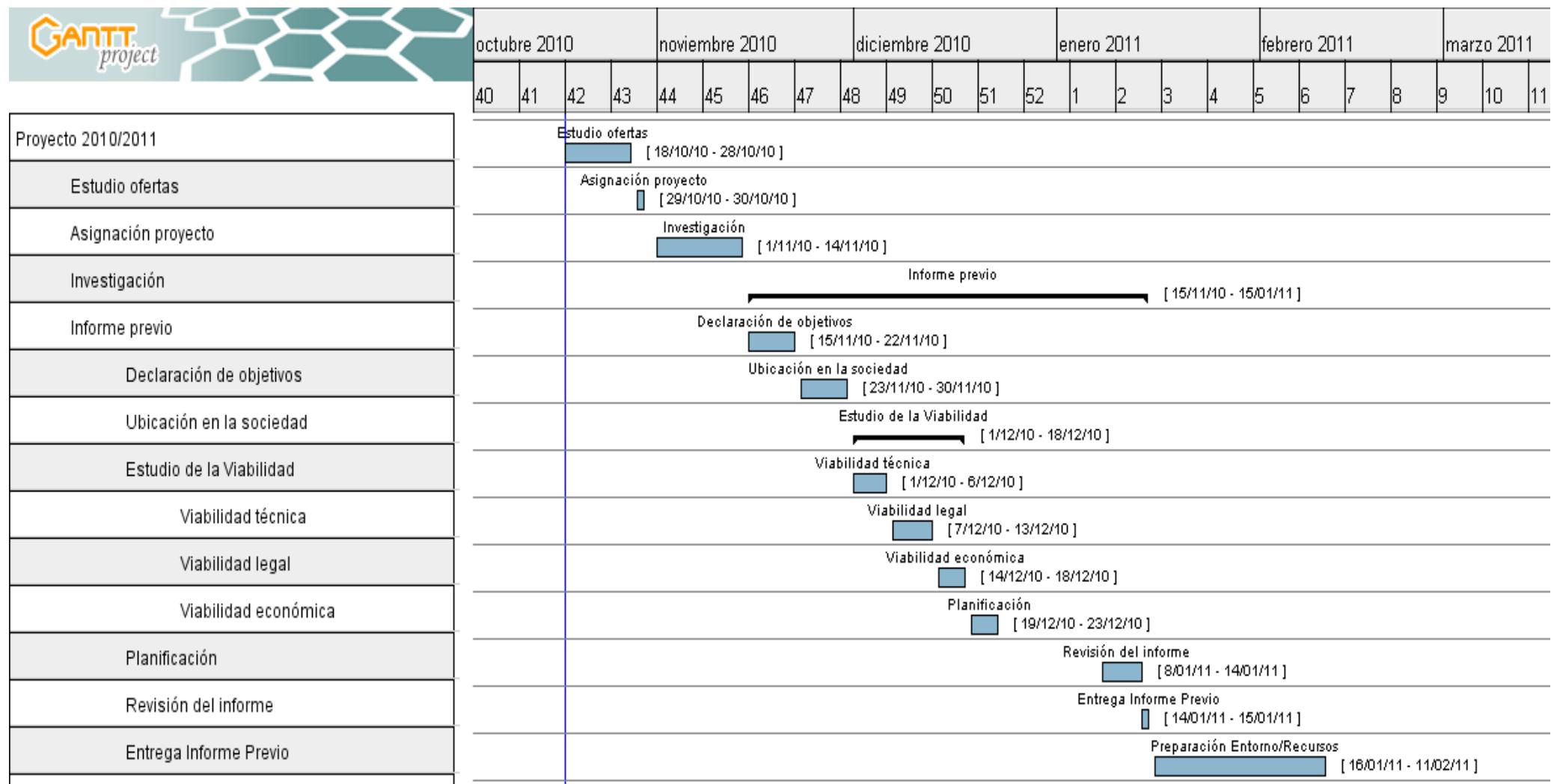
**Figura 2.** Diagrama de Gantt proyecto Marta Laperal Martín 2010/2011.

DIAGRAMA DE GANTT (II/II)

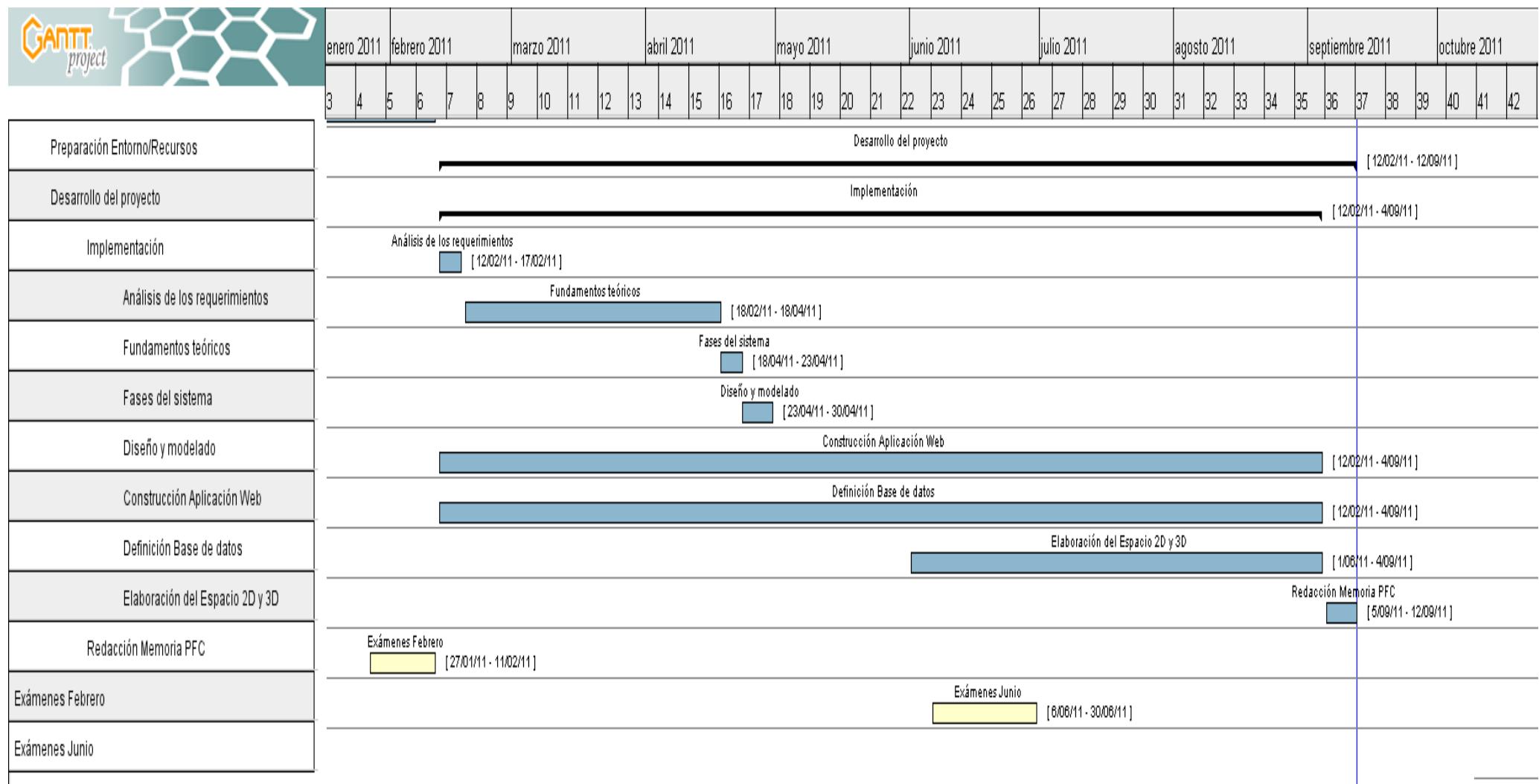


Figura 2. Diagrama de Gantt proyecto Marta Laperal Martín 2010/2011.

