UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6

Título: Aplicación Web para la gestión de componentes gráficos y sonoros del proyecto UCI-CIMEQ.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Frangers Luis Mercader Guevara

Tutor: Lic. Luis Gabriel Viciedo Caraballoso

Asesor: Ing. Roberto Elías Pérez Ozete

La Habana, Cuba. 2014

"Año 56 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de la Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Frangers Luis Mercader Guevara	Lic. Luis Gabriel Viciedo Caraballoso
_____	_____
Firma del Autor	Firma del Tutor

Agradecimientos

Agradecer infinitamente a mis padres por ser los mejores que jamás pude haber imaginado, a ellos que se han sacrificado día a día por mi sin escatimar esfuerzos. A ellos por brindarme todo su amor y cariño incondicionales.

A toda mi familia, mi hermano, tías, tíos y primos a los que quiero mucho.

A todos los compañeros de aula con los que he compartido estos maravillosos cinco años de universidad.

A todos los que de una forma u otra me han ayudado a llegar donde esto hoy.

Dedicatoria

Dedico este trabajo especialmente a mis padres a ellos que debo mi razón de ser y mi existir.

Resumen

El trabajo de diploma muestra el proceso de diseño e implementación de una aplicación Web para gestionar componentes de software, específicamente gráficos y sonoros que son generados por el equipo de desarrollo del proyecto UCI-CIMEQ, durante la construcción de un Ambiente Virtual de Aprendizaje Interactivo. Actualmente estos componentes de software son administrados de forma manual e ineficiente, lo que provoca pérdidas de tiempo en el proceso productivo. En la solución informática obtenida, se utilizó la metodología ágil de desarrollo XP (*Extreme Programming*), el *framework* de desarrollo Web Symfony 2.4, el lenguaje de programación PHP 5.3 y los estándares de programación Web HTML 5, CSS 3, WebGL 1.0.1 y JavaScript. Como resultado la solución obtenida organiza y automatiza los procesos de guardado, conservación, visualización, búsqueda y descarga de los componentes gráficos y sonoros que son creados por el equipo de desarrollo del proyecto UCI-CIMEQ del Centro de Entornos Interactivos 3D (VERTEX) de la Facultad 5 en la Universidad de las Ciencias Informáticas.

Índice

Introducción.

En las últimas décadas, la producción de software ha crecido constantemente gracias a la creciente demanda y desarrollo de los sistemas informáticos. A medida que la importancia del software ha crecido, la comunidad del software ha intentado de manera continua desarrollar tecnologías que hagan más fácil, rápida y menos costosa la construcción y el mantenimiento de aplicaciones informáticas de alta calidad.

La Universidad de las Ciencias Informáticas (UCI), se considera una universidad productiva, cuya misión es formar profesionales comprometidos con su Patria y altamente calificados en la rama de la Informática. Igualmente debe producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación y servir de soporte a la industria cubana de la Informática (1). La producción de esta entidad académica, se concentra en el desarrollo de proyectos en más de treinta polos productivos y se destacan resultados en las esferas de Salud, Educación, Software Libre, Teleformación, Sistemas Legales, Realidad Virtual, Automatización, Bioinformática, Procesamiento de Imágenes y Señales, entre otras.

El Centro de Rehabilitación de la Cara y Prótesis Bucomaxilofacial, perteneciente al Hospital CIMEQ, desarrolla tecnologías asociadas con las prótesis bucomaxilofaciales. Los estudiantes del cuarto año de las carreras de Estomatología y Licenciatura en Tecnología de la Salud, deben entrenarse en un grupo de tecnologías médicas de la especialidad y desarrollar las habilidades y competencias exigidas por el programa de estudios, todo lo cual está actualmente con dificultades en su ejecución, por el alto costo de las tecnologías médicas necesarias.

Esto propició un acercamiento entre la dirección de este centro de investigación del CIMEQ y un grupo de estudiantes y profesores de la Facultad 5 de la UCI, en la búsqueda de una aplicación informática para el adiestramiento de los estudiantes. Por este motivo se crea un nuevo proyecto de investigación y desarrollo, nombrado Proyecto UCI-CIMEQ, perteneciente al Centro de Entornos Interactivos 3D (VERTEX). Este proyecto se encarga de la implementación de Ambientes Virtuales de Aprendizaje para el entrenamiento de personal profesional o de empresas. Para dicha tarea se crean, objetos, escenarios, animaciones tridimensionales y sonidos que integran estos ambientes. Debido a que los integrantes de los equipos de desarrollo cambian constantemente al graduarse o abandonar la universidad, en ocasiones se pierden estos componentes gráficos y sonoros desarrollados.

Actualmente el proyecto UCI-CIMEQ cuenta con gran cantidad de objetos, escenarios, animaciones y efectos de sonido. Debido a las características del proyecto y las metodologías empleadas, es necesario que los miembros del proyecto ejecuten constantemente acciones de filtrado, búsqueda, selección y visualización de los componentes gráficos y sonoros almacenados. Y todo esto sin tener que recurrir a programas específicos como Blender, 3DMax o reproductores de audio solo para ver o reproducir respectivamente objetos 3D o sonidos. Fue detectado que estas acciones son realizadas en el proyecto productivo de forma ineficiente, al estar ubicados los componentes de software en diferentes computadoras, se consume tiempo en la búsqueda, lo que ralentiza el proceso productivo.

Además es necesario establecer como parte del proceso productivo, un grupo de metadatos en cada uno de los componentes de software como son: autores, fechas de realización, formatos, herramientas empleadas, cantidad de polígonos, entre otros, que por simple observación del componente no es posible obtener. En ocasiones se recurren a componentes descargados de diferentes fuentes, como Internet, por lo que resulta significativo asignar sus metadatos correspondientes, para evaluar si pueden ser usados posteriormente.

Desde el punto de vista organizativo del proyecto, es necesario armonizar las acciones de gestión de los componentes gráficos y sonoros que se encuentran distribuidos en diferentes máquinas y carpetas, y al mismo tiempo, visualizarlos con sus metadatos correspondientes. En recientes trabajos recopilatorios, como los de Marfisi-Schottman, se indican que estas situaciones son frecuentes en equipos de desarrollo multidisciplinarios dedicados a la realización de simulaciones y videojuegos con fines de aprendizaje, independientemente de las metodologías de desarrollo de software empleadas y que si no son tomadas acciones al respecto, se genera disminución de la eficiencia del trabajo y en los costos de producción en estos tipos de proyectos.

Dada la situación problemática anteriormente expuesta se define como **problema de la investigación**: ¿Cómo gestionar los componentes gráficos y sonoros generados durante el proceso de desarrollo del proyecto UCI-QIMEQ?

Como **objetivo general** se define: Desarrollar una aplicación web que permita gestionar los componentes gráficos y sonoros generados durante el proceso de desarrollo del proyecto UCI-CIMEQ.

El problema planteado está contenido en el **objeto de estudio**: Gestión de componentes de software en el proceso de desarrollo de aplicaciones informáticas enmarcado en el **campo de acción**: Gestión de componentes gráficos y sonoros en el desarrollo de ambientes virtuales de aprendizaje.

Para guiar la investigación se definen las siguientes preguntas de investigación:

1. ¿Cuáles son los fundamentos teóricos que rigen el proceso de desarrollo de aplicaciones web para la gestión de componentes gráficos y sonoros empleados durante el desarrollo de ambientes virtuales de aprendizaje?

2. ¿Cuáles son las características y capacidades que debe cumplir la aplicación web para permitir la gestión de los componentes gráficos y sonoros generados durante el proceso de desarrollo del proyecto UCI-CIMEQ?

3. ¿La aplicación web desarrollada permite la gestión de los componentes gráficos y sonoros generados durante el proceso de desarrollo del proyecto UCI-CIMEQ?

Para cumplir con el objetivo propuesto se precisan las siguientes **tareas de investigación**:

1. Análisis de documentación especializada sobre las metodologías, principios y aplicaciones relacionadas con la gestión de componentes gráficos y sonoros, para comprender las responsabilidades con las que debe cumplir la aplicación.
2. Fundamentación de las tecnologías y herramientas a emplear para la implementación de la aplicación Web.
3. Especificación de los requisitos funcionales y no funcionales del sistema, para su implementación en la solución propuesta.
4. Diseño e implementación de la solución propuesta para obtener una aplicación funcional.
5. Realización de pruebas para la comprobar el correcto funcionamiento de la aplicación obtenida y la aceptación del usuario.

Métodos de investigación:

Para llevar a cabo el proceso de investigación se utilizaron varios métodos de investigación, los cuales son descritos a continuación:

Métodos teóricos:

Analítico-sintético: Para el análisis y descomposición del problema a resolver en tareas más concretas para guiar el proceso de investigación y desarrollo. Se realizó un estudio de la bibliografía y herramientas relacionadas a la gestión de componentes de software.

Modelación: Método utilizado que será para representar mediante diagramas las diferentes vistas que le dan solución al problema.

Método empírico:

Análisis estático: Se analizó la estructura y el funcionamiento de diferentes sistemas Web de gestión de componentes gráficos y sonoros, con el objetivo de recopilar información relevante a tener en cuenta para el desarrollo la aplicación.

Técnicas de obtención de información:

Tormenta de Ideas: Se consultó a diseñadores y especialistas del equipo de desarrollo que gestionan componentes gráficos y sonoros en los proyectos productivos, con el fin de conocer sobre de las funcionalidades y prestaciones a brindar por la aplicación para satisfacer las necesidades de estos.

El contenido de este trabajo de diploma está desglosado en tres capítulos, estructurados de la siguiente manera:

Capítulo 1 – Fundamentación teórica sobre la gestión de componentes gráficos y sonoros.

En este capítulo, se muestran los principales conceptos, principios y características fundamentales referentes a la gestión de componentes de software. Se analizan sistemas similares existentes asociadas al campo de acción y se muestran los resultados de la fundamentación y selección de las herramientas necesarias para implementar la aplicación.

Capítulo 2 – Características y diseño de la solución.

En este capítulo realiza la descripción de la solución propuesta, se identifican los requisitos funcionales y no funcionales de la aplicación a partir de las Historias de Usuarios (HU) definidas por el cliente. Además se definen las tarjetas CRC, la arquitectura y los patrones de diseño empleados. Por último se describe el plan de iteraciones en el cual se precisa el tiempo necesario para el desarrollo de la aplicación y se especifica cada una de estas iteraciones planificadas para la fase de implementación.

Capítulo 3 – Implementación y prueba de la solución.

En este capítulo se aborda las fases de implementación y prueba de la aplicación donde se puntualizan las tareas de ingeniería realizadas por cada HU y por ultimo las pruebas unitarias y de aceptación realizadas.

Capítulo 1: Fundamentación Teórica sobre la gestión de componentes gráficos y sonoros.

1.1 Gestores de Contenidos

Los contenidos digitales en la actualidad están presentes de forma creciente en las diferentes organizaciones. Estos pueden estar relacionados entre sí y encontrarse distribuidos en diferentes ubicaciones. Además, la descripción relacionada con ellos es cada vez más importante gestionarla, pues en muchos casos se necesita que no sea estática sino dinámica, de forma tal que se puedan agregar, modificar y eliminar elementos a la descripción. También la manera en que se van a mostrar los contenidos es una información que se puede necesitar, así como el registro de las versiones que han sido cambiadas.

Esta complejidad de los contenidos ha traído como consecuencia, que actualmente en el mundo exista una definición denominada: Objetos Digitales, la que se refieren a productos “de origen digital” en formato electrónico. Estos pueden ser textos, imágenes, videos, materiales gráficos, páginas web o programas informáticos, entre otros; dentro de los muchos formatos posibles en la diversidad creciente (2).

Para lograr la gestión de contenidos y objetos digitales se fueron desarrollando paulatinamente desde la década del 90, un grupo de herramientas informáticas, las cuales se han nombrado Sistemas de Gestión de Contenidos o Gestores de Contenidos (3). Bajo el término Gestión de Contenidos se pueden encontrar disímiles sistemas, que muestran diferentes orientaciones y prestaciones, enfocadas a diferentes objetivos y grupos de usuarios.

1.2 Definición de Gestión de Contenido

La definición de gestión de contenidos puede ser una tarea algo difícil debido a que el término se utiliza en ámbitos diversos y con puntos de vista diferentes.

Algunos conceptos en distintos contextos son:

- En el ámbito de la Informática se refiere a un sistema usado para crear, editar, gestionar y publicar contenido digital en diversos formatos. El gestor de contenidos genera páginas dinámicas

interactuando con el servidor para generar la página web bajo petición del usuario, con el formato predefinido y el contenido extraído de la base de datos del servidor (4).

- Por otra parte, la gestión de contenidos también puede equipararse a un planteamiento más amplio enfocado a la gestión global de los recursos de información de una institución o empresa mediante tecnologías web (Internet e Intranet). Este enfoque brinda a la tecnología un papel facilitador y el mayor peso recae en los aspectos relacionados con la identificación de recursos de información internos y externos, su valoración, gestión y tratamiento eficiente (5).

En los tres casos se pueden encontrar puntos comunes. El más importante de ellos, la necesidad de utilizar tecnologías de la información y sistemas informáticos para el almacenamiento, la gestión y distribución de la información.

1.3 Procesos de la Gestión de Componentes de Software

Profundizando específicamente en la Gestión de Componentes de Software no se encontró una definición globalizada de término, aunque la mayoría de los trabajos referentes al tema plantean la necesidad de utilizar tecnologías de la información y sistemas informáticos para el almacenamiento, la gestión y distribución de las componentes. Los trabajos de gestión de componentes de software coinciden en señalar como procesos elementales, la creación, la gestión y la publicación (6).

Según el modelo planteado por Dayni Pérez Hernández y Martha Dunia Delgado Dapena (6), un gestor de componentes tienes tres procesos fundamentales como se muestra en la Figura 1.

Figura 1: Procesos de la gestión de componentes de software.

Proceso de Captura: En este se pueden incorporar recursos de diversas fuentes a los componentes, así como agregarle nuevos elementos a los recursos propiamente creados. Los metadatos que describen a los componentes son otro elemento importante a utilizar, pues sentará las bases para poder buscarlos y reutilizarlos. Este proceso debe permitir la gestión flexible de los metadatos, con el objetivo de que se puedan agregar, eliminar o modificar nuevos campos a los metadatos.