DESCRIPCIÓN DEL PROYECTO

ANÁLISIS DE LOS REQUERIMIENTOS

Una vez realizada la introducción, en la que se ha hablado de los objetivos, la viabilidad y la planificación del proyecto; entraremos más en materia especificando cuáles son los requerimientos software que posee la aplicación realizada.

En este estudio tendremos en cuenta tanto los aspectos físicos como los lógicos, indicaremos así los requerimientos que proporcionen a la aplicación características como son la funcionalidad, consistencia, ausencia de ambigüedad, verificabilidad, trazabilidad y la disponibilidad ante modificaciones.

REQUISITOS FUNCIONALES

Comprenden aquellos servicios que el sistema software debería proporcionar y el comportamiento de éste ante distintas situaciones:

- ❖ El usuario podrá acceder a todas las secciones de la página web y en concreto podrá registrarse como nuevo cliente facilitando sus datos personales en el formulario disponible para tal efecto.
- ❖ El usuario tendrá acceso a la zona reservada a clientes introduciendo su nombre y contraseña en el área de identificación.
- ❖ El sistema facilitara al usuario una herramienta de dibujo en la zona privada, mediante la cual realizar el diseño de un museo plano (dos dimensiones).
- ❖ El sistema permitirá guardar el diseño realizado por el usuario, garantizando la disponibilidad de éste cada vez que el usuario vuelva a entrar en la zona de dibujo.
- ❖ El sistema ofrecerá la posibilidad de cargar archivos de tipo imagen para que el usuario pueda subir fotografías directamente desde un directorio de su ordenador.

- ❖ Una vez realizado el diseño del museo plano y la carga de imágenes al servidor, el sistema permitirá al usuario visualizar el mismo museo diseñado con las imágenes indicadas, ahora en tres dimensiones.
- ❖ El sistema permitirá pues al usuario en la zona 3D, poder moverse a través del museo virtual, visualizando el exterior e interior del museo, además de los cuadros de las pinturas cargadas.
- ❖ El sistema ofrecerá además al usuario un listado de las obras de su museo, donde verá una miniatura del cuadro junto a una descripción con el autor y título de la pintura.
- ❖ En todo momento el sistema se encargará de almacenar en la base de datos del servidor, todos aquellos útiles y datos suministrados por el usuario, que comprenden sus datos personales, el diseño en dos dimensiones del museo dibujado y la visualización del museo resultante en tres dimensiones.
- ❖ Además el sistema deberá separar los datos privados de cada usuario.
- ❖ Por último, el sistema intentará facilitar lo máximo posible la tarea del usuario a lo largo de la aplicación: mostrándole sugerencias o aclaraciones, notificando las operaciones realizadas con éxito y avisándole de los fallos o restricciones del sistema.

REQUISITOS NO FUNCIONALES

Este tipo de requisitos se caracterizan por ser restricciones, características técnicas y legales del sistema, etc. Para mi aplicación son los siguientes:

- ❖ Ha de ofrecer un servicio con el que permitir a cualquier persona acceder a la página web, con una interfaz cómoda, nítida y fácilmente navegable.
- ❖ Respecto al tiempo de respuesta, el sistema software garantizará un **funcionamiento rápido, eficaz y sostenible**; tanto de la página web como de las operaciones internas no visibles al usuario.
- ❖ El usuario trabajará con un sistema **fácil de usar**, que buscará una interacción del usuario con el sistema rápida, sencilla y sobretodo, sin necesidad de aprendizaje previo o conocimientos requeridos para utilizar la aplicación.
No obstante el sistema ofrecerá al usuario un manual de uso para esclarecer su tarea y resolver dudas surgidas durante su navegación por la página.

- ❖ Basándose en la **confiabilidad**, el sistema protegerá en todo momento los datos relativos a cada usuario, que abarcan desde los personales hasta el diseño y resultado final del museo.
- ❖ Para garantizar la **seguridad** del sistema, no se permitirá a ninguna otra persona, bot o programa intrusivo acceder a los datos de un usuario registrado.
- ❖ Asimismo el sistema deberá ofrecer la **disponibilidad** en todo momento a los usuarios registrados, de sus diseños y datos de contacto.
- ❖ De tal modo, el sistema para ser robusto deberá tener un sistema de **almacenamiento eficiente**, seguro, suficientemente potente para ofrecer tal servicio y que no dé lugar a errores.
- ❖ Bajo ningún caso el sistema podrá tener problemas como la pérdida, difusión o insostenibilidad de los datos.
- ❖ Acerca de la **compatibilidad**, el sistema estará disponible para usarse en todos aquellos sistemas operativos y navegadores de páginas web que soporten las herramientas utilizadas.
- ❖ El sistema se desarrollará con software de libre uso o licencias abiertas (pruebas de evaluación), por lo que garantizará un **servicio** creado de forma **legal y responsable**.

REQUERIMIENTOS DEL DOMINIO

Hacen referencia a aquellos requisitos que definen los rasgos, restricciones y limitaciones del sistema. Por ello deben satisfacerse pues en caso contrario el sistema podría ser impracticable.

En este caso serán los siguientes:

- ❖ El sistema ha de ser capaz de trabajar con todos los lenguajes requeridos para crear la aplicación. En concreto permitirá entre otros el uso de lenguajes de programación web como **HTML 5, CSS 3** y **WebGL**, bases centrales de mi aplicación.
- ❖ El sistema deberá usarse con navegadores web que soporten todos los lenguajes con los que se creó la aplicación, los cuales actualmente son **Mozilla Firefox, GooGLE Chrome** y **Safari**. El uso en el navegador Internet Explorer está aún en desarrollo, por lo que aún no tiene soporte.

- ❖ Específicamente, para que funcione WebGL, la gráfica del ordenador que se utilice deberá soportar **OpenGL ES** versión 2.0.
- ❖ Al usar WebGL para la parte gráfica, **no** será necesario instalar ningún **plugin** adicional en el caso de Chrome, para Firefox es necesario instalar “Nightly” y configurar el navegador para permitir reproducir WebGL, mientras que para Safari esta herramienta viene integrada más es necesario activarlo en el navegador.

FUNDAMENTOS TEÓRICOS

Mi proyecto se ha centrado en investigar acerca de aquellas tecnologías que trabajan hoy en día en los sectores de la programación web y la animación. Hablaré de ellos por separado:

CREACIÓN DE UNA PÁGINA WEB

La programación de sitios web es un mundo muy extenso en el que existen muchos y variados lenguajes de programación con los que construir una web. En este caso hablaré de los dos cuatro lenguajes principales con los que funciona mi aplicación, que son [HTML 5, CSS 3, JavaScript y PHP](#).

HTML 5

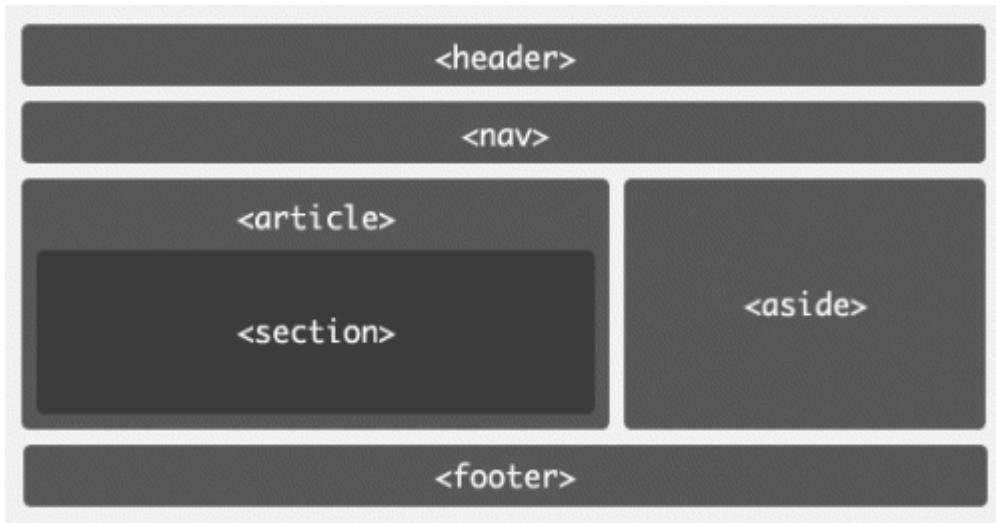
Es la última versión del lenguaje de marcación HTML², con el cual se define la estructura de una página web, indicando cuales son cada una de las partes en las que ésta se compone. Con ello se pretende definir de una forma fácil cómo es el contenido de la página que se esté explorando.

Concretamente, HTML 5 se creó con el propósito de agrupar distintas zonas de la página web para ahorrar trabajo y hacer el código más intuitivo y legible.

Este lenguaje define cada una de esas zonas mediante etiquetas, las cuales poseen unos atributos y operaciones asociadas.

² HyperText Markup Language, versión 5. Enlace: http://es.wikipedia.org/wiki/HTML_5

Principalmente, la nueva estructuración de páginas que propone HTML 5 es la siguiente:



Podemos hablar resumidamente de estas nuevas etiquetas:

- **Header**: Representa a la cabecera de la página web, normalmente contiene el título y logotipo del autor o empresa creadora de la página.
- **Nav**: Permite definir un elemento con el que navegar por la página web, como sería un menú o barra principal/secundarios.
- **Article**: Especifica una zona independiente al resto del contenido de la página web, como por ejemplo artículos, comentarios, blogs, foros, etc.
- **Section**: Como su nombre bien indica, se trata de una sección dentro de un artículo donde pueden haber distintos apartados del mismo tema.
- **Aside**: Puede entenderse como un lateral, una zona aparte de la página web donde se suele insertar publicidad, enlaces externos o cualquier información al margen del artículo.
- **Footer**: No es más que el pie de página de nuestra interfaz web, suele contener información acerca del creador de la página, como su nombre, ubicación, datos de contacto y derechos legales.

CSS 3

Es la última versión del lenguaje CSS³, encargado de dar estilo a la página web, con él se define el tipo de letra utilizada (color, tamaño, tipografía, etc.); aspecto de las imágenes, fondos y dibujos que aparezcan en la página; efectos y comportamiento en la interacción interacción con el cliente; y un largo etc.

³ Cascade Style Sheet, versión 3. Enlace: http://es.wikipedia.org/wiki/Hojas_de_estilo_en_cascada

Más específicamente, esta nueva versión de CSS nos trae algunas propiedades como:

- **Bordes:** Como son las propiedades **border-image**, **border-radius** y **box-shadow**. Mediante las cuales se pueden definir los bordes con una imagen, esquinas redondeadas y sombras bajo el borde.
- **Fondos:** Nuevas propiedades como **background-origin**, **background-clip**, **background-size**. Con las que establecer la posición de la imagen, su tamaño, etc. Además de poder crear un solo fondo con múltiples capas de imágenes.
- **Colores:** Se ofrecen nuevas gamas de colores, con las que definir de modo distinto la escala cromática a usar en nuestra web. Permiten entre otros efectos como la trasparencia. Son las gamas **HSL**, **HSLA** y **RGBA**.
- **Texto:** Nuevos efectos para el texto como sombra propia (**text-shadow**) y rotura de palabras demasiado largas o que no caben en el espacio dado (**text-overflow**).
- Etc.

JAVASCRIPT

Este lenguaje⁴ se encarga de toda la parte dinámica de la aplicación, es la herramienta básica con la que trabaja **WebGL** y se encarga de definir, gestionar y controlar toda la interacción del usuario con la página web, además de permitir cambiar el aspecto de ésta y ofrecer avisos al usuario imprimiendo mensajes sobre la pantalla. Además posee entre otras una librería, **JQuery**, que contiene funciones predefinidas que facilitan mucho la tarea del programador y permiten entre otros, cargar trozos de códigos en distintos lenguajes, cambiar el aspecto de la página web de forma dinámica (sin recargar), etc.

PHP

PHP⁵ es un lenguaje de programación web creado en 1994 y que hoy en día es uno de los puntos fuertes de las páginas web.

Es un lenguaje del lado del servidor encargado de gestionar, al igual que JavaScript, toda la interacción del usuario con la página web.

Especialmente, mediante este lenguaje se pueden almacenar datos para llevarlos de una página web a otra (enviados/recibidos mediante Post, Get, Files o incluso empleando arrays como los objetos **JSON** y **XML**); intercambiar entre ellas variables útiles para la respuesta ante eventos; etc. Además sirve de mediador entre la página web y la base de datos subyacente, mediante **MySQL** es posible almacenar, modificar, eliminar y consultar datos del cliente y de la página en sí.

⁴ Lenguaje orientado a objetos JavaScript. Enlace: <http://es.wikipedia.org/wiki/JavaScript>

⁵ PHP: Hipertext Preprocessor. Enlace: <http://es.wikipedia.org/wiki/PHP>

ANIMACIÓN Y GRÁFICOS

CANVAS (HTML 5)

Se trata de una nueva etiqueta perteneciente al lenguaje HTML 5. Canvas⁶ no es más que un enlace entre la página web y el reproductor de gráficos en dos y tres dimensiones.

Mediante esta etiqueta es posible dibujar sobre un lienzo incrustado en la pantalla, con dimensiones a definir y contenido totalmente editable. Permite además la interacción del usuario con el lienzo, ofreciéndole la posibilidad de dibujar en él, moverse por el mundo 3D, o definir eventos y comunicación entre ambas partes (lienzo y usuario).

WEBGL

WebGL⁷ es una nueva tecnología aún en desarrollo y por tanto no disponible en todos los navegadores, dedicada al mundo de los gráficos en las páginas web. Se trata de una API multiplataforma basada en el lenguaje OpenGL versión 2, que se puede incrustar en las páginas web como se ha explicado anteriormente, con la etiqueta Canvas de HTML 5.

Nociones a tener en cuenta según cómo funciona su base de OpenGL son: al definir los **vértices** de una figura se han de establecer los índices en sentido contrario a las agujas del reloj; asimismo las transformaciones sobre los objetos se harán mediante operaciones matemáticas que serán con signo positivo si por ejemplo se rota en sentido contrario a las agujas de reloj, mientras que tendrán signo negativo si se hace en sentido contrario; etc.

⁶ Canvas: Gráficos en páginas web. Enlace: <http://es.wikipedia.org/wiki/Canvas>

⁷ Khronos Group. WebGL. Enlace: <https://www.khronos.org/registry/WebGL/specs/1.0/>

FASES DEL SISTEMA

Para poder llevar a cabo una aplicación como la de mi proyecto, son necesarias muchas y variadas herramientas, no obstante se puede establecer con claridad que las fases en las que se ha dividido la creación de mi proyecto son las siguientes:

1. Construcción de una **aplicación web**.
2. Definición de una **base de datos** subyacente.
3. Dibujo de una **animación** en dos y en tres dimensiones.