Proceso de Administración: Se ocupa de realizar los flujos de trabajos que puedan ejecutarse, gestionar las versiones, los roles y permisos para un componente. Un papel importante que tiene el proceso de Administración es proveer los métodos, mecanismos y herramientas para el almacenamiento de los componentes de software, incluyendo sus metadatos y recursos asociados.

Proceso de Entrega: Este proceso se encarga de la presentación y recuperación de los Objetos Digitales y sus recursos asociados; de manera que estos pueden ser presentados en diferentes vistas según los permisos y roles definidos para los usuarios finales de la aplicación. Este proceso posibilita además la realización de búsquedas y reportes sobre los contenidos. El componente social es todo el conocimiento agregado por los usuarios alrededor del Objeto Digital.

1.4 Sistemas Web de gestión de componentes gráficos y sonoros

Enfocándonos más en la gestión de componentes gráficos y sonoros se analizaron varias aplicaciones web con el objetivo de buscar sistemas similares que pueden dar solución al problema planteado o brindar elementos e información de referencia para reutilizarse en la solución propuesta. En la actualidad existen un gran número de sitios Web que se enfocan total o parcialmente a la gestión (captura, administración y entrega) de modelos 3D y de audio, algunos con fines comerciales, de apoyo a comunidades o proyectos, entre otros fines. A continuación se muestran algunos de los sistemas analizados.

1.4.1 TurboSquid

Este es una aplicación Web profesional de ámbito internacional dedicada a la gestión de modelos 3D. Presenta características de mercado virtual, donde se brinda a los usuarios la posibilidad de publicar sus productos y ponerlos a la venta bajo los términos establecidos por el sitio, de manera que los interesados puedan buscar los modelos según sus intereses y descargarlos luego de pagar su precio. Provee al usuario con la posibilidad de realizar búsquedas mediante una palabra clave que se indica a través de un campo de tipo búsqueda en la parte superior del sitio, además cuenta con una barra lateral en la que

brinda un gran número de opciones de filtrado dadas por las características de los modelos. Al realizar búsquedas de modelos con TurboSquid, este brinda una vista en miniatura del modelo tridimensional, el precio, los formatos en que está disponible y un vínculo a una página donde se muestran más información sobre el modelo encontrado. Además brinda la opción de poder visualizar de forma tridimensional algunos de los modelos en dependencia de su formato.

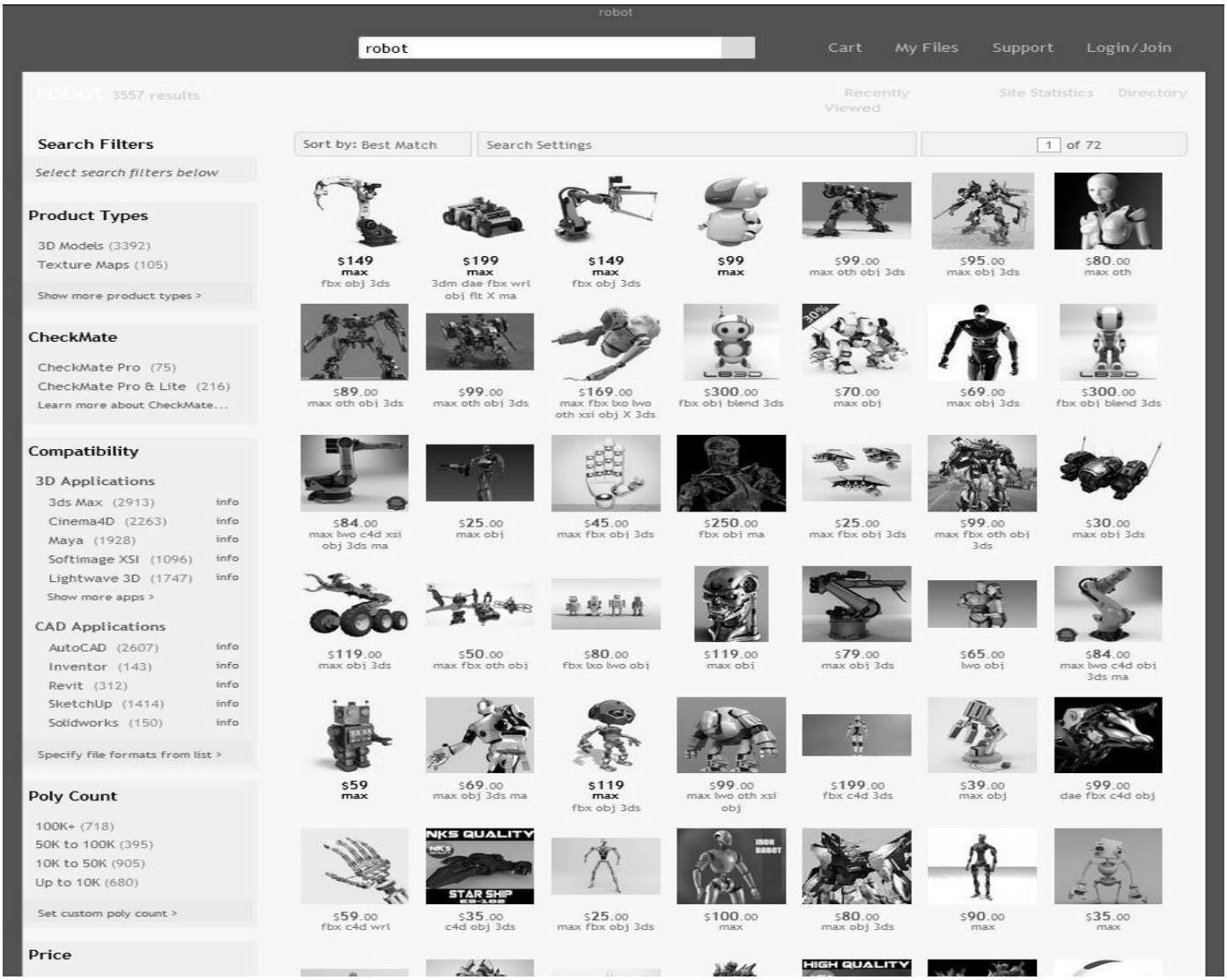


Figura 2: Sitio TurboSquid (7).

1.4.2 SoundSnap

Esta aplicación Web representa una opción esencial para un amplio grupo de usuarios como: artistas de la música, creadores de videojuegos, productores de películas y entre otros. Actualmente cuenta con alrededor de 160 000 efectos de sonido y *loops* (sonidos cíclicos), que se encuentran distribuidos en distintas categorías y subcategorías. La búsqueda sobre su contenido se realiza filtrando mediante las categorías definidas y además se puede indicar una palabra clave a través del campo de tipo búsqueda en la parte superior derecha del sitio para agilizar el proceso. Durante el proceso se muestran los resultados dando al usuario la posibilidad de reproducir los audios y se muestra información referente a estos, como su formato, duración, titulo, velocidad de reproducción y ofrece un enlace para su descarga.

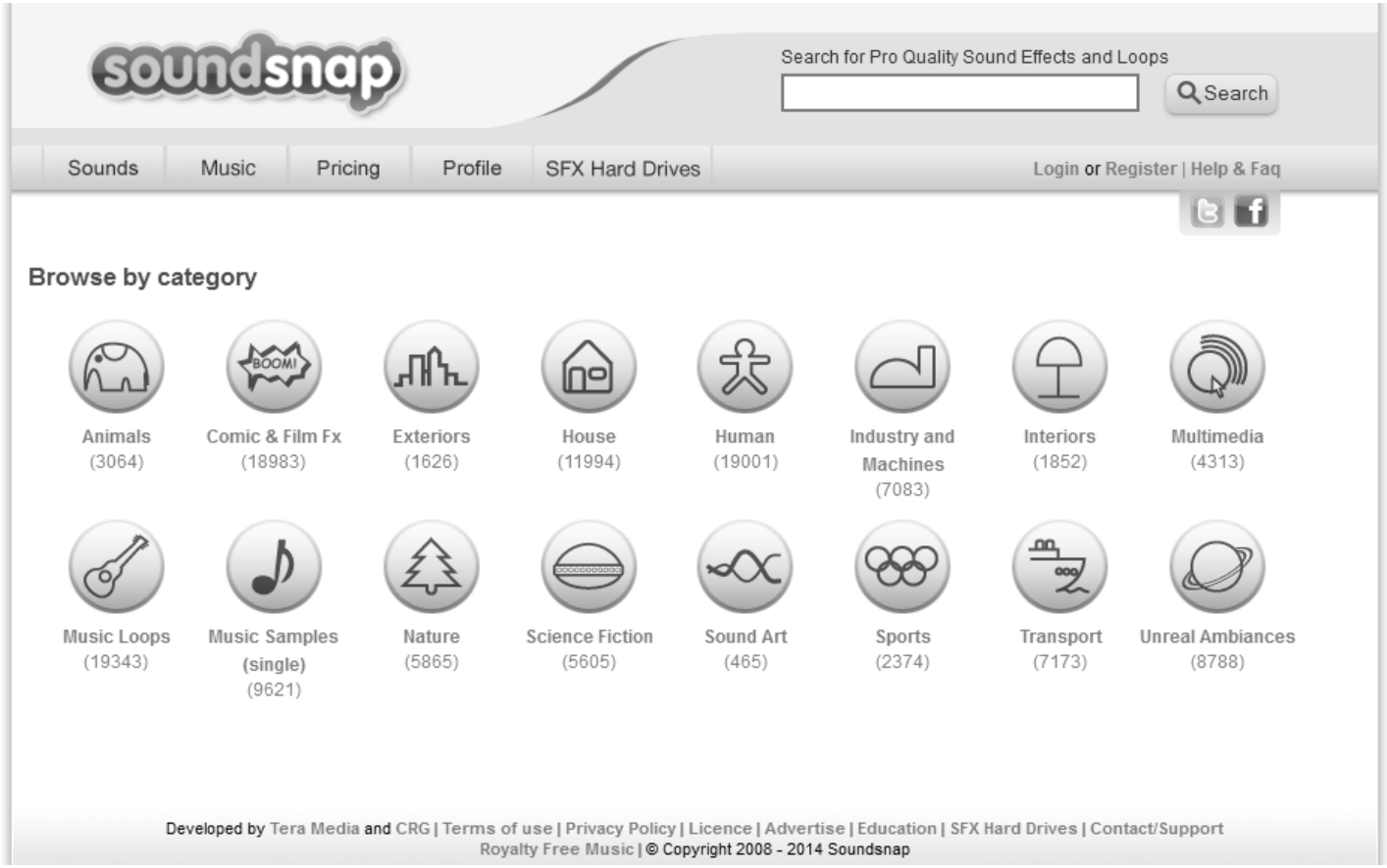


Figura 3: Sitio SoundSnap (8).

Consideraciones parciales

A partir del análisis anterior referente al objeto de estudio se concluye que la aplicación de forma general, dada las necesidades del cliente, debe abarcar diferentes características y funcionalidades que poseen los diferentes sistemas analizados. Entre ellas la captura y almacenamiento organizado de los componentes y sus metadatos correspondientes, además la aplicación debe permitir la publicación de los componentes de manera que los usuarios puedan acceder fácilmente a ellos, realizar búsquedas y consigan consultar información, reproducir o visualizar los componentes gráficos y sonoros.

1.5 Herramientas y tecnologías a utilizar

Guiado por las pautas definidas en el centro se realizó un análisis de las herramientas y tecnologías necesarias para el desarrollo de la solución.

1.5.1 Lenguajes del lado del cliente

Hoy en día el desarrollo de la tecnología y el Internet ha posibilitado el surgimiento de una gran variedad de lenguajes, que permiten a los usuarios finales lograr una interfaz amigable para la visualización de contenido en la web. A continuación se muestran algunos de los más importantes y utilizados, para la confección de una interfaz web.

HTML

HTML, siglas de *HyperText Markup Language* (Lenguaje de Marcas de Hipertexto), es un lenguaje de marcado para la elaboración de páginas Web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto e imágenes. El lenguaje HTML basa su filosofía de desarrollo en la referenciación. Para añadir un elemento externo a la página (imagen, video, *script*), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. En su última versión (HTML5) brinda un grupo de nuevas y poderosas herramientas y funcionalidades como los son sus etiquetas “<video>”, “<audio>”, “<svg>” y “<canvas>” entre otras que permiten la agregación de audio, videos, animaciones y componentes de interfaz complejos sin la necesidad de utilizar *plugins* (9).

CSS

Hojas de Estilo en Cascada (*Cascading Style Sheets*), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. Su última versión (CSS 3) ofrece nuevas opciones que hacen del diseño una tarea más fácil. Ejemplo de estas opciones son el uso de la propiedad "*font-face*" que posibilidad usar cualquier fuente de letra en HTML, nuevos selectores como "*odd" y "even*" para dar estilos a los elementos de una tabla, la propiedad "*opacity"* para indicar la transparencia de un elemento, entre otras muchas opciones de utilidad (9).

JavaScript

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas (10). Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos y pueden ser probados directamente en cualquier navegador sin necesidad de procesos intermedios (11).

Selección del lenguaje del lado del cliente

Dada las características propias de la aplicación se hace indispensable el uso de todos los lenguajes anteriormente analizados HTML, CSS y JavaScript para la construcción, estructuración y diseño de la parte visual de la aplicación.

1.5.2 Lenguajes del lado del servidor

Para implementar las funcionalidades del sistema, se hace necesario la selección y utilización de un lenguaje de programación del lado del servidor. El principal objetivo de estos lenguajes es permitir la combinación de código interpretable con HTML y que de este modo el servidor pueda traducirlo a programas ejecutables (12). Actualmente existen diferentes lenguajes orientados a esta tarea, a continuación se muestran dos de los utilizados por la comunidad internacional en esta área.

PHP

PHP es el acrónimo de *Hypertext Preprocessor*, es un lenguaje *Open Source* interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender (12). La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

Quizás la característica más potente y destacable de PHP es su soporte para una gran cantidad de bases de datos. Escribir un interfaz vía web para una base de datos es una tarea simple con PHP. Entre los gestores pueden destacarse:

- MySQL.
- Oracle.
- PostgreSQL.

PHP es multiplataforma el cual se puede usar en la mayoría de los sistemas operativos empleados en el mercado internacional como: Linux, Unix Microsoft Windows y Mac OS X entre otros. Además PHP soporta la mayoría de servidores web de hoy en día, incluyendo Apache, Microsoft Internet Information Server, Personal Web Server, Oreilly Website Pro server, Caudium, Xitami, OmniHTTPd (13). Posee módulos disponibles para la mayoría de los servidores y para aquellos otros que soporten el estándar CGI, PHP puede usarse como procesador CGI.

JSP

JSP es la abreviatura de *Java Server Page*, que consiste en una tecnología del lado del servidor y consiste en una extensión de la tecnología de *servlets* de Java que fue desarrollado por Sun. JSP tienen capacidad de *scripting* dinámico que trabaja en conjunto con código HTML, separando la lógica de la página a partir de los elementos estáticos para ayudar a hacer el código HTML más funcional (14). La tecnología *Java Server Pages* permite a los desarrolladores web y diseñadores desarrollar rápidamente y mantener fácilmente, páginas ricas en información, páginas Web dinámicas que aprovechan los sistemas empresariales existentes. Como parte de la familia de la tecnología Java, la tecnología JSP permite el rápido desarrollo de aplicaciones basadas en la Web que son independientes de la plataforma (15). La tecnología JSP separa la interfaz de usuario de la generación de contenidos, permitiendo a los diseñadores cambiar el diseño general de la página sin alterar el contenido dinámico subyacente (15).

Selección del lenguaje del lado del servidor

De los lenguajes anteriormente analizados se seleccionó PHP como lenguaje de programación del lado del servidor dada la sencillez de su código, la amplia documentación existente, la gran comunidad de desarrolladores con que cuenta a nivel mundial, además es *Open Source* y se ajusta a las necesidades del proyecto. Otras ventajas a destacar es que posee manejadores para bases de datos como MySQL y PostgreSQL. Es un lenguaje multiplataforma y orientado completamente al desarrollo de aplicaciones web dinámicas. Soporta el paradigma de Programación Orientada a Objetos.

1.5.8 Tecnologías de visualización 3D en la Web

La popularidad y el uso de representaciones tridimensionales han crecido en los últimos años, lo que abrió un camino con nuevas y amplias posibilidades para las aplicaciones Web. Por esta razón varias empresas e instituciones se han encaminado en la tarea de desarrollar tecnologías que faciliten la creación y visualización de entornos, objetos o imágenes 3D sobre la Web.

WebGl

WebGL es una librería para la creación de gráficos 3D, la cual permite a los navegadores Web modernos recrear escenas tridimensionales de forma eficiente, interactuando directamente con el GPU (Unidad de Procesamiento Gráfico) y sin la necesidad de recurrir a *plugins* para el navegador. Esta ha sido incorporada a HTML5 mediante la combinación entre JavaScript, OpenGL y la etiqueta "<*canvas*>*"*.

En comparación con otras tecnologías WebGL presenta varias ventajas como (37):

- **Programación en JavaScript:** El lenguaje JavaScript resulta bastante natural tanto para los programadores web como los navegadores web. Al trabajar sobre este lenguaje te permite acceder a todos los elementos de DOM de una página. Además esto permite a las aplicaciones desarrolladas con WebGL integrarse fácilmente con otras librerías como Jquery y otras tecnologías HTML5.
- **Gestión automática de memoria:** WebGL sigue las reglas definidas por JavaScript para manejar el ámbito de las variables, por lo que estas se eliminan automáticamente de la memoria cuando ya no son necesarias. Esto simplifica el proceso de programación al reducir el código necesario haciéndolo más claro y fácil de entender.

- **Rendimiento:** El rendimiento de las aplicaciones WebGL es comparable con aplicaciones autónomas con algunas excepciones. Esto es posible gracias a la capacidad que posee de acceder e interactuar con el hardware gráfico local donde se ejecuta.
- **Cero compilación:** Como está escrito sobre JavaScript su código no necesita compilarse antes de ejecutado en el navegador.

Alternativa3D

Es un motor 3D basado en la plataforma Adobe Flash que posibilita mostrar ambientes 3D, juegos, visitas virtuales o simplemente objetos en el navegador (38). Una de las mayores ventajas es que utiliza Flash, que se encuentran en la mayoría de los navegadores. Recientemente, exactamente en su versión 8 Alternativa3D brinda una serie de características impresionantes, entre ellas:

- **Alto rendimiento Prestación de GPU:** Permite la visualización de objetos sobre los 3 millones de polígonos.
- **Sistema de iluminación:** Brinda iluminación puntual, direccional y lugar de fuentes de luz. Cada objeto puede ser iluminado por seis fuentes de luz al mismo tiempo.
- **Materiales avanzados:** Permite el uso de iluminación dinámica completa con el mapa normal, el mapa especular, hoja brillante o más materiales ligeros.
- **Carácter de dibujo:** Potente sistema de mezcla de animación jerárquica. Modelos de los personajes de una complejidad sin límite de cantidad de hueso.

Su principal desventaja radica en que es un software propietario y se necesita de licencia para su utilización.

Selección de la tecnología de visualización 3D

Se escogió la tecnología WebGL para la implementar la funcionalidad de visualización 3D gracias a las facilidades que brinda como su integración con HTML5 y JavaScript fundamental para la interacción y manejo con los elementos del DOM. También por su total independencia al contrario de Alternativa3D que necesita para su desempeño la instalación de Adobe Flash en los navegadores. Y por último WebGL es software libre.

1.5.3 Framework de desarrollo

Un *framework* es un producto que implementa un grupo de procesos y rutinas estándares, que han sido probadas y no se necesitan volver a programar. Por lo que sirven para la programación avanzada de aplicaciones informáticas, evitando a los desarrolladores tener que implementar tareas básicas que son repetitivas y muy comunes en la implementación de estas aplicaciones (16).

Symfony 2

Symfony 2 es un *framework* PHP de tipo *full-stack* construido con varios componentes independientes creados por el proyecto Symfony. Su código y el de todos los componentes y librerías que incluye, se publican bajo la licencia MIT de software libre (17) (18). La documentación del proyecto también es libre e incluye varios libros y decenas de tutoriales específicos. Symfony permite acceder a una gran variedad de proyectos: el *framework* Symfony2 para crear aplicaciones Web complejas, el micro *framework* Silex para sitios web sencillos y los componentes Symfony para otras aplicaciones PHP. Según estadísticas del sitio GitHub (19), “Symfony es el proyecto PHP más activo”, lo que garantiza que nunca quedar atrapado en un proyecto sin actividad. Los componentes de Symfony son tan útiles y están tan probados, que proyectos tan gigantescos como Drupal 8 están construidos con ellos (20).

Entre las características más generales del *framework* se destacan:

- Es fácil de instalar y configurar en la mayoría de plataformas (y con la garantía de que funciona correctamente en los sistemas Windows y Unix estándares).
- Es independiente del sistema gestor de bases de datos.
- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- Posee una capa de abstracción de objetos que se encarga de mapear los datos relacionales a objetos mediante la librería Doctrine 2.
- Preparado para aplicación empresarial y adaptable a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- Código fácil de leer que incluye comentarios de phpDocumentor y que permite un mantenimiento muy sencillo.
- Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros.

Zend Framework

Zend Framework (ZF) es un *framework* de código abierto para desarrollar aplicaciones web y servicios web con PHP 5 (21). El hecho de ser patrocinado por la empresa Zend Technologies, a quien también se considera como una de las principales impulsoras de PHP, le otorga un mérito adicional aunque otras compañías como Google y Microsoft han contribuido también. ZF es una implementación que usa código completamente orientado a objetos. En la estructura de los componentes de ZF; cada componente está construido con un bajo acoplamiento de otros componentes. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado.

Entre las características que presenta ZF se encuentran:

- ZF ofrece una implementación del patrón arquitectónico MVC, una abstracción de base de datos y un componente de formularios que implementa la creación de formularios HTML.
- Otros componentes, como Zend_Auth y Zend_Acl, proveen autentificación de usuarios y autorización diferentes a las tiendas de certificados comunes.
- Posee componentes que implementan bibliotecas de cliente para acceder de forma sencilla a los servicios web más populares, archivos PDF y canales RSS.
- Completa documentación y pruebas de alta calidad.

Selección del *framework*

Como *framework* se escogió Symfony 2, ya que automatiza y simplifica muchas de las practicas estándares del desarrollo web, como: la generación dinámica de formularios; altos niveles de seguridad en el sistema; procesos de administración; posee una capa de abstracción de objetos que se encarga de mapear los datos relacionales a objetos, la cual protege al sistema contra las inyecciones SQL y otros ataques a la bases de datos; además posee un módulo para realizar pruebas unitarias y funcionales de forma automatizada.

1.5.4 Servidor Web

Es un programa que gestiona aplicaciones en el lado del servidor, realizando conexiones bidireccionales y unidireccionales, síncronas y asíncronas con el cliente, generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza un protocolo. Generalmente se utiliza

el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI (12) (22).

Básicamente un servidor WEB consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido sea solicitado. Cuando el cliente recibe el código con la respuesta a una solicitud, este lo interpreta y finalmente muestra el resultado en la pantalla.

Apache

El Servidor Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows y Macintosh, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Apache tiene amplia aceptación de la comunidad desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años (23). Entre sus características más importantes es que esta implementado de forma modular, lo que permite que el núcleo sea pequeño y manejable con sus funciones básicas, mientras sus funcionalidades adicionales se integran mediante diferentes módulos (17).

Otras de sus características de gran utilidad son:

- Soporte de host virtuales: Apache es además uno de los primeros servidores Web en soportar tanto host basados en IP como host virtuales.
- Soporte de autentificación HTTP: Apache soporta autentificación básica basada en la Web.
- Soporte de scripts PHP: este es un lenguaje muy utilizado y Apache ofrece un amplio soporte de PHP.
- Estado del servidor y adaptación de registros: Apache le da una gran cantidad de flexibilidad en el registro y la monitorización del estado del servidor. El estado del servidor puede monitorizarse mediante un navegador Web y adaptar sus archivos de registro a su gusto.

IIS

IIS (*Internet Information Services*) es un servidor web que proporciona servicios con las herramientas y funciones necesarias para administrar de forma sencilla un servidor web. Brinda servicios como FTP, SMTP, NNTP y HTTP/HTTPS. Además procesa páginas de ASP y ASP.NET; y puede incluir también PHP o Perl. Una de las desventajas que presenta este servidor web es que solo se puede utilizar en sistemas

de Windows (24). El servidor IIS actualmente forma parte de la distribución estándar de Windows, por lo que es muy popular entre los usuarios del sistema operativo. Este servicio convierte a una PC en un servidor web para Internet o una intranet, permitiéndole publicar páginas web tanto local como de forma remota (25).

En la lista siguiente se muestran algunas de las ventajas de usar IIS (26):

- Ampliar la seguridad en Internet a través de un espacio de servidor reducido y el aislamiento automático de las aplicaciones
- Implementar y ejecutar fácilmente aplicaciones web ASP.NET, ASP clásico y PHP en el mismo servidor
- Aislar las aplicaciones proporcionando a los procesos de trabajo una identidad única y una configuración de espacio aislado de forma predeterminada, con la consiguiente reducción de los riesgos de seguridad
- Agregar, quitar e incluso reemplazar fácilmente componentes de IIS integrados por módulos personalizados, adecuados a las necesidades del cliente
- Agilizar el sitio web mediante las características integradas de almacenamiento en caché dinámico y compresión mejorada

Selección del Servidor Web

Se seleccionó el servidor Web Apache debido a las ventajas de este respecto a IIS, entre las que se pueden mencionar que es un servidor Web *Open Source*, multiplataforma, modular, extensible y cuenta con una amplia documentación que puede servir de ayuda y soporte y a la cual se puede acceder en Internet.

1.5.5 Sistema Gestor de Bases de Datos

Un Sistema Gestor de Base de Datos (DBMS por sus siglas en inglés) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Además brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso

fácil y eficiente a los mismos. Proporciona una interfaz entre las bases de datos y las aplicaciones que la utilizan.

MySQL

MySQL es un sistema gestor de bases de datos relacionales eficiente, sólido y flexible. Es idóneo para la creación de bases de datos con acceso desde páginas web dinámicas, posibilitando realizar múltiples y rápidas consultas. Está desarrollado en C y C++, facilitando su integración en otras aplicaciones desarrolladas en estos lenguajes (27).

Es un sistema cliente/servidor, por lo que permite trabajar como servidor multiusuario y de subprocesamiento múltiple, o sea, cada vez que se crea una conexión con el servidor, el programa servidor establece un proceso para manejar la solicitud del cliente, controlando así el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso a usuarios autorizados solamente (28). Es uno de los sistemas gestores de bases de datos más utilizados en la actualidad. Algunas de las compañías que utilizan MySQL son Yahoo! Finance, Google y Motorola.

PostgreSQL

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postgres. Es un gestor de código abierto, brinda un control de concurrencia multi-versión (MVCC por sus siglas en inglés) que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación (29).

Posee características significativas del motor de datos, entre las que se pueden incluir subconsultas, valores por defecto, restricciones a valores en los campos (*constraints*) y disparadores (*triggers*). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales (29).

El código fuente se encuentra disponible para todos sin costo alguno. Está disponible para 34 plataformas con la última versión estable. Es totalmente compatible con ACID (acrónimo de *Atomicity, Consistency, Isolation and Durability*). Posee una integridad referencial e interfaces nativas para lenguajes como C, C++, PHP, PERL, PYTHON y RUBY. Funciona en múltiples sistemas operativos Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows (29).

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico.

Selección del SGBD

Se decide utilizar PostgreSQL ya que permite alcanzar escalabilidad ajustando de forma óptima el número de procesadores y la cantidad de memoria, es capaz de soportar una mayor cantidad de peticiones simultáneas, es un gestor multiusuario, multiprogramado, con arquitectura cliente-servidor y control de privilegios de acceso. Su funcionamiento es mucho más poderoso mientras más altos sean los volúmenes de datos. Implementa el uso de subconsultas y transacciones y ofrece la capacidad de almacenar procedimientos y comprobar la integridad en la propia base de datos. Además el personal cuenta con gran experiencia en el uso de PostgreSQL, lo cual no es el caso para MySQL.