PRIMERA FASE: CONSTRUCCIÓN DE UNA APLICACIÓN WEB

Se puede decir que es la base del sistema y la parte más importante, ya que sólo una vez creada la página web es posible entonces añadirle características como la interacción con la base de datos y la reproducción de animaciones.

En concreto se tratará de una página web compuesta por 5 secciones:

- **Inicio:** Página principal donde se visualice el título, la zona de identificación, el menú de navegación, una imagen ilustrativa de un museo, una breve explicación de la funcionalidad de la web y un pie de página donde se indique el nombre de la autora.
- **Registro:** Página secundaria donde el usuario se podrá registrar indicando sus datos personales, los cuales se enviarán a la base de datos para almacenarlos en el sistema.
- **Acerca de:** Página secundaria dedicada a la autora, donde se muestra información acerca de su procedencia, estudios, explicación del proyecto y motivación.
- **Museo Virtual:** Página secundaria en la que se podrá crear el museo para posteriormente visualizarlo. Se trata de una herramienta para diseñar el museo en dos dimensiones a partir de una rejilla en la que cada cuadro representa un objeto (suelo, pared, ventana, puerta o pintura). Al finalizar el dibujo se cargan las fotos correspondientes a los cuadros desde el directorio del usuario y para terminar se accede a la visualización en tres dimensiones con el resultado del museo diseñado. En esta última sección el usuario se podrá mover a través de museo con unas teclas determinadas y consultar aparte un listado con las obras y su información respectiva.
- **Manual:** Página secundaria donde se encuentra la sección de ayuda, en la que el usuario podrá consultar cómo utilizar tanto la página web como el lienzo 2D y la animación 3D.

SEGUNDA FASE: DEFINICIÓN DE UNA BASE DE DATOS SUBYACENTE

La base de datos será la encargada de almacenar los datos con los que trabaje la página web y aquellos asociados a todos los usuarios registrados.

En concreto tendremos las siguientes secciones:

- Listado de clientes junto a sus datos personales.
- Definición de la composición de cada museo por usuario, donde se indicarán del diseño en la rejilla 2D las coordenadas (x,y) en el lienzo de dibujo y un identificador del objeto insertado en esa casilla.
- Conjunto de cuadros de cada museo por usuario, de cada cual se indicará la ruta de la imagen asociada (en el directorio), la coordenada (x,y) donde se encuentra el objeto cuadro y el número que ocupa el cuadro según el lugar que tenga dentro del museo.

A lo largo de la aplicación, esta información se va modificando conforme se creen los usuarios, se diseñen los museos y se cambie el contenido de todo ello.

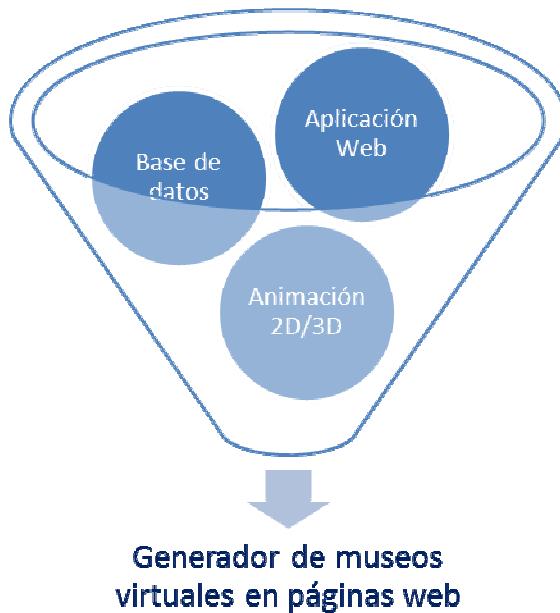
TERCERA FASE: DIBUJO DE UNA ANIMACIÓN EN DOS Y EN TRES DIMENSIONES

Para poder reproducir gráficos en el navegador es necesario:

- Crear un lienzo de dibujo con unas dimensiones determinadas.
- Ubicarlo en una posición específica de la página web y darle estilo.
- Definir los atributos que tendrá la escena como el color, la iluminación, la transparencia, etc.
- Crear un contexto en dos dimensiones para la zona donde se dibuja en la rejilla.
- Crear un contexto en tres dimensiones en la sección donde se reproduce e interacciona con la animación del museo virtual.
- Definir un traductor (enlace) entre el diseño generado en dos dimensiones para transformarlo en una animación en tres dimensiones.
- Implementar las funciones que establezcan las operaciones necesarias para modificar, iluminar, insertar o eliminar objetos en la cuadrilla dos dimensiones.
- Por último implementar además las funciones que a partir de los objetos insertados en la cuadrilla, crean un espacio tridimensional donde se levante y dé volumen a estos objetos, dibujándolos en la posición que le correspondía en la rejilla; pero esta vez con un alto, ancho, largo, textura y color.

DISEÑO DEL SISTEMA

Como se ha explicado anteriormente, mi aplicación consiste en una interacción imprescindible entre los tres pilares bajo los que funciona, que son:



Para implementar cada una de estas partes descritas en las distintas fases de construcción, antes es necesario definir un modelo a seguir con la funcionalidad de un boceto y a la vez una base a partir de la cual construir la aplicación.

Para ello definiremos el conjunto de actores que interactuarán con el sistema en las distintas operaciones:

ACTORES Y SUS ROLES

Representan el conjunto de papeles que pueden tomar los usuarios del sistema. En nuestro caso serán los siguientes:

CLIENTE

Actor primario: Será aquella persona que acceda a la página web, se registre y haga uso de las herramientas que ésta ofrece.

ADMINISTRADOR

Actor primario: Representa al sistema en sí, encargado de realizar todas las operaciones necesarias que permitan la interacción entre el cliente y el navegador web.

SERVIDOR

Actor secundario: Su utilidad es la de garantizar el funcionamiento del servidor, interpretando las órdenes dadas en el código generado y administrando la base de datos según sea requerido por el administrador.

NAVEGADOR WEB

Actor secundario: Herramienta básica mediante la cual el sistema puede funcionar, es el medio con el que se visualiza la aplicación.

A partir de este conjunto de actores podemos crear el equivalente Diagrama de Caso de Uso (Figura 3):