1.5.6 Entornos de Desarrollo Integrado

Los entornos de desarrollo integrados (IDE) son aplicaciones compuestas a su vez de varias herramientas que se complementan para dar soporte a los desarrolladores de software de manera que facilitan y optimizan el trabajo a la hora de escribir el código de un programa, corregirlos y ejecutarlos. Algunos ejemplos de herramientas que pueden estar integradas a un IDE son:

- Editor de código.
- Compilador.
- Intérprete.
- Constructor de interfaz gráfica.
- Depurador.

NetBeans

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para otros lenguajes de programación. Existe además un número importante de módulos para extender el IDE. Además es un producto libre y gratuito sin restricciones de uso.

Todas las funciones del IDE son provistas por módulos. Cada módulo proporciona una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones (30).

Algunas de las principales ventajas de NetBeans se muestran a continuación (31):

- El editor NetBeans PHP proporciona plantillas de código y herramientas de generación de código, como la generación de "g*etter"* y "*setter*", refactorización, renombrado instantáneo, consejos y soluciones rápidas y completamiento de código inteligente.
- Provee resaltado sintáctico y semántico del código.
- Permite la compilación en tiempo real.
- Es compatible con diferentes *frameworks* para PHP como Zend y Symfony. Permite crear nuevos proyectos con estos *frameworks*, ejecutar los comandos de los *frameworks*, usa las anotaciones de los *frameworks* y realiza completamiento de código referente al *framework* utilizado.
- El editor de PHP se integra dinámicamente con HTML, JavaScript y las características de edición de CSS.
- Es totalmente compatible con el desarrollo iterativo, por lo que la experimentación de proyectos PHP sigue los patrones clásicos familiares para los desarrolladores web.
- La producción de un programa PHP se muestra en una pantalla de líneas de comandos en el propio IDE y se puede inspeccionar el código HTML generado sin tener que cambiar a un navegador.

Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar IDEs, como *Java Development Toolkit* (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse). Sin embargo, también se puede usar para otros tipos de aplicaciones cliente.

Algunas de las ventajas más importantes para Eclipse son (32):

- Eclipse dispone de un Editor de texto con resaltado de sintaxis.
- Compilación de código en tiempo real.

- Tiene pruebas unitarias con JUnit, control de versiones con CVS (*Concurrent Versioning System*), integración con Ant, asistentes (*wizards*) para creación de proyectos, clases, tests y refactorización.
- A través de *plugins* libremente disponibles es posible añadir control de versiones con Subversion e integración con Hibernate.
- Eclipse provee al programador con *frameworks* para el desarrollo de aplicaciones gráficas, definición y manipulación de modelos de software, aplicaciones web, etc.

Selección del IDE

Como entorno de desarrollo integrado se escogió NetBeans dado principalmente que este IDE soporta y permite el desarrollo ágil de aplicaciones web mediante el *framework* seleccionado, Symfony 2. Además cuenta con un alto nivel profesional independientemente de ser un producto libre y totalmente gratuito.

1.5.7 Metodología de Desarrollo de Software y Lenguaje de Modelado.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas, herramientas y soporte documental para desarrollar un producto informático, en el que se van indicando paso a paso todas las actividades a realizar para alcanzar los objetivos propuesto (33). En cualquier proceso de desarrollo de software, esta define: “Qué” debe hacer el software, “Quién” debe realizar cada actividad, “Cuándo” hacerla y “Cómo” se debe hacer (34). Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías de desarrollo se pueden dividir en dos grupos de acuerdo con sus características y los objetivos que persiguen: ágiles y robustas (o tradicionales).

Metodologías robustas o tradicionales

Están guiadas por una fuerte planificación. Centran su atención en llevar una documentación exhaustiva de todo el proceso de desarrollo y en cumplir con un plan de proyecto, definido en la fase inicial del mismo. Entre las metodologías robustas se encuentran: MSF (por sus siglas en inglés *Microsoft Solution Framework*), MÉTRICA 3 y RUP (siglas de *Rational Unified Process*) (33).

Metodologías ágiles

Las metodologías ágiles se caracterizan por hacer énfasis en la comunicación cara a cara entre el cliente y el desarrollador, quienes trabajan constantemente juntos y establecen así una estrecha comunicación.

Estas metodologías están orientadas al resultado del producto y no a la documentación; exige que el proceso sea adaptable, permitiendo realizar cambios de último momento. Se puede mencionar como metodologías agiles XP, SCRUM y *Crystal Methodologies*.

XP

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, la comunicación fluida entre todos los participantes, la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre.

XP incluye 12 prácticas fundamentales (33) (36):

- Jugar el juego de planificación.
- Hacer pequeños *Releases*.
- Hacer historias y usar metáforas.
- Diseñar simple.
- Probar –Testear.
- Rearmar – Refactorizar.
- Programar por pares.
- Propiedad Colectiva.
- Integrar Continuamente.
- Semanas de 40 horas.
- Cliente *On-Site*.
- Usar Estándares de Codificación.

La metodología XP divide su desarrollo en cuatro fases principales: Planificación, donde los clientes escriben las funcionalidades con que debe cumplir el sistema, especificando las características que deben ser adicionadas al sistema mediante las Historias de Usuario; Diseño, donde se crean diseños simples y

claros, siendo constantemente revisados y probablemente modificados durante el desarrollo, implementando con el orden definido en las iteraciones planificadas; Desarrollo, para ir cumpliendo con el objetivo de realizar entregas frecuentes al cliente el cual debe estar presente solucionando las dudas cara a cara; Pruebas, que se realizan en todo momento asegurando el cumplimiento de lo planteado en el diseño, participando el equipo de desarrollo y el cliente que brinda los aportes más importantes (36).

Selección de la metodología

A partir del análisis de las metodologías se llegó a la conclusión de utilizar XP ya que es adaptable al software a desarrollar. Propone enfocarse en el proceso de implementación y no hace mucho énfasis en la documentación, permitiendo al desarrollador emplear más tiempo en la primera tarea. Los requerimientos actuales pueden estar sujetos a futuros cambios, este es un punto donde la metodología es flexible, ya que permite administrar estos cambios de forma óptima. Se requiere un grupo pequeño de programadores para trabajar con esta metodología, lo cual no es así para RUP que define un mayor número de roles y actividades.

1.6 Conclusiones del capítulo

Una vez terminado el estudio y análisis referente al campo de acción y concluido la fundamentación de las metodologías y herramientas a utilizar para el desarrollo de la aplicación se arriban a las siguientes conclusiones:

- La aplicación a desarrollar implementará diferentes funcionalidades inherentes a los distintos sistemas analizados.
- Para el desarrollo de la aplicación se decide utilizar las herramientas y tecnologías: HTML 5, CSS 3, JavaScript 1.8.5, PHP 5, WebGL 1.0.1, Apache 2.2, PostgreSQL 9.1, NetBeans 7.3 y Symfony 2.4.
- Se propone realizar el desarrollo de la solución propuesta siguiendo los métodos y principios planteados por la metodología XP.

Capítulo 2: Características y diseño de la solución.

En el presente capitulo se procede confeccionar la propuesta del sistema para la gestión de componentes gráficos y sonoros basándonos en la metodología de desarrollo XP. Se definirá la arquitectura del sistema, los patrones de diseño utilizados, los requisitos funcionales y no funcionales dadas por el cliente mediantes Historias de Usuarios (HU), la estimación de esfuerzo por HU, las iteraciones planificadas para el proceso de implementación y las tarjetas CRC.

2.1 Modelo de Dominio

Para una mejor comprensión de los procesos vinculados al campo de acción se confecciona el modelo de dominio mediante un diagrama de clases UML. El diagrama consiste en una representación visual de las clases conceptuales u objetos del mundo real y sus relaciones, que son de interés conocerlas.

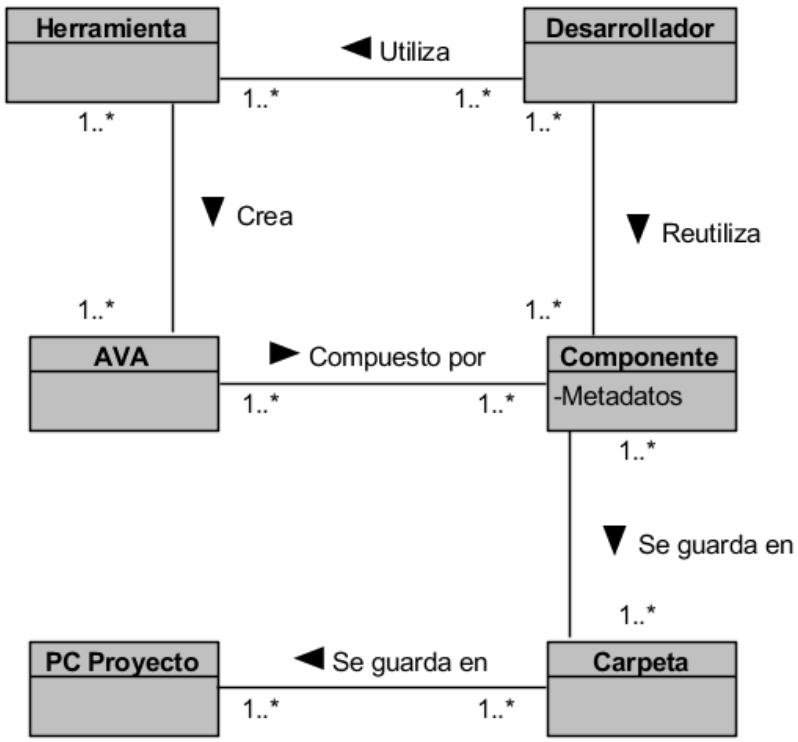


Figura 4: Modelo de Dominio.

Desarrollador: Personal que integra el equipo de desarrollo del proyecto UCI-CIMEQ.

Herramienta: Aplicaciones que se utilizan para crear y editar los AVA. Ejemplos: Blender y 3DMax.

AVA: Ambientes Virtuales de Aprendizaje desarrollados por el proyecto.

Componente: Componentes que integran un AVA. Ejemplos: objetos 3D y sonidos.

Carpeta: Carpeta o directorio donde se guardan los componentes.

PC Proyecto: Máquinas donde trabajan los desarrolladores del proyecto.

2.2 Solución propuesta

Después de haber analizado la situación del proyecto UCI-CIMEQ se concluye que es necesario automatizar algunas de las acciones para agilizar la gestión de los componentes del proyecto.

Se propone la implementación de un sistema que permita almacenar todos los componentes generados en el proyecto de forma centralizada. Este permite automatizar todo el proceso de gestión de componentes que se realizan en el proyecto, abarcando las acciones de captura de los componentes y sus metadatos correspondientes, almacenamiento y publicación. Además brinda la posibilidad de gestionar los usuarios mediante el servicio LDAP del dominio UCI. Esto posibilita definir roles para cada usuario, con el fin de garantizar la seguridad de la solución.

Para el proceso de publicación se hará uso de las más recientes potencialidades que brinda de HTML5, como la reproducción de los componentes de audio y la capacidad de mostrar imágenes y animaciones tridimensionales complejas. Todo esto haciendo uso de sus nuevas etiquetas “<audio>” y su capacidad de integración con WebGL mediante la etiqueta “<canvas>” y JavaScript sin la necesidad de utilizar *plugins.* Garantizando la compatibilidad entre navegadores como Firefox, Opera, Safary y Chrome.

Se brinda al usuario las opciones de búsqueda y filtrado de los componentes, estas opciones permiten la especificación de diferentes criterios determinados por los metadatos que se almacenan en el proceso de captura para agilizar y optimizar el proceso de reutilización de los componentes.

El sistema consiste de una aplicación Web implementada sobre el *framework* Symfony 2 y que basa su arquitectura en el patrón Modelo-Vista-Controlador (MVC, por sus siglas en inglés). Para almacenar y consultar la información manejada por el sistema se utiliza el gestor de base de datos PostgreSQL.

2.2.1 Usuarios del sistema

Tabla 1: Usuarios del sistema.

Usuario	Descripción
Administrador	Realiza la gestión del contenido del sistema y de los usuarios. Cuenta con acceso a todas las funcionalidades definidas en el sistema. Debe autenticarse, de lo contrario navega como un usuario de tipo Invitado.
Miembro	Solo dispones de los servicios de búsqueda, carga, descarga y actualización de componentes. No tiene permisos para eliminar componentes. Debe autenticarse.
Invitado	Puede realizar búsquedas, visualizar y descargar los contenidos, no puede subir, actualizar o eliminar componentes. No necesita autenticación.

2.4 Fase de Exploración

Fase en la cual el equipo de desarrollo dialoga con el cliente para definir a grandes rasgos las características del sistema, las cuales se traducen según la metodología en cuestión XP a Historias de Usuarios, siendo estos los principales objetivos a cumplir por la aplicación. Además el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto (40).

2.4.1 Historias de Usuario

Las Historias de Usuario (HU) son la técnica utilizada en XP para especificar los requisitos del software, donde el usuario decide qué tareas realizará la aplicación. Las HU se utilizan como herramientas para dar a conocer los requerimientos funcionales y no funcionales del sistema al equipo de desarrollo. Son pequeños textos en los que el cliente describe las funcionalidades, prestaciones y características que desea para el sistema (41).

Una HU está compuesta por el **número** que la identifica, el **nombre**, los **usuarios** que van a interactuar con la aplicación, los **puntos de estimación** que son las semanas que durará la implementación de dicha

HU, la **prioridad** que va a tener para el negocio, la iteración en que será implementada, los **riesgos** en el desarrollo y una breve **descripción** de lo que realizará la misma.

Tabla 2: HU Gestionar audio.

Historia de Usuario	
Número: 1	Nombre: Gestionar audios
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 1
Prioridad: Alta	Riesgos en el desarrollo: Media
Descripción: El usuario puede subir, actualizar, descargar y eliminar audio.	
Observaciones: Si el usuario esta autenticado como invitado solo puede descargar audios, si es Miembro puede actualizar solo los audios subidos por él mismo y no puede eliminar.	

Tabla 3: HU Gestionar objetos 3D.

Historia de Usuario	
Número: 2	Nombre: Gestionar objetos 3D
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 1
Prioridad: Alta	Riesgos en el desarrollo: Alta
Descripción: El usuario puede subir, actualizar, descargar y eliminar objetos 3D.	
Observaciones: Si el usuario esta autenticado como invitado solo puede descargar objetos, si es Miembro puede actualizar solo los objetos subidos por él mismo y no puede eliminar.	

Tabla 4: HU Visualizar contenido.

Historia de Usuario	
Número: 3	Nombre: Visualizar contenido
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 1
Prioridad: Alta	Riesgos en el desarrollo: Media
Descripción: Permite visualizar los datos de un contenido en concreto. Si es un objeto 3D, se muestran todos sus datos e imágenes de muestra. Si es un audio, se muestran todos sus datos y se brinda la funcionalidad de reproducir el audio desde la web.	
Observaciones:	

Tabla 5: HU Registrar usuario.

Historia de Usuario

Número: 4	Nombre: Registrar usuario
Usuario: Invitado	Puntos de estimación: 1
Prioridad: Media	Riesgos en el desarrollo: Media
Descripción: El usuario puede registrarse en la aplicación.	
Observaciones: El usuario debe registrarse mediante su usuario y contraseña del dominio UCI.CU.	

Tabla 6: HU Gestionar rol.

Historia de Usuario	
Número: 5	Nombre: Gestionar rol
Usuario: Administrador	Puntos de estimación: 0.5
Prioridad: Media	Riesgos en el desarrollo: Baja
Descripción: Permite administrar el rol de cada usuario registrado.	
Observaciones: El usuario debe estar autenticado.	

Tabla 7: HU Listar audios.

Historia de Usuario	
Número: 6	Nombre: Listar audios
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 0.5
Prioridad: Alta	Riesgos en el desarrollo: Media
Descripción: Permite listar todos los audios que hay en la aplicación.	
Observaciones:	

Tabla 8: HU Listar objetos 3D.

Historia de Usuario	
Número: 7	Nombre: Listar objetos 3D
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 0.5
Prioridad: Alta	Riesgos en el desarrollo: Media
Descripción: Permite listar todos los objetos 3D que hay en la aplicación.	
Observaciones:	

Tabla 9: HU Listar usuarios.

Historia de Usuario

Número: 8	Nombre: Listar usuarios
Usuario: Administrador	Puntos de estimación: 0.5
Prioridad: Baja	Riesgos en el desarrollo: Baja
Descripción: Permite listar todos los usuarios registrados en la aplicación.	
Observaciones:	

Tabla 10: HU Búsqueda y filtrado.

Historia de Usuario	
Número: 9	Nombre: Búsqueda y filtrado
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 1
Prioridad: Media	Riesgos en el desarrollo: Media
Descripción: Permite realizar búsquedas sobre los contenidos de la aplicación y filtrar los listados por diferentes criterios definidos en la aplicación.	
Observaciones:	

Tabla 11: HU Visualizar objeto 3D.

Historia de Usuario	
Número: 10	Nombre: Visualizar objeto 3D
Usuario: Invitado, Miembro, Administrador	Puntos de estimación: 1
Prioridad: Baja	Riesgos en el desarrollo: Alta
Descripción: Permite visualizar un objeto 3D en la web.	
Observaciones:	

2.4.2 Especificación de los requisitos de Software

Los requerimientos de un sistema describen los servicios que ha de ofrecer el sistema y las restricciones asociadas a su funcionamiento. Los requisitos de software son "*las condiciones que debe cumplir o poseer un sistema o uno de sus componentes para satisfacer un contrato, una norma o una especificación*" (42).

A continuación se muestra una tabla con la lista de reserva del producto en la cual se detallan los requisitos definidos para la aplicación.

Tabla 12: Lista de reserva del producto

Requisitos funcionales

No	Descripción	Estimación	Estimado por
Prioridad: Alta			
1	Subir audio	0.3	Analista
2	Actualizar audio	0.3	Analista
3	Descargar audio	0.2	Analista
4	Eliminar audio	0.2	Analista
5	Subir objeto 3D	0.3	Analista
6	Actualizar objeto 3D	0.3	Analista
7	Descargar objeto 3D	0.2	Analista
8	Eliminar objeto 3D	0.2	Analista
9	Listar audios	0.5	Analista
10	Listar objetos 3D	0.5	Analista
11	Visualizar contenido	1	Analista
Prioridad: Media			
12	Registrar usuario	1	Analista
13	Administrar rol	0.5	Analista
14	Listar usuario	0.5	Analista
Prioridad: Baja			
15	Búsqueda y filtrado	1	Analista
16	Visualizar objeto 3D	1	Analista
Requisitos no funcionales			
Software			
17	Instalación del servidor web Apache en su versión 2.2.		
18	Instalación del sistema gestor de BD PostgreSQL 9.1.		
19	Uso de los navegadores Firefox 4.0, Chrome 11, Safari 10.6 u Opera 12 o versiones superiores que brinden soporte para WebGL.		
Integridad			
20	La información solo será modificada o eliminada por las personas autorizadas.		
Interfaz			

21	La aplicación presentará una interfaz amigable la cual debe permitir una rápida comprensión por parte de los usuarios.		

2.5 Fase de Planificación

La planificación es una fase corta, en la que el cliente y el grupo de desarrolladores acuerdan el orden en que se deben implementar las HC. Se realiza una estimación del esfuerzo necesario de cada una de ellas y asociadas a estas, las entregas. Típicamente en esta fase se realizan una o varias reuniones grupales de planificación. El resultado de esta fase es un Plan de Entregas, o "*Release Plan*" (43).

2.5.1 Estimación de esfuerzo por historia de usuario

Tabla 13: Estimación de esfuerzo por HU.

No	Historias de Usuario	Estimación(semanas)
1	Gestionar audio	1
2	Gestionar objeto 3D	1
3	Visualizar contenido	1
4	Registrar usuario	1
5	Gestionar rol	0.5
6	Listar audios	0.5
7	Listar objetos 3D	0.5
8	Listar usuarios	0.5
9	Búsqueda y filtrado	1
10	Visualizar objeto 3D	1
	Total	8

2.5.2 Iteraciones

Una vez que se describen las Historias de Usuario y se realiza la estimación previa del esfuerzo para las Historias de Usuario, se procede a planificar la etapa de implementación del sistema. Esta planificación se enfoca en definir la prioridad para desarrollar las Historias de Usuarios organizadas por iteraciones, para la cuales se estima una fecha de entrega (44).

Plan de iteraciones

Las HU son desarrolladas y probadas en un ciclo de iteración, siendo el cliente quien decide qué historias se implementarán en cada iteración para maximizar su valor. Se decidió agrupar las HU en tres iteraciones de manera que en la primera iteración se pueda establecer una arquitectura base del sistema a utilizar durante el resto del proyecto. En una segunda iteración se agruparon las HU que permitieran establecer la base para establecer la seguridad del sistema y en una última iteración aquellas que representaran un nivel de prioridad más bajo para el cliente y que dependieran de la previa implementación de otras HU.

1ra Iteración

Esta primera iteración tiene como tareas fundamentales la implementación de las Historias de Usuarios 1, 2, 6, 7 y 3, debido a que estas conforman la base del sistema y constituyen la base para las restantes Historias de Usuario. Además estas HU poseen un nivel de prioridad alta para el cliente. Las HU que se incluyen en esta iteración son responsables del proceso de gestión de los componentes gráficos y sonoros y su publicación para que los usuarios puedan acceder a los mismos.

2da Iteración

La segunda iteración tiene como tareas fundamentales la implementación de las Historias de Usuario 4, 5 y 8, que tienen relación con la seguridad del sistema. Estas permiten controlar los niveles de permisos y definir los roles para los usuarios.

3ra Iteración

En esta última iteración se agrupan las Historias de Usuarios 9 y 10, que se encargan de brindar mayor usabilidad a la aplicación al incluirle la funcionalidad de búsqueda sobre los componentes que gestiona el sistema y la visualización de objetos 3D en la web.

En consecuencia con lo planteado anteriormente, el plan de iteraciones queda de la siguiente manera.

Tabla 14: Plan de iteraciones

Iteraciones	Historias de Usuarios	Duración
1ra Iteración	1. Gestionar audio 2.Gestionar objeto 3D 6.Listar audios	4 semanas

	7.Listar objetos 3D 3.Vizualizar contenido	
2da Iteración	4.Registrar usuario 5.Gestionar rol 8.Listar usuario	2 semanas
3ra Iteración	9.Búsqueda y filtrado 10.Visualizar objeto 3D	2 semanas

2.5.3 Plan de entregas

En el plan de entregas se establece el orden de las HU que se agrupan para establecer una entrega. La finalidad de este plan es mostrar la duración de cada iteración, lo que ayuda a obtener una idea aproximada del tiempo necesario para desarrollar el sistema en su totalidad.

Tabla 15: Plan de entrega.

Historia de Usuario	Fin 1ra Iteración	Fin 2da Iteración	Fin 3ra Iteración
1.Gestionar audio	Versión 1.0	Terminado	Terminado
2.Gestionar objeto 3D	Versión 1.0	Terminado	Terminado
6.Listar audios	Versión 1.0	Terminado	Terminado
7.Listar objetos 3D	Versión 1.0	Terminado	Terminado
3.Vizualizar contenido	Versión 1.0	Terminado	Terminado
4.Registrar usuario		Versión 1.0	Terminado
5.Gestionar rol		Versión 1.0	Terminado
8.Listar usuario		Versión 1.0	Terminado
9.Búsqueda y filtrado			Versión 1.0
10.Visualizar objeto 3D			Versión 1.0

2.6 Arquitectura de software

La arquitectura "*constituye el estilo arquitectónico que tendrá el sistema, la estructura y las propiedades de los componentes que ese sistema comprende, y las interrelaciones que tienen lugar entre todos los componentes arquitectónicos del sistema*" (36).

Generalmente una aplicación desarrollada sobre Symfony se caracteriza por su arquitectura de tipo MVC (Modelo-Vista-Controlador) estar estructurado en tres partes fundamentales: el **modelo**, que incluye tanto la estructura de la base de datos como todo el código de consulta; la **vista**, que constituye el modo en que se van a representar los datos y finalmente el **controlador**, que se encarga del procesamiento de los datos y la lógica del sistema (45).

La adopción de esta arquitectura brinda una serie de ventajas como:

- Facilita agregar nuevos tipos de datos según sea requerido por la aplicación ya que son independientes del funcionamiento de las otras capas.
- Crea independencia de funcionamiento.
- Facilita el mantenimiento en casos de errores.
- Permite mayor escalabilidad en el sistema.

2.7 Patrones de diseño empleados

Los desarrolladores orientados a objetos con experiencia (y otros desarrolladores de software) acumulan un repertorio tanto de principios generales como de soluciones basadas en aplicar ciertos estilos que les guían en la creación de software. Estos principios y estilos, se definen describiendo un problema común para múltiples escenarios del desarrollo de software y su solución de forma genérica, los cuales son conocidos como patrones de diseño (46).

Existen una gran variedad de patrones que son utilizados de forma frecuente, entre los más destacados se encuentran los GRASP y los GoF. Los GRASP (*General Responsibility Assignment Software Patterns*) o patrones generales de software para asignar responsabilidades, recogen los principios más importantes y elementales para el diseño de software orientado a objetos. Por otra parte los patrones GoF (*Gang of Four*) son un grupo de 23 patrones recogidos en el libro *Design Patterns*; que deben su nombre a que son cuatro los autores del libro: Erich Gamma; Richard Helm, Ralph Johnson y John Vlissides.

A continuación se muestran los patrones utilizados en el desarrollo de la aplicación:

GRASP

- **Experto en Información:** Este patrón plantea que la asignación de las responsabilidades a una clase deben estar acorde con la información necesaria para manejar dichas responsabilidades. Esto permite que el sistema sea más fácil de entender, mantener y de reutilizar sus componentes en futuras aplicaciones. Este patrón se evidencia en las clases “Objeto3D” y “Audio” que son las

encargadas de establecer los datos de los nuevos componentes subidos y almacenarlos en el directorio adecuado ya que cuentan con los atributos y funciones adecuadas para ello.

- **Creador:** Este patrón plantea definir principios generales para la asignación de las responsabilidades de creación. En la aplicación este patrón se manifiesta dada la existencia y función de los tres controladores “AudioController”, “ObjetoController” y “AsuarioController” las cuales son las únicas responsables de la creación de las instancias de las clases “Audio”, “Objeto3D” y “Usuario” respectivamente.
- **Bajo Acoplamiento:** El objetivo a seguir por este patrón es tratar de asignar responsabilidades de manera que se mantenga un nivel de dependencia débil entre las clases o componentes del sistema. Este patrón está presente en las clases “Audio”, “Objeto3D” y “Usuario” las cuales no dependen de otras clases o componentes para realizar sus funciones a excepción de la clase “Usuario” la cual depende solo de la clase “LDAP” para obtener los datos de los usuarios.
- **Alta Cohesión:** Con este patrón se trata de asignar responsabilidades de manera que se mantenga una alta cohesión o sea que las responsabilidades de los componentes estén altamente relacionadas y al mismo tiempo estos no realicen una gran cantidad de trabajo. Ejemplo de ello en la aplicación son la implementación de las cases “AudioType”, “ObjetoType” y “UsuarioType” a las cuales se delegan la responsabilidad de crear los formularios y la clase “LDAP” la cual se encarga de realizar la autenticación y recopilación de los usuarios del sistema. De manera que al mismo tiempo ahorran y evitan sobrecarga de responsabilidades a las clases controladoras.
- **Controlador:** Este patrón se encarga de definir clases o componentes que serán los responsables de controlar los eventos generados por el sistema, usuario u otro actor externo. Estas clases o componentes son llamados Controladores, los cuales no pertenece a la interfaz de usuario y se encargan de manejar los eventos del sistema, ejemplos de estas en la aplicación son las clases controladoras “AudioController”´, “ObjetoController” y “AudioController”.

Patrones GoF

- **Decorador:** Este patrón responde a la necesidad de modificar, agregar o retirar dinámicamente funcionalidades o responsabilidades a un objeto. Ejemplo de su uso en la aplicación son las plantillas HTML para la interfaz del usuario. En Symfony 2 las plantillas son tratadas como objetos y pueden extender de otras plantillas e incluir otras dentro de ellas, por lo que se define una plantilla “base.html” que contiene lo que es común para todas las interfaces a la cual se le agregan otras plantillas asociadas para cada acción, conformando así las vistas que se muestran al usuario.