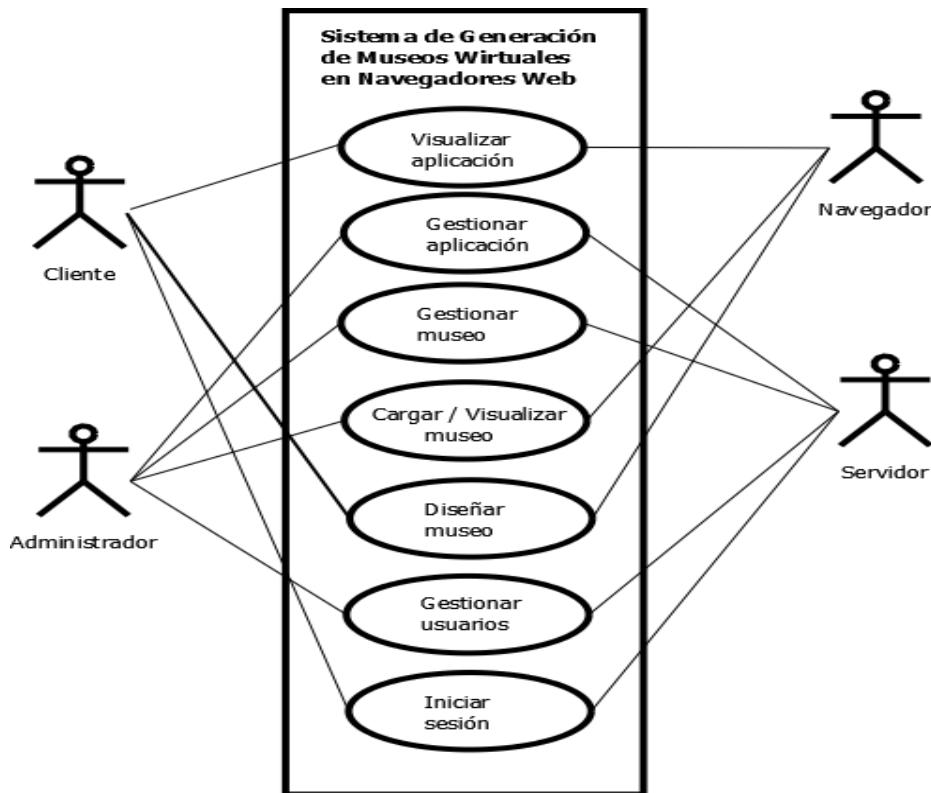


Figura 3. Diagrama Caso de Uso “Generación de Museos Virtuales para Navegadores”

Además mostraremos continuación cuál sería la descripción del Caso de Uso concreto de la acción “Cargar/Visualizar Museo” (Figura 4):

Nombre:	Cargar-Visualizar Museo
Descripción:	<p>Se muestra al cliente el museo almacenado en la base de datos, en forma de cuadrilla si está en la zona 2D o bien como animación si está en la zona 3D.</p>
Actores:	<p>Administrador del sistema.</p> <p>Navegador web en marcha.</p>
Precondiciones:	<p>El cliente debe estar previa y correctamente identificado.</p> <p>El cliente ha accedido a la zona 2D o a la zona 3D.</p>
Flujo Normal:	<ol style="list-style-type: none"> 1. Si el cliente ha accedido a la zona 2D: <ol style="list-style-type: none"> 1.1. El administrador emplea el nombre de usuario del cliente para consultar en la base de datos si existe un museo con ese identificativo. 1.2. Si existe, el administrador extrae la información acerca de la rejilla contenedora de los objetos relativos al museo. 1.3. El administrador se la facilita a la aplicación cargándola en el sistema, momento en el que el navegador reproduce la animación en dos dimensiones de la rejilla equivalente. 2. Si el cliente ha accedido a la zona 3D: <ol style="list-style-type: none"> 1.1. El administrador emplea el nombre de usuario del cliente para consultar en la base de datos si existe un museo con ese identificativo. 1.2. El administrador extrae la información acerca de la rejilla contenedora de los objetos relativos al museo. 1.3. El administrador utiliza tal información para generar mediante código WebGL la equivalencia a esa rejilla, creando un mundo en 3D. 1.4. El administrador facilita el resultado implementado a la aplicación, la cual se encarga de reproducir una animación en tres dimensiones con un museo virtual.
Flujo Alternativo:	<ol style="list-style-type: none"> 1.2. No existe ningún museo guardado para ese cliente. Entonces el administrador cargará en el sistema una rejilla con una configuración inicial, que sirva de plantilla para el diseño. 2.2. A este apartado sólo se puede acceder si existe un museo relativo a ese usuario, por lo que existirá en la base de datos.
Postcondiciones:	<p>Ha sido cargada y se visualiza la rejilla (zona 2D) o el museo virtual (zona 3D).</p> <p>Ahora el usuario puede interaccionar con ambas animaciones.</p>

Figura 4. Caso de Uso “Cargar-Visualizar Museo”

Para terminar con la parte del diseño indicaremos cómo sería el intercambio de mensajes en un momento dado, en concreto mostramos el Diagrama de Secuencia “Iniciar Sesión” (Figura 5):

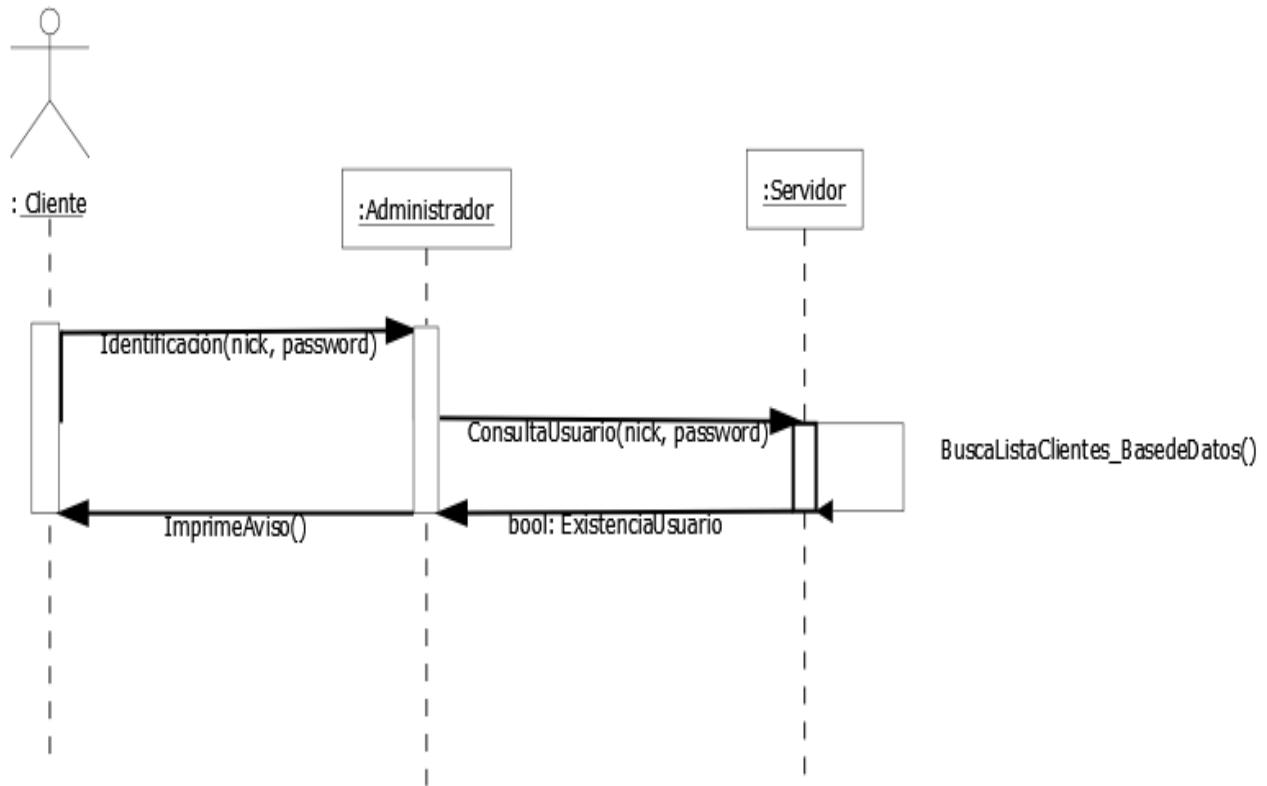


Figura 5. Diagrama de secuencia “Iniciar Sesión”.