2.8 Prototipo de interfaz de usuario

Para tener una idea de la estructura de la interfaz del sistema y de la distribución de los contenidos, se diseñó un prototipo de interfaz de modo *wireframe* con el cual se definió una plantilla base para estandarizar el diseño gráfico del sistema.

El *wireframe* se enfoca en la estructura visual y en esquematizar el orden y ubicación de los contenidos que se muestran en la interfaz, no en su diseño gráfico (39).
Wireframe:

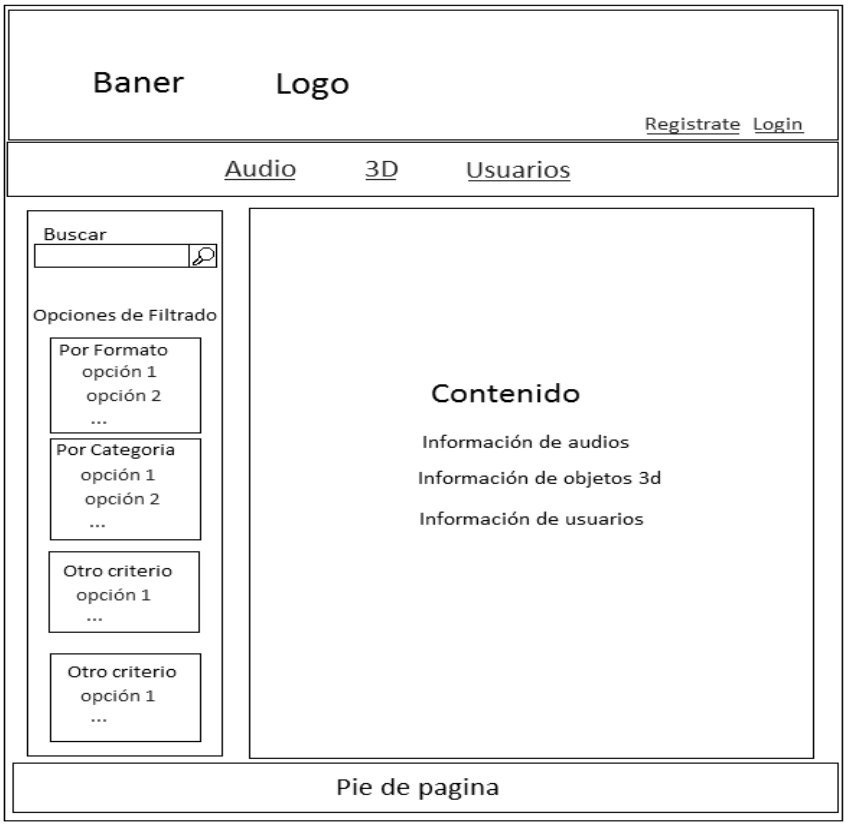


Figura 5: Wireframe de la interfaz del sistema.

Plantilla base:

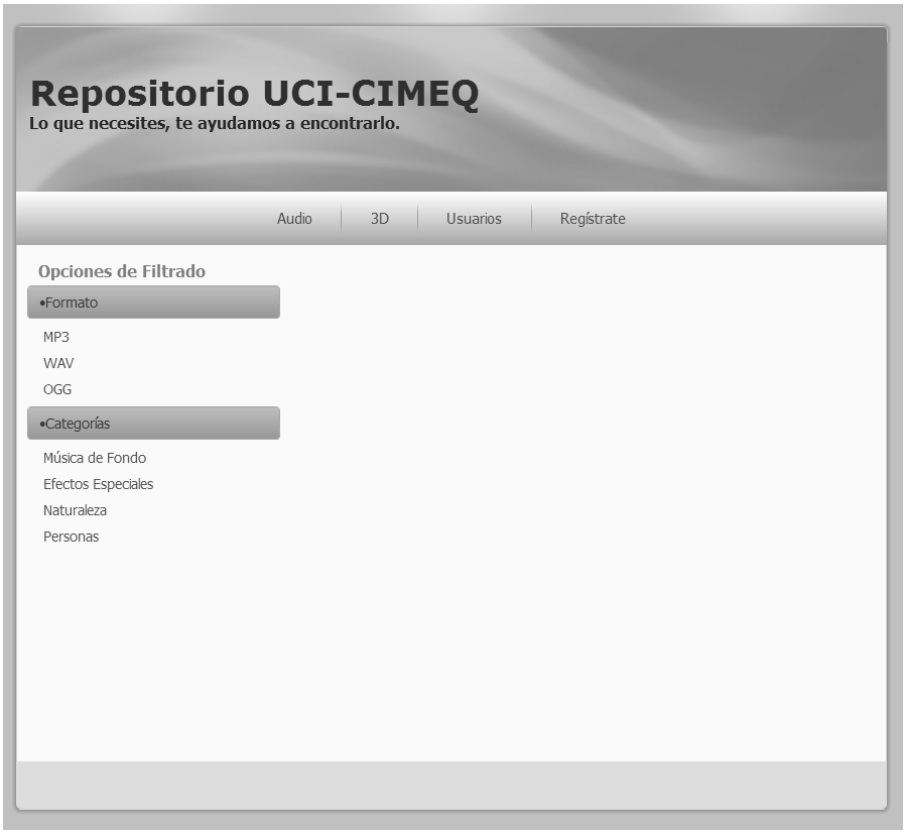


Figura 6: Prototipo de plantilla base.

El prototipo de interfaz descrito anteriormente representa el modelo a seguir durante la implementación del sistema, lo cual no significa que el producto final siga estrictamente dicho modelo, ya que puede estar propensa a cambios en dependencia de las necesidades del cliente durante este periodo de implementación.

2.9 Tarjetas CRC

Las tarjetas CRC son el método de diseño que se utiliza en XP, estas se encargan de representar de forma sencilla las clases del sistema. En ellas se muestra el **nombre de la clase**, sus **responsabilidades** y **colaboradores.** Las responsabilidades son las funciones que realiza y la información que conoce para ejecutar las mismas. Los colaboradores son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades.

A continuación se muestran las tarjetas CRC identificadas para implementar las historias de usuarios.

Tabla 16: Tarjeta CRC Audio.

Audio	
Responsabilidades	Colaboradores
• Esta clase contiene la información necesaria para generar un nuevo audio. • Permite definir y cambiar el valor los atributos de un audio. • Permite devolver el valor de cualquier atributo de un audio. • Permite subir un archivo de audio.	

Tabla 17: Tarjeta CRC Objeto 3D.

Objeto3D	
Responsabilidades	Colaboradores
• Esta clase contiene la información necesaria para generar un nuevo objeto 3D. • Permite definir y cambiar el valor los atributos de un objeto 3D. • Permite devolver el valor de cualquier atributo de un objeto 3D. • Permite subir un objeto 3D.	

Tabla 18: Tarjeta CRC Usuario.

Usuario	
Responsabilidades	Colaboradores
• Esta clase contiene la información necesaria para generar un nuevo usuario. • Permite definir y cambiar el valor los atributos de un usuario. • Permite devolver el valor de cualquier atributo de un usuario.	LDAP

Tabla 19: Tarjeta CRC CategoriaAudio.

CategoriaAudio	
Responsabilidades	Colaboradores
• Esta clase contiene la información necesaria para generar una nueva categoría de audio.	

Tabla 20: Tarjeta CRC CategoriaObjeto.

CategoriaObjeto	
Responsabilidades	Colaboradores
• Esta clase contiene la información necesaria para generar una nueva categoría de objetos 3D.	

Tabla 21: Tarjeta CRC AudioController.

AudioController	
Responsabilidades	Colaboradores
• Adicionar nuevo audio. • Actualizar audio. • Eliminar audio. • Buscar audio. • Listar audios. • Mostrar detalles de un audio. • Filtrar audio por un atributo determinado.	Audio AudioType

Tabla 22: Tarjeta CRC ObjetoController.

ObjetoController	
Responsabilidades	Responsabilidades
• Adicionar objeto 3D. • Actualizar objeto 3D. • Eliminar objeto 3D. • Buscar objeto 3D. • Listar objetos 3D. • Mostrar detalles de un objeto 3D. • Filtrar objeto 3D por un atributo determinado.	Objeto3D ObjetoType

Tabla 23: Tarjeta CRC UsuarioController.

UsuarioController	
Responsabilidades	Colaboradores
• Registrar nuevo usuario. • Administrar rol de los usuarios. • Buscar usuario. • Listar usuarios. • Mostrar detalles de un usuario. • Filtrar usuario por un atributo determinado.	LDAP Usuario UsuarioRegistroType UsuarioActualizarRolType

Tabla 24 Tarjeta CRC AudioType.

AudioType

Responsabilidades	Colaboradores
• Generar un formulario con los campos necesarios para obtener información de un audio.	

Tabla 25: Tarjeta CRC ObjetoType.

ObjetoType	
Responsabilidades	Colaboradores
• Generar un formulario con los campos necesarios para obtener información de un objeto 3D.	

Tabla 26: Tarjeta CRC UsuarioRegistroType.

UsuarioRegistroType	
Responsabilidades	Colaboradores
• Generar un formulario con los campos necesarios para registrar un usuario.	

Tabla 27: Tarjeta CRC UsuarioActualizarRolType.

UsuarioActualizarRolType	
Responsabilidades	Colaboradores
• Generar un formulario con los campos necesarios para actualizar el rol de un usuario.	

Tabla 28: Tarjeta CRC LDAP.

LDAP	
Responsabilidades	Colaboradores
• Permite autenticar un usuario por el dominio uci.cu y obtener sus datos.	

Conclusiones del capítulo

Finalizado el presente capítulo donde se presentan las decisiones y artefactos generados durante las fases de exploración y planificación se arriba a las siguientes conclusiones:

- La aplicación cuenta con un total de diez Historias de Usuarios a implementar, para lo cual se planifican tres iteraciones con un tiempo total estimado de 8 semanas.
- Se determinó el patrón arquitectónico MVC, así como algunos de los patrones de diseños GRASP y GoF que se utilizan para la correcta implementación de la solución.

- La confección del prototipo de interfaz y las tarjetas CRC aseguran las bases para la posterior implementación de las historias de usuarios.

Capítulo 3: Implementación y Prueba de la solución.

Según la metodología XP las tareas de implementación y prueba se desarrollan durante las fases de Iteraciones y Producción, apoyándose en los resultados obtenidos en las fases de Exploración y Planificación. Además plantea que estas tareas deben realizarse de forma iterativa e incremental, donde al final de cada iteración se obtiene un producto que debe ser probado y expuesto al cliente para lograr una constante retroalimentación entre este y el equipo de desarrollo. En este capítulo se muestran los detalles de cada iteración exponiendo las tareas de ingeniería realizadas por cada HU, el modelo de despliegue de la aplicación y finalmente se muestran las pruebas realizadas al sistema para su validar su correcto funcionamiento y aceptación por parte del cliente.

3.1 Iteraciones

Esta fase se caracteriza por establecer y seguir un calendario, el cual se divide en iteraciones. Dicho calendario se basa en el plan de iteraciones confeccionado durante la planificación. Durante esta fase, el trabajo se centra en el cumplimiento de las tareas de ingeniería definidas para dar cumplimiento a las HU agrupadas encada iteración.

3.1.1 Plan de Tareas de Ingeniería

El plan de tareas consiste en dividir las HU en las tareas que se consideren necesarias para cumplirlas.

Tareas de Ingeniería por HU:

Tabla 29: Tareas de ingeniería.

Historia de Usuario	Tarea de Ingeniería
Gestionar audio	1-Investigar los datos y metadatos de los componentes de audio a tener en cuenta por la aplicación. 2-Definir y crear las entidades para dar soporte a la información de los audios. 3-Crear los componentes para el procesamiento lógico y visual de los contenidos de audio.
Gestionar objeto 3D	1-Investigar los datos y metadatos de los componentes gráficos a tener en cuenta por la aplicación. 2-Definir y crear las entidades para dar soporte a la información de los componentes

	gráficos. 3-Crear los componentes para el procesamiento lógico y visual de los contenidos de componentes gráficos.
Visualizar componentes gráficos	1-Crear los componentes para el procesamiento lógico y visual para mostrar los contenidos del sistema.
Registrar usuario	1-Investigar los datos y metadatos de los usuarios que debe tener por la aplicación. 2-Definir y crear las entidades para dar soporte a la información de los usuarios. 3-Crear los componentes para el procesamiento lógico y visual que permiten al usuario poder registrarse.
Gestionar rol	1-Crear los componentes para el procesamiento lógico y visual que permiten al usuario poder gestionar los roles.
Listar audio	1-Crear los componentes para el procesamiento lógico y visual para listar los audios.
Listar objeto 3D	1-Crear los componentes para el procesamiento lógico y visual para listar los objeto 3D.
Listar usuarios	1-Crear los componentes para el procesamiento lógico y visual para listar los usuarios.
Búsqueda y filtrado	1-Determinar los datos o características de los componentes para realizar las búsquedas y filtrado. 2-Crear los componentes para el procesamiento lógico y visual que permiten al usuario realizar la búsqueda y filtrado de los componentes.
Visualizar objeto 3D	1-Crear los componentes para el procesamiento lógico y visual para mostrar un objeto 3D en la web.

3.1.2 Iteración 1

Para la primera iteración se acordó la implementación de las HU que serán de vital importancia para conformar la estructura base del sistema y al mismo tiempo cuentan con un alto nivel de prioridad para el cliente.

Tareas por HU realizadas en la primera iteración

HU-Gestionar audio:

Tabla 30: Tarea Análisis de los componentes sonoros.

Tarea: Análisis de los componentes sonoros.	
Número de HU: 1	Número de Tarea: 1
Tipo de Tarea: Investigación	Puntos Estimados: 0.2
Fecha Inicio: 01/04/2014	Fecha Fin: 03/04/2014

Descripción: Se seleccionan los datos y metadatos de los contenidos de audio que debe tener la aplicación.

Tabla 31: Tarea Crear el soporte para la información de audio.

Tarea: Crear el soporte para la información de audio.	
Número de HU: 1	Número de Tarea: 2
Tipo de Tarea: Implementación	Puntos Estimados: 0.4
Fecha Inicio: 03/04/2014	Fecha Fin: 05/04/2014
Descripción: Definir y crear las entidades para dar soporte a la información de los audios.	

Tabla 32: Tarea Implementar la lógica y el visual.