Construimos finalmente un Diagrama de Clases (Figura 6) donde se definirán todos aquellos elementos en los que se puede clasificar el sistema, junto a sus atributos y las relaciones entre ellos:

(SIGUIENTE PÁGINA)

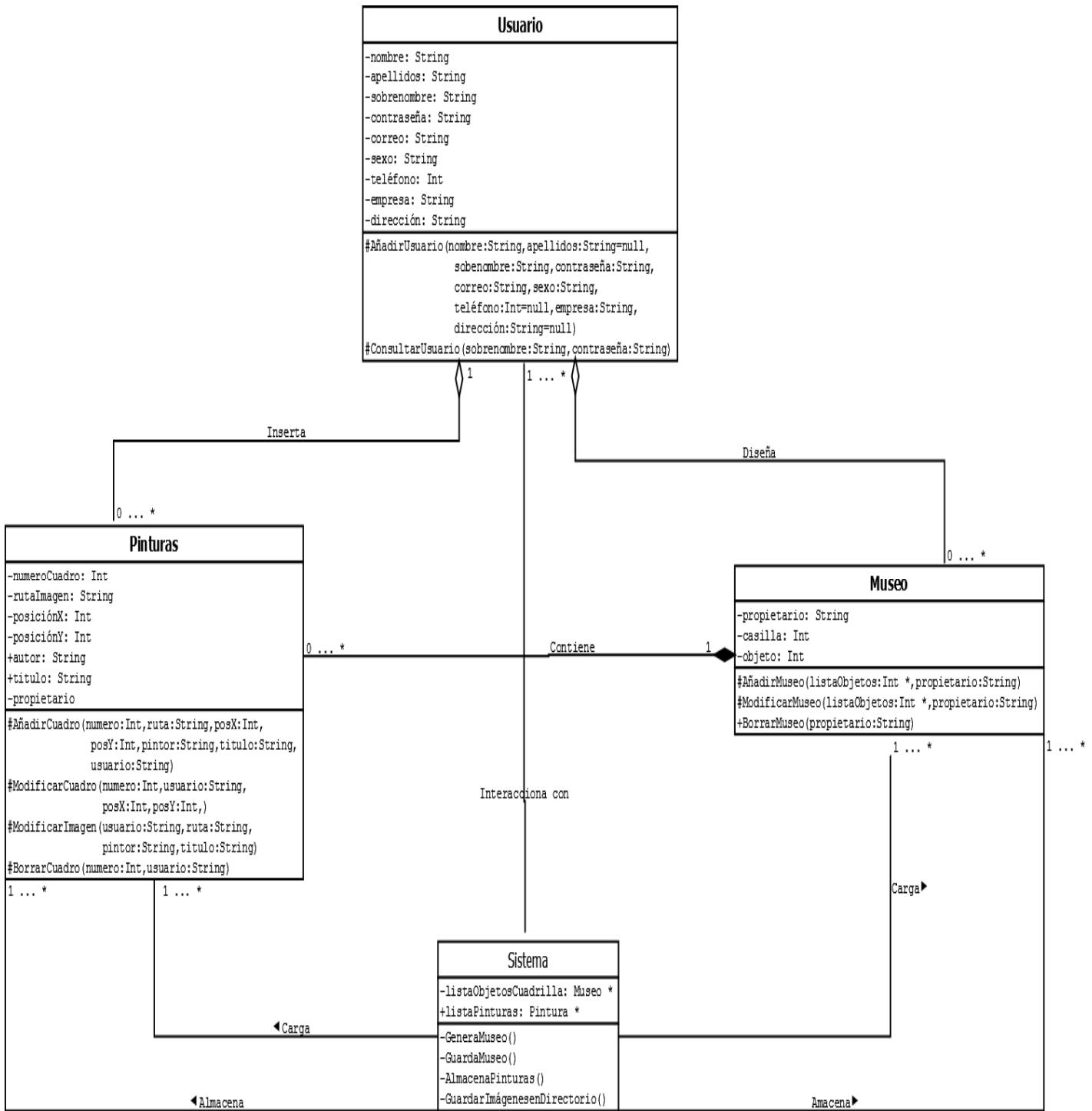


Figura 6. Diagrama de las clases “Generador de museos virtuales en navegadores web”

IMPLEMENTACIÓN

Respecto al código utilizado y los útiles necesarios para poder programar cada una de las fases anteriormente descritas, se puede explicar de la siguiente manera (siguiendo el modelo de las partes en las que se divide la aplicación):

APLICACIÓN WEB

Para realizar la página web y todo lo que ella comporta, se ha utilizado: el lenguaje **HTML 5** para declarar su estructura; **CSS 3** para darle estilo; **JavaScript** para trabajar con las animaciones; **JQuery** para cambiar el contenido de las páginas dinámicamente; **JSON** para almacenar la información del museo 2D y de las fotos cargadas del directorio; **PHP** para intercambiar información entre las distintas partes de la página web y **MySQL** para crear y gestionar una base de datos donde almacenar la información del cliente y de sus diseños en dos y tres dimensiones.

Mostramos el resultado con una captura (Figura 7) de la página de Inicio:

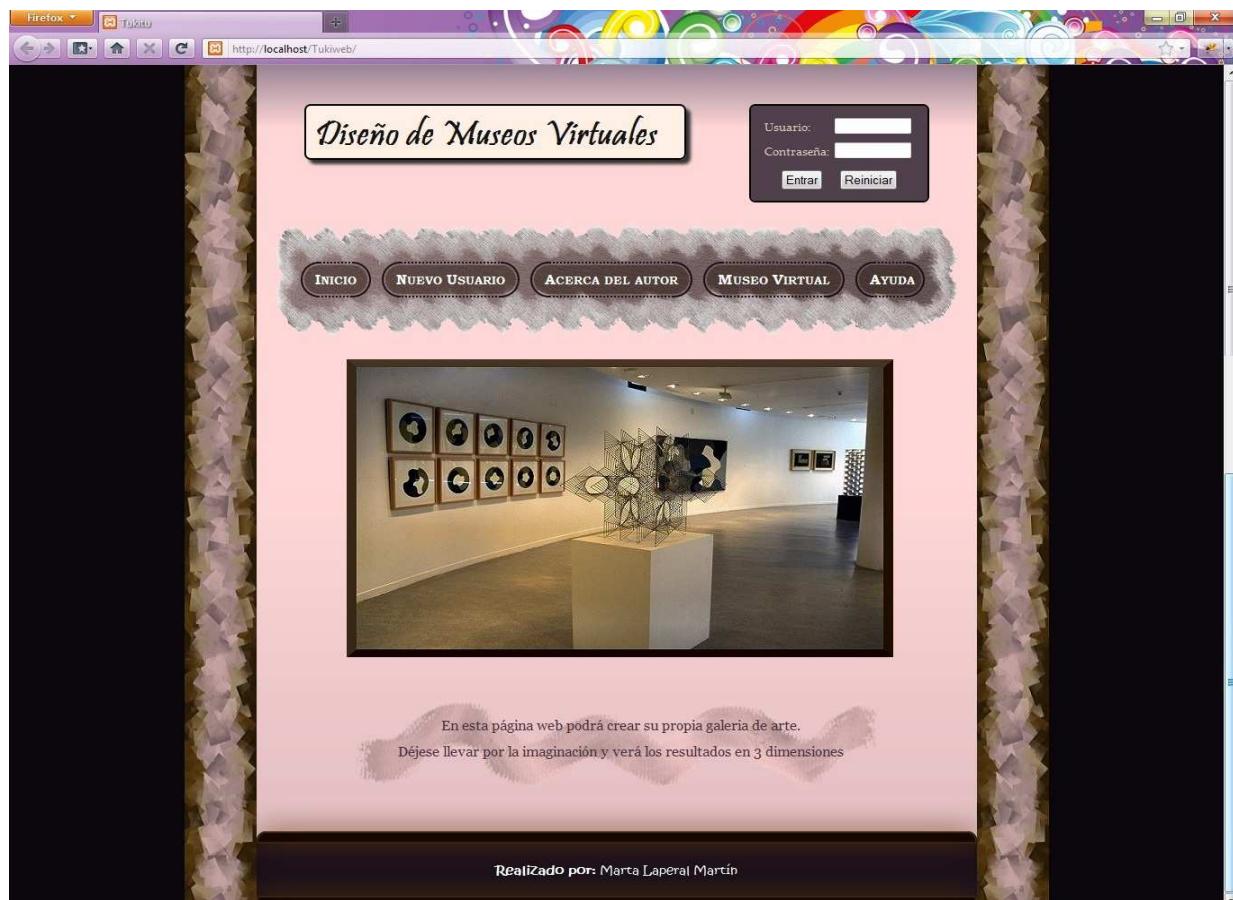


Figura 7. Página de Inicio aplicación web.

Para generar este tipo de página se ha creado una estructura con **HTML 5** compuesta por:

- **Una cabecera (Head):** Contiene entre otros el título de la página, la codificación hipertexto y las inclusiones de las librerías necesarias para trabajar.

```
<head>
    <meta http-equiv="Content-Type" content="text/HTML; charset=UTF-8">
    <title>Tukitu</title>
    <link href="CSS/estilo.CSS" type="text/CSS" rel="stylesheet" media="screen">
    <script src="js/rejilla.js" type="text/JavaScript"></script>
    <script src="js/JQuery.js" type="text/JavaScript"></script>
    ...
</head>
```

- **Un cuerpo (Body):** Que contendrá el encabezado (formado por un título y la sección de identificación); un navegador con el menú y sus distintas secciones; además de un artículo con distintas secciones que irá variando dinámica según se acceda a una sección u a otra; más un pie de página siempre presente al final.

```
<body>
<header>
    <h1>Dise&ntilde;o de Museos Virtuales</h1>
</header>
<div id="login">
    <?PHP include('acceso/intro.PHP'); ?>
</div>
<nav><div class="menu" id="menu2">
    <ol class="seccion">
        <li class="inicio">
            <a class="explicacion" id="p0" title="Volver al Inicio">Inicio</a>
        </li>
        <li class="logeo">
            <a class="explicacion" id="p1" title="Registrarse">Nuevo Usuario</a>
        </li>
    </ol>
</div>
</nav>
```

```

...etc.

</div></nav>

<div id="cuerpo" name="cuerpo">      ... </div>

<footer>

    <div class="subfooter" id="up"></div>

    <p><b>Realizado por:</b> Marta Laperal Mart&iacute;n</p>

    <div class="subfooter" id="down"></div>

</footer>

</body>

```

A parte será necesario gestionar el contenido del cuerpo (refiriéndome al **div**) para que el resto de secciones permanezcan intactas mientras que éste cambie su aspecto de forma dinámica. Esto es posible con la librería **JQuery** de JavaScript, pongamos un ejemplo:

```

<script type="text/JavaScript">

    $(document).ready(
        function(){
            $("#p0").click(
                function(evento){
                    evento.preventDefault();
                    $("#cuerpo").load("index/inicio.PHP");
                });
        }
        ...
    );

```

Además es posible incluir trozos de código escritos en PHP mediante PHP mismo, con los cuales poder gestionar los datos que se transmiten y con los que trabaja la aplicación:

```

<div id="cuerpo" name="cuerpo">

    <?PHP
        if (!isset($_SESSION['exito']))
            $_SESSION['exito']=0; // se inicializa

```

```

...
$exito = $_SESSION['exito'];

...
if ( $exito != 0 )
    include('registro/datos.PHP'); ... ?>
</div>

```

Por último, mediante JavaScript podremos generar avisos que informen al usuario del éxito o fallo de las operaciones con el sistema; como muestra la Figura 8, en la que un usuario no registrado ha intentado acceder a la zona privada de la página web:



Figura 8. Error acceso a la sección “Museo Virtual”.

BASE DE DATOS

A la hora de construir la base de datos, a nivel de código usaremos **PHP** para realizar consultas a la base de datos con secuencias **MySQL** que permitan insertar, actualizar, borrar, etc. los datos con los que trabaja la aplicación. Vemos un ejemplo de consulta:

```

$cnx = conecta();
$prop = $_SESSION['nombre'];

```

```
$borra = "delete from imagenes where propietario='".$prop. "'";  
$borra = mySQL_query($borra);
```

Por otro lado podemos mostramos mediante capturas cómo se ha estructurado la base de datos:

TABLA CLIENTE

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<u>id_user</u>	int(11)			No	0
usuario	varchar(50)	latin1_swedish_ci		Sí	NULL
pass	varchar(50)	latin1_swedish_ci		Sí	NULL
nombre	varchar(50)	latin1_swedish_ci		Sí	NULL
apellidos	varchar(50)	latin1_swedish_ci		Sí	NULL
correo	varchar(50)	latin1_swedish_ci		Sí	NULL
sexo	varchar(50)	latin1_swedish_ci		Sí	NULL
telefono	int(11)			Sí	NULL
empresa	varchar(50)	latin1_swedish_ci		Sí	NULL
direccion	varchar(50)	latin1_swedish_ci		Sí	NULL

En la que podemos observar todos los campos que contendrán los datos introducidos por el cliente a la hora de registrarse.

TABLA MUSEO

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<u>id_museo</u>	int(11)			No	0
casilla	int(11)			Sí	NULL
objeto	int(11)			Sí	NULL
propietario	varchar(50)	latin1_swedish_ci		Sí	NULL

Contiene los datos relativos a la configuración del museo diseñado en la rejilla de dos dimensiones, se identificará por el nombre de usuario y contendrá un listado con todas las casillas junto al objeto insertado en ellas para toda la cuadrilla.

TABLA IMÁGENES

Campo	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado
<u>id_ima</u>	int(11)			No	0
cuadro	int(11)			Sí	NULL
ruta	varchar(50)	latin1_swedish_ci		Sí	NULL
pos_x	int(11)			Sí	NULL
pos_y	int(11)			Sí	NULL
autor	varchar(50)	latin1_swedish_ci		Sí	NULL
titulo	varchar(50)	latin1_swedish_ci		Sí	NULL
propietario	varchar(50)	latin1_swedish_ci		Sí	NULL

Contiene el conjunto de imágenes correspondiente a las subidas al servidor y almacenadas en una carpeta por usuario, esta tabla tendrá: el número de cuadro que representa a la pintura en la rejilla; la ruta donde se haya la imagen en el directorio; la posición en la cuadrilla, el autor y título de la obra; y el nombre del cliente.

ANIMACIÓN 2D Y 3D

Como bien se ha indicado en el apartado “Fundamentos teóricos” pág. 28, las bases para desarrollar esta sección son la etiqueta **Canvas** de HTML, **JavaScript** y **WebGL**.

Para poder usarlo Canvas tan sólo es necesario incluir en la estructura de la página web la etiqueta:

```
<canvas identificador="mi dibujo" width="100" height="200"></canvas>
```

A partir del identificador podrá JavaScript trabajar con él usando WebGL. Por otro lado “width” y “height” indican el ancho y alto respectivamente que se le quiere dar al lienzo.

La diferencia principal para generar gráficos en dos o en tres dimensiones está a la hora de definir el contexto del lienzo, donde al inicializarlo se le indicará si vamos a pintar en 2D o en 3D. Hecho esto ya se usarán funciones determinadas en cada caso propias de WebGL.