Tarea: Implementar la lógica y el visual.	
Número de HU: 1	Número de Tarea: 3
Tipo de Tarea: Implementación	Puntos Estimados: 0.4
Fecha Inicio: 05/04/2014	Fecha Fin: 08/04/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para gestionar los contenidos de audio.	

HU-Gestionar objeto 3D:

Tabla 33: Tarea Análisis de los contenidos de objetos 3D.

Tarea: Análisis de los contenidos de objetos 3D.	
Número de HU: 2	Número de Tarea: 1
Tipo de Tarea: Investigación	Puntos Estimados: 0.2
Fecha Inicio: 09/04/2014	Fecha Fin: 11/04/2014
Descripción: Se analizan y seleccionan los datos y metadatos de los contenidos 3D que debe abarcar la aplicación.	

Tabla 34: Tarea Creación del soporte para la información de objetos 3D.

Tarea: Creación del soporte para la información de objetos 3D.	
Número de HU: 2	Número de Tarea: 2

Tipo de Tarea: Implementación	Puntos Estimados: 0.4
Fecha Inicio: 11/04/2014	Fecha Fin: 13/04/2014
Descripción: Definir y crear las entidades para dar soporte a la información de los objetos 3D.	

Tabla 35: Tarea Implementación de la lógica y el visual.

Tarea: Implementación de la lógica y el visual.	
Número de HU: 2	Número de Tarea: 3
Tipo de Tarea: Implementación	Puntos Estimados: 0.4
Fecha Inicio: 13/04/2014	Fecha Fin: 16/04/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para gestionar los contenidos de objetos 3D.	

HU-Listar audios:

Tabla 36: Tarea Implementación de la funcionalidad listar audios.

Tarea: Implementación de la funcionalidad listar audios.	
Número de HU: 6	Número de Tarea: 1
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 17/04/2014	Fecha Fin: 21/04/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para listar los audios.	

HU-Listar objetos 3D:

Tabla 37: Tarea Implementación de la funcionalidad listar objetos 3D.

Tarea: Implementación de la funcionalidad listar objetos 3D.	
Número de HU: 6	Número de Tarea: 1
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 22/04/2014	Fecha Fin: 26/04/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para listar los audios.	

HU-Visualizar contenido:

Tabla 38: Tarea Implementación de la funcionalidad visualizar contenido de un audio.

Tarea: Implementación de la funcionalidad visualizar contenido de un audio.	
Número de HU: 3	Número de Tarea: 1
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 26/04/2014	Fecha Fin: 29/04/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para visualizar el contenido de un audio.	

Tabla 39: Tarea Implementación de la funcionalidad visualizar contenido de un objeto 3D.

Tarea: Implementación de la funcionalidad visualizar contenido de un objeto 3D.	
Número de HU: 3	Número de Tarea: 2
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 29/04/2014	Fecha Fin: 01/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para visualizar el contenido de un objeto 3D.	

3.1.3 Iteración 2

Tareas por HU realizadas en la segunda iteración:

HU-Registrar usuario:

Tabla 40: Tarea Análisis de los datos de los usuarios.

Tarea: Análisis de los datos de los usuarios.	
Número de HU: 4	Número de Tarea: 1
Tipo de Tarea: Investigación	Puntos Estimados: 0.2
Fecha Inicio: 02/05/2014	Fecha Fin: 04/05/2014
Descripción: Se analizan y seleccionan los datos acerca de los usuarios que debe abarcar la aplicación.	

Tabla 41: Tarea Creación del soporte para la información de los usuarios.

Tarea: Creación del soporte para la información de los usuarios.	
Número de HU: 4	Número de Tarea: 2

Tipo de Tarea: Implementación	Puntos Estimados: 0.3
Fecha Inicio: 04/05/2014	Fecha Fin: 06/05/2014
Descripción: Definir y crear las entidades para dar soporte a la información de los usuarios.	

Tabla 42: Tarea Implementación de la lógica y el visual.

Tarea: Implementación de la lógica y el visual.	
Número de HU: 4	Número de Tarea: 3
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 06/05/2014	Fecha Fin: 09/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para registrar usuarios.	

HU-Gestionar rol:

Tabla 43: Tarea Implementación de la funcionalidad gestionar rol.

Tarea: Implementación de la funcionalidad gestionar rol.	
Número de HU: 5	Número de Tarea: 1
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 12/05/2014	Fecha Fin: 15/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para gestionar el rol de los usuarios.	

HU-Listar usuarios:

Tabla 44: Tarea Implementación de la funcionalidad de listar usuarios.

Tarea: Implementación de la funcionalidad de listar usuarios.	
Número de HU: 8	Número de Tarea: 1
Tipo de Tarea: Implementación	Puntos Estimados: 0.5
Fecha Inicio: 15/05/2014	Fecha Fin: 17/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para listar los usuarios.	

3.1.4 Iteración 3

Tareas por HU realizadas en la tercera iteración:

HU-Búsqueda y filtrado:

Tabla 45: Tarea Análisis de datos para búsqueda y filtrado.

Tarea: Análisis de datos para búsqueda y filtrado.	
Número de HU: 9	Número de Tarea: 1
Tipo de Tarea: Investigación	Puntos Estimados: 0.1
Fecha Inicio: 19/05/2014	Fecha Fin: 20/50/2014
Descripción: Se analizan y seleccionan los datos o atributos de los contenidos para realizar la búsqueda y el filtrado.	

Tabla 46: Tarea Implementación de la búsqueda y filtrado de audios.

Tarea: Implementación de la búsqueda y filtrado de audios.	
Número de HU: 9	Número de Tarea: 2
Tipo de Tarea: Investigación	Puntos Estimados: 0.3
Fecha Inicio: 20/05/2014	Fecha Fin: 22/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para la búsqueda y filtrado de audios.	

Tabla 47: Tarea Implementación de la búsqueda y filtrado de objetos 3D.

Tarea: Implementación de la búsqueda y filtrado de objetos 3D.	
Número de HU: 9	Número de Tarea: 2
Tipo de Tarea: Investigación	Puntos Estimados: 0.3
Fecha Inicio: 22/05/2014	Fecha Fin: 24/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para la búsqueda y filtrado de objetos 3D.	

Tabla 48: Tarea Implementación de la búsqueda y filtrado de usuarios.

Tarea: Implementación de la búsqueda y filtrado de usuarios.	
Número de HU: 9	Número de Tarea: 2
Tipo de Tarea: Investigación	Puntos Estimados: 0.3
Fecha Inicio: 22/05/2014	Fecha Fin: 24/05/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para la búsqueda y	

filtrado de usuarios.

HU-Visualizar objeto 3D:

Tabla 49: Tarea Implementación de la visualización de un objeto 3D.

Tarea: Implementación de la visualización de un objeto 3D.	
Número de HU: 10	Número de Tarea: 1
Tipo de Tarea: Investigación	Puntos Estimados: 1
Fecha Inicio: 26/05/2014	Fecha Fin: 03/06/2014
Descripción: Se crean las funciones necesarias en el controlador y las plantillas de la vista para la visualización de un objeto 3D en la web.	

3.2 Modelo de despliegue

El modelo de despliegue se utiliza para describir la topología o estructura del sistema con relación al hardware y el software. Define la configuración de los elementos de procesamiento en tiempo de ejecución y los componentes de software (35) (47). En la Figura 7 se muestra el Modelo de Despliegue para la aplicación desarrollada.

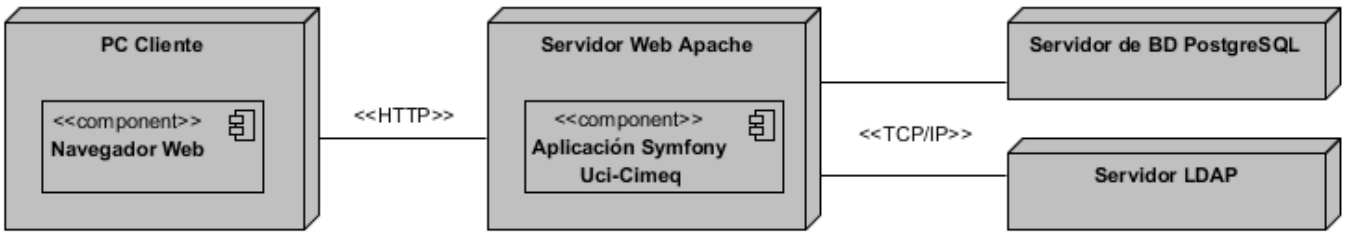


Figura 7: Modelo de despliegue.

A continuación se describe el modelo de despliegue, explicando que representa cada uno de sus nodos y su función.

Nodo PC Cliente: Este nodo representa la maquina desde donde los usuarios acceden al sistema mediante un navegador Web el cual se comunica mediante el protocolo HTTP para realizar peticiones y recibir las repuestas.

Nodo Servidor Web Apache: Este nodo representa el servidor donde se encuentra alojado el sistema que se encarga de recibir las peticiones del usuario mediante el protocolo HTTP y realizar las operaciones

necesarias para proveer las respuestas al usuario. Dentro de estas operaciones se encuentra las consultas a la base de datos, mediante el protocolo TCP/IP, para consultar o persistir los datos que maneja la aplicación. Además se comunica con el servidor LDAP del dominio también empleando TCP/IP, para establecer conexiones para la autenticación de usuarios y solicitar información referente a estos.

Nodo de BD PostgreSQL: Representa el servidor donde se encuentra alojada la base de datos del sistema la cual conserva toda la información necesaria.

Nodo Servidor LDAP: Representa el servidor LDAP del dominio UCI.CU.

3.3 Pruebas

Uno de los puntos más importantes durante el proceso de desarrollo son las pruebas, las cuales se realizan continuamente. Existen diferentes niveles, tipos y métodos de pruebas. Para guiar estas pruebas se diseñan casos de prueba. Los niveles se aplican persiguiendo diferentes objetivos, entre ellos están las pruebas de desarrollador, independiente, de unidad, de integración, de sistema y de aceptación. Los tipos de pruebas son varios, pueden ser de funcionalidad, usabilidad, fiabilidad, rendimiento y soportabilidad. Los métodos pueden ser pruebas de caja negra y pruebas de caja blanca (36).

Para la validación de la aplicación desarrollada se optó por el empleo de pruebas propuestas por la metodología XP:

Nivel de pruebas: Pruebas unitarias y pruebas de aceptación.

Tipo de prueba: Pruebas de funcionalidad.

Método de prueba: Pruebas de caja negra.

3.3.1 Pruebas Unitarias

Las pruebas unitarias deben diseñarse antes de comenzar con la implementación de cada funcionalidad, de manera que se definan los posibles casos en los que el código pueda fallar y una vez que se termina con la implementación se corren dichas pruebas para verificar que funcione correctamente el código. Estas pruebas son implementadas y realizadas por el programador para lograr que estas sean rápidas y efectivas de modo que se puedan realizar a diario de modo que se proporcione una indicación continua del progreso o se alerte tempranamente al equipo si algo sale mal (36).

3.3.2 Pruebas de Aceptación

Una prueba de aceptación es un escenario de utilización del sistema y el comportamiento que de él se espera, visto desde la perspectiva del cliente, usuario o sistema externo que interactúa con el programa, por lo que deben ser definidas por el cliente y verificarse por este en conjunto con los miembros del equipo de desarrollo (36). Estas constituyen el criterio de éxito en cuanto a la implementación de las historias de usuario.

A continuación se describen los casos de pruebas de aceptación realizados para cada HU.

3.3.3 Iteración 1

HU-Gestionar audio:

Tabla 50: Caso de prueba PA1-HU1.

Caso de Prueba de Aceptación	
Código:PA1-HU1	Historia de Usuario: Gestionar audio
Nombre: Subir audio.	
Descripción: Prueba para la funcionalidad de subir un archivo de audio al sistema.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Miembro o Administrador.	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Subir audio. 2. El usuario llena los campos del formulario que le muestra la aplicación. 3. El usuario da clic en el botón enviar.	
Resultado esperado: Se agrega un nuevo archivo de audio en el sistema y se guardan sus datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria. Ver figura 8.	

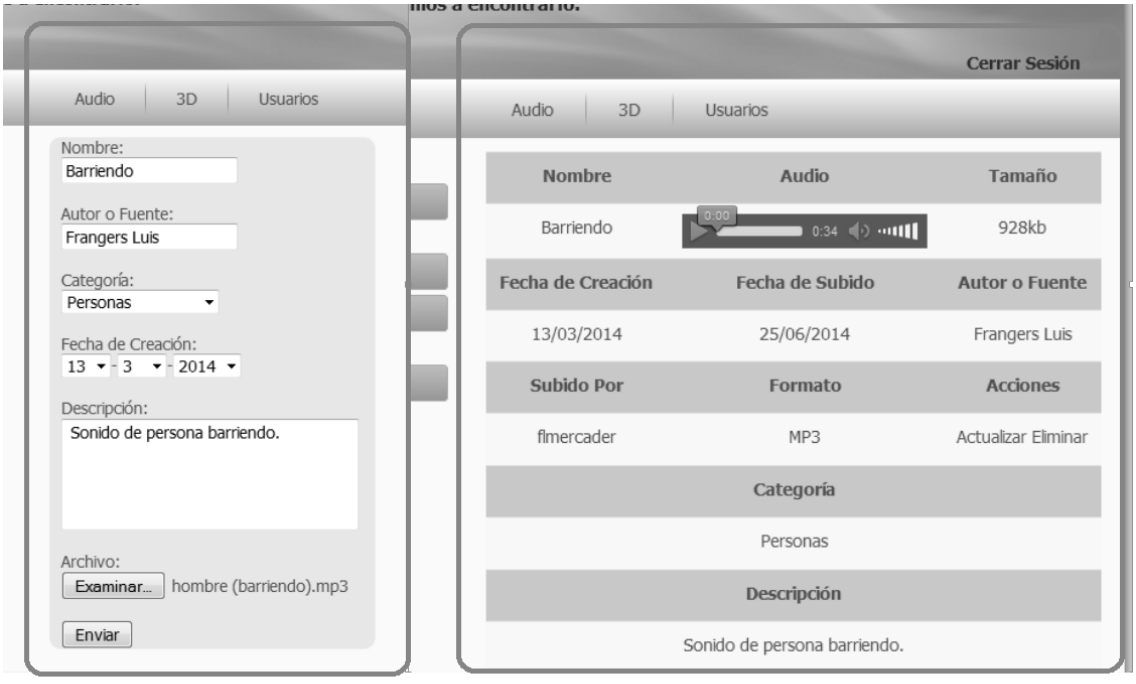
Tabla 51: Caso de prueba PA2-HU1.