Contexto para el 2D: canvas.getContext("2d");

Contexto para el 3D: canvas.getContext("WebGL");

Para poder trabajar con esta herramienta es necesario dividir su definición en dos partes:

1. Por un lado en la página web se ha ordenar que se cargue la inicialización de todo el material necesario, con la función:

```
Window.onload = WebGLStart();
```

El sistema buscará en las librerías el archivo JavaScript que contenga toda la funcionalidad de WebGL y cargará el contexto requerido. Es un requisito también imprescindible incluir al principio de nuestra página las librerías con las que funciona WebGL, que son:

```
<script type="text/JavaScript" src="GLMatrix-0.9.5.min.js"></script>
<script type="text/JavaScript" src="WebGL-utils.js"></script>
```

Por último se deben definir los **shaders**, que son los encargados de tratar los datos generados mediante WebGL que se quieran pintar, definiendo el estado de la escena y almacenando el resultado a pintar en unas variables llamadas “**varying variables**”, que especificarán propiedades como el color, la luminosidad, la transparencia, etc.

2. Por otro lado, se ha de crear un archivo de tipo JavaScript en el cual definiremos todas las funciones con las que se carga el lienzo; se inicializan y rellenan los shaders con el resultado final; se crean las **figuras 2D** y **3D**; se define la animación como la interacción del **ratón** y del **teclado** con el usuario; se aplican efectos sobre las figuras como la **traslación**, **rotación** y **escalado**; etc.

Además para cada objeto se ha de definir la posición de su vértices en coordenadas (x,y,z), si se rellena con **color** se definirán en 4 variables RGBA que establecerán el nivel de rojo, verde, azul y transparencia para cada vértice, ya que la forma de dibujar en WebGL es interpolar linealmente entre los vértices para colorear la figura.

Si bien se desea llenar con una **textura**, se deberán definir las coordenadas (x,y) de escalado por cada vértice y crear además un manejador donde definir y cargar una imagen del directorio en un formato dado, para asociarla con esa textura.

Estas son nociones básicas para poder trabajar con WebGL, no se ha explicado más a fondo la sintaxis por querer destacar sólo lo importante a conocer.

Tras generar estas animaciones tenemos los siguientes resultados (Figuras 9 y 10):



Figura 9. Animación 2D. Ejemplo de un diseño del museo realizado por un cliente.

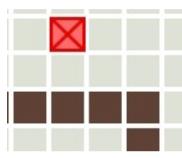
En esta figura podemos ver cómo el usuario puede insertar objetos en cada una de las casillas de la rejilla, haciendo clic en los botones de la barra lateral izquierda, donde cada uno representa a un objeto con un color asociado (para identificarlo en la cuadrilla): puerta, ventana, pared, pintura y suelo. Este último objeto tiene la funcionalidad de borrar, de modo que una vez activado el botón, si se pulsa sobre una casilla con otro objeto existente, lo eliminará colocando una baldosa en su lugar.



Además a la hora de pintar nos aparecerán los siguientes efectos junto al ratón:



Si el cuadro aparece en azul, significa que podemos insertar el objeto elegido en esa casilla.



Si el cuadro aparece en rojo con una cruz, significa que estamos infrin-
giendo alguna de las reGLas anteriormente mencionadas respecto a ese
objeto en concreto.

Vemos a continuación la zona 3D con una parte de la visualización del museo virtual:



Figura 10. Animación 3D. Ejemplo de la visualización museo diseñado tras navegar por él.

Para terminar, en la sección con la animación en tres dimensiones, se incluye generando código HTML 5 mediante JavaScript, todo el listado de imágenes almacenadas a partir del objeto **JSON**:

```
var museo = {    "suelo":{  
        "texturas":[ 0.0, 1.0, 0.0, 0.0, 1.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0, 0.0 ]  
        "vertices": [ -x/2, 0.0, -z/2, -x/2, 0.0, alto, ancho, 0.0, alto,  
                    -x/2, 0.0, -z/2, ancho, 0.0, -z/2, ancho, 0.0, alto],  
        "numfilas": 6  
    }, ... };
```

A partir de este vector con formato JSON cargamos mediante JavaScript el listado de pinturas a especificar, ofreciendo resultados como la captura de la Figura 11:

The screenshot shows a user interface for viewing artworks. On the left, there is a small image of a painting depicting a garden path leading to a building, identified as 'CUADRO N° 1'. To the right of the image, there is descriptive text: 'Use las teclas W / A / S / D para moverse hacia delante, la izquierda, atrás y la derecha respectivamente.' Below this, a button labeled 'ACERCA DE LAS OBRAS' is visible. Further down, there is a list of details about the painting: '• Autor de la obra (*): Jordi Pagans' and '• Titulo (*): La Alhambra'.

Figura 11. Parte del listado de obras subidas por el usuario junto a su información.

COMENTARIO FINAL

Tras terminar mi proyecto y una vez realizadas todas las etapas que lo componen, se puede decir que he conseguido casi todos los objetivos que fueron planteados al inicio del proyecto. He procurado cumplir todos los requisitos y servicios que debía ofrecer la aplicación.

Mis bases a seguir han sido la simplicidad, eficacia, rapidez y robustez. Buscaba desarrollar una aplicación fácil de usar y lo más intuitiva posible, que trabajara con los nuevos y antiguos lenguajes que imperan en el mundo de la programación web.

Ante todo mi objetivo principal era realizar una investigación de las tecnologías principales de programación web, por lo que me he centrado en aprender con detenimiento los nuevos fuertes de las aplicaciones web, como son HTML5, CSS3, JQuery y WebGL.

Aquellas metas no alcanzadas que podrán ser posibles ampliaciones del software son las siguientes:

- Generación de varios museos para un mismo usuario, tanto en la zona 2D como en la zona 3D.
- Posibilidad de rotar la cámara dentro del mundo 3D (no obstante esta característica estaba incluida en la aplicación, mas la eliminé al no verla realmente necesaria).
- Creación de un archivo donde guardar el código generador de la visualización 3D de un modelo concreto de museo diseñado.
- Adición de efectos con distintos tipos de iluminación y uso de la transparencia en las texturas.
- Permitir una carga ilimitada de imágenes al servidor, actualmente el límite son 20 imágenes por subida, ya que es una restricción no modificable propia del servidor PHP.

BIBLIOGRAFÍA

HTML5:

- <http://www.desarrolloweb.com/articulos/que-es-HTML5.HTML>
- <http://ayudawordpress.com/HTML5-todas-las-nuevas-etiquetas/>

CSS3:

- <http://www.desarrolloweb.com/manuales/CSS3.HTML>
- <http://gradients.GLrzad.com/>

JQUERY:

- <http://www.maestrosdelweb.com/editorial/JavaScript-facil-y-rapido-con-JQuery/>
- <http://theproc.es/2010/2/15/13527/eventos-en-JQuery--tutorial-basico-para-comenzar-a-trabajar-con-JQuery--parte-2-de-4->

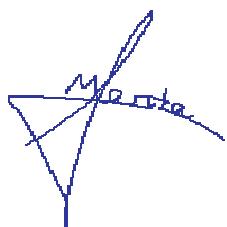
WEBGL:

- <https://sites.googLe.com/site/desarrolloenWebGL/bienvenida>
- <http://learningWebGL.com/blog/>

APUNTES DE LA AUTORA

Este proyecto ha supuesto para mí una experiencia muy instructiva y enriquecedora, que ha servido para hacerme ver todo el trabajo que se realiza a lo largo del estudio de una carrera como es la de informática. Gracias al proyecto he aprendido mucho acerca de los nuevos lenguajes dedicados a la programación web y al mundo de la animación en los navegadores.

En definitiva ha sido un reto personal muy interesante y productivo, que ha supuesto un largo trabajo de investigación y desarrollo. No obstante ha sido divertido y agradezco a mi tutor Daniel Riera Terrén y a la Escola d'Enginyeria haberme dado la oportunidad de llevármelo a cabo.



Firmado: Marta Laperal Martín.