Caso de Prueba de Aceptación	
Código:PA2-HU1	Historia de Usuario: Gestionar audio
Nombre: Actualizar audio.	
Descripción: Prueba para la funcionalidad de actualizar los datos de un archivo de audio.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador o ser miembro y ser el que lo haya subido.	
Entrada/Pasos de ejecución: 1. El usuario busca el audio a actualizar. 2. El usuario da clic en la opción Actualizar. 3. El usuario llena los campos del formulario que le muestra la aplicación.	

4. El usuario da clic en el botón actualizar.
Resultado esperado: Se actualizan los datos del archivo de audio con la información especificada por el usuario.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 51: Caso de prueba PA3-HU1.

Caso de Prueba de Aceptación	
Código:PA3-HU1	Historia de Usuario: Gestionar audio
Nombre: Eliminar audio.	
Descripción: Prueba para la funcionalidad de eliminar un archivo de audio.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador.	
Entrada/Pasos de ejecución: 1. El usuario busca el objeto a eliminar. 2. El usuario da clic en la opción Eliminar.	
Resultado esperado: Se eliminan el archivo de audio y sus datos correspondientes.	
Evaluación de la prueba: Prueba satisfactoria.	



Formulario ``Subir audio´´ Visualización del audio subido

Figura 8: Subir y visualizar audio.

HU-Gestionar objeto:

Tabla 52: Caso de prueba PA1-HU2.

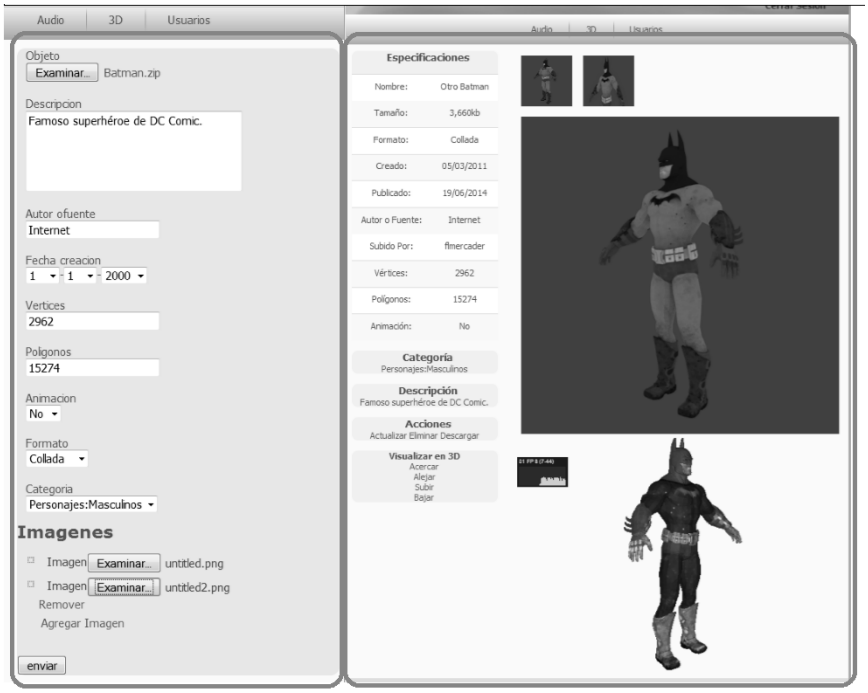
Caso de Prueba de Aceptación	
Código:PA1-HU2	Historia de Usuario: Gestionar objeto 3D
Nombre: Subir objeto 3D.	
Descripción: Prueba para la funcionalidad de subir un archivo de objeto 3D al sistema.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Miembro o Administrador.	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Subir objeto 3D. 2. El usuario llena los campos del formulario que le muestra la aplicación. 3. El usuario da clic en el botón enviar.	
Resultado esperado: Se agrega un nuevo archivo de objeto 3D en el sistema y se guardan su datos correctamente.	
Evaluación de la prueba: Prueba satisfactoria. Ver Figura 9.	

Tabla 52: Caso de prueba PA2-HU2.

Caso de Prueba de Aceptación	
Código:PA2-HU2	Historia de Usuario: Gestionar objeto 3D
Nombre: Actualizar objeto 3D.	
Descripción: Prueba para la funcionalidad de actualizar los datos de un objeto 3D.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador o miembro y debe ser el que subió previamente el objeto 3D que se va a actualizar.	
Entrada/Pasos de ejecución: 1. El usuario busca el objeto 3D a actualizar. 2. El usuario da clic en la opción Actualizar. 3. El usuario llena los campos del formulario que le muestra la aplicación. 4. El usuario da clic en el botón Actualizar.	
Resultado esperado: Se actualizan los datos del archivo de objeto 3D con la información especificada por el usuario.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 53: Caso de prueba PA3-HU2.

Caso de Prueba de Aceptación	
Código:PA3-HU2	Historia de Usuario: Gestionar objeto 3D
Nombre: Eliminar objeto 3D.	
Descripción: Prueba para la funcionalidad de eliminar un archivo de objeto 3D.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador.	
Entrada/Pasos de ejecución: 1. El usuario busca el objeto 3D a eliminar. 2. El usuario da clic en la opción Eliminar. 3. El usuario da clic en el botón Aceptar del cuadro de confirmación.	
Resultado esperado: Se elimina el archivo de objeto 3D y sus datos correspondientes.	
Evaluación de la prueba: Prueba satisfactoria.	



Formulario subir objeto 3D Visualización del objeto subido

Figura 9: Subir y visualizar objeto 3D.

HU-Listar audio:

Tabla 54: Caso de prueba PA1-HU6.

Caso de Prueba de Aceptación

Código:PA1-HU6	Historia de Usuario: Listar audios
Nombre: Listar audios.	
Descripción: Prueba para la funcionalidad de listar los audios del sistema.	
Condición de ejecución: Deben existir archivos de audio en el sistema.	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Audio situado en la barra de navegación.	
Resultado esperado: Se muestra en forma de lista todos los archivos de audio.	
Evaluación de la prueba: Prueba satisfactoria. Ver Figura 10	

Figura 10: Caso de prueba Listar audios.

HU-Listar objetos 3D:

Tabla 55: Caso de prueba PA1-HU7.

Caso de Prueba de Aceptación	
Código:PA1-HU7	Historia de Usuario: Listar objetos 3D
Nombre: Listar objetos 3D.	

Descripción: Prueba para la funcionalidad de listar los objetos 3D de sistema.
Condición de ejecución: Deben existir archivos de audio en el sistema.
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción 3D situado en la barra de navegación.
Resultado esperado: Se muestra en forma de lista todos los archivos de objetos 3D.
Evaluación de la prueba: Prueba satisfactoria.

HU-Visualizar contenido:

Tabla 56: Caso de prueba PA1-HU3.

Caso de Prueba de Aceptación	
Código:PA1-HU3	Historia de Usuario: Visualizar contenido
Nombre: Visualizar objeto 3D.	
Descripción: Prueba para la funcionalidad de visualizar todos los datos de un objeto 3D y sus imágenes.	
Condición de ejecución: Deben existir objetos 3D registrados en el sistema.	
Entrada/Pasos de ejecución: 1. El usuario debe buscar el objeto 3D a visualizar. 2. El usuario da clic en la opción Detalles.	
Resultado esperado: Se muestran todos los datos de un archivo 3D y las imágenes de este.	
Evaluación de la prueba: Prueba satisfactoria. Ver Figura 9.	

Tabla 57: Caso de prueba PA2-HU3.

Caso de Prueba de Aceptación	
Código:PA2-HU3	Historia de Usuario: Visualizar contenido
Nombre: Visualizar audio.	
Descripción: Prueba para la funcionalidad de visualizar todos los datos de un audio.	
Condición de ejecución: Deben existir audios registrados en el sistema.	
Entrada/Pasos de ejecución: 1. El usuario debe buscar el audio a visualizar. 2. El usuario da clic en la opción Detalles.	
Resultado esperado: Se muestran todos los datos de un archivo de audio y un reproductor para poder escucharlo.	
Evaluación de la prueba: Prueba satisfactoria. Ver Figura 8.	

3.3.4 Iteración 2

HU-Registrar usuario:

Tabla 58: Caso de prueba PA1-HU4.

Caso de Prueba de Aceptación	
Código:PA1-HU4	Historia de Usuario: Registrar usuario
Nombre: Registrar usuario.	
Descripción: Prueba para la funcionalidad de registrar un usuario en el sistema.	
Condición de ejecución: El usuario debe registrarse con su usuario y contraseña del dominio uci.cu.	
Entrada/Pasos de ejecución: 1. El usuario debe dar clic en la opción Registrarse. 2. El usuario debe ingresar su usuario y contraseña en el formulario mostrado. 3. El usuario debe dar clic en el botón Registrarse.	
Resultado esperado: Se comprueban si el usuario y contraseña son correctos y de ser así se registran los datos del dominio referentes al usuario, en caso contrario se muestra un mensaje de error.	
Evaluación de la prueba: Prueba satisfactoria. Ver Figura 11.	

Figura 11: Usuario registrado.

HU-Gestionar rol:

Tabla 59: Caso de prueba PA1-HU5.

Caso de Prueba de Aceptación	
Código:PA1-HU5	Historia de Usuario: Gestionar rol
Nombre: Gestionar rol.	
Descripción: Prueba para la funcionalidad de gestionar el rol de los usuarios del sistema.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador.	
Entrada/Pasos de ejecución: 1. Se busca el usuario al cual se le quiere cambiar el rol. 2. Se da clic en la opción Actualizar. 3. En el formulario mostrado se selecciona el nuevo rol del usuario. 4. Finalmente se da clic en el botón Actualizar.	
Resultado esperado: Se remplaza el rol del usuario especificado con el nuevo rol seleccionado.	
Evaluación de la prueba: Prueba satisfactoria.	

HU-Listar usuarios:

Tabla 60: Caso de prueba PA1-HU8.

Caso de Prueba de Aceptación	
Código:PA1-HU8	Historia de Usuario: Listar usuarios
Nombre: Listar usuarios.	
Descripción: Prueba para la funcionalidad de listar los usuarios del sistema.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador.	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Usuarios situado en la barra de navegación.	
Resultado esperado: Se muestra en forma de lista todos los usuarios del sistema.	
Evaluación de la prueba: Prueba satisfactoria.	

3.3.5 Iteración 3

HU-Búsqueda y filtrado:

Tabla 61: Caso de prueba PA1-HU9.

Caso de Prueba de Aceptación	
Código:PA1-HU9	Historia de Usuario: Búsqueda y filtrado
Nombre: Buscar audios.	

Descripción: Prueba para la funcionalidad de búsqueda y filtrado de audios en el sistema.
Condición de ejecución:
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Audio situado en la barra de navegación. 2. Especificar las opciones de búsqueda y filtrado que se encuentran en la barra lateral. 3. Ingresar los valores en los campos de búsqueda especificados. 4. Por último dar clic la opción Buscar.
Resultado esperado: Se muestran todos los audios que cumplen con los criterios de búsqueda especificados.
Evaluación de la prueba: Prueba satisfactoria.

Tabla 62: Caso de prueba PA2-HU9.

Caso de Prueba de Aceptación	
Código:PA2-HU9	Historia de Usuario: Búsqueda y filtrado
Nombre: Buscar objetos 3D.	
Descripción: Prueba para la funcionalidad de búsqueda y filtrado de objetos 3D en el sistema.	
Condición de ejecución:	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Audio situado en la barra de navegación. 2. Especificar las opciones de búsqueda y filtrado que se encuentran en la barra lateral. 3. Ingresar los valores en los campos de búsqueda especificados. 4. Por último dar clic la opción Buscar.	
Resultado esperado: Se muestran todos los objetos 3D que cumplen con los criterios de búsqueda especificados.	
Evaluación de la prueba: Prueba satisfactoria.	

Tabla 63: Caso de prueba PA3-HU9.

Caso de Prueba de Aceptación	
Código:PA3-HU9	Historia de Usuario: Búsqueda y filtrado
Nombre: Buscar usuarios.	
Descripción: Prueba para la funcionalidad de búsqueda y filtrado de usuarios del sistema.	
Condición de ejecución: El usuario debe estar autenticado con el rol de Administrador.	
Entrada/Pasos de ejecución: 1. El usuario da clic en la opción Usuarios situado en la barra de navegación. 2. Especificar las opciones de búsqueda y filtrado que se encuentran en la barra lateral. 3. Ingresar los valores en los campos de búsqueda especificados. 4. Por último dar clic la opción Buscar.	
Resultado esperado: Se muestran todos los usuarios que cumplen con los criterios de búsqueda especificados.	
Evaluación de la prueba: Prueba satisfactoria.	

HU-Visualizar objeto 3D:

Tabla 64: Caso de prueba PA1-HU10.

Caso de Prueba de Aceptación	
Código:PA1-HU10	Historia de Usuario: Visualizar objeto 3D
Nombre: Visualizar objeto 3D.	
Descripción: Prueba para la funcionalidad de visualizar un objeto 3d en la Web.	
Condición de ejecución: El objeto a visualizar debe estar en formato collada (.dae).	
Entrada/Pasos de ejecución: 1. El usuario debe buscar el objeto a visualizar. 2. Dar clic en la opción Detalles. 3. Por último dar clic en la opción Visualizar.	
Resultado esperado: Se renderiza el objeto 3D en la web.	
Evaluación de la prueba: Prueba satisfactoria. Ver figura 9.	

3.3.6 Resultado de las pruebas

Se llevaron a cabo tres iteraciones de pruebas de aceptación y se aplicaron un total de 17 casos de pruebas. En la primera iteración se detectaron 12 no conformidades (NC), de ellas 5 de código, 3 de interfaz y 4 de ortografía. En la segunda iteración se detectaron 6 NC, de ellas 2 de código, 3 de interfaz y 1 de ortografía. Y se realizó una tercera iteración en la cual no se encontraron nuevas NC y se verificó que las NC de las anteriores iteraciones estuvieran resueltas.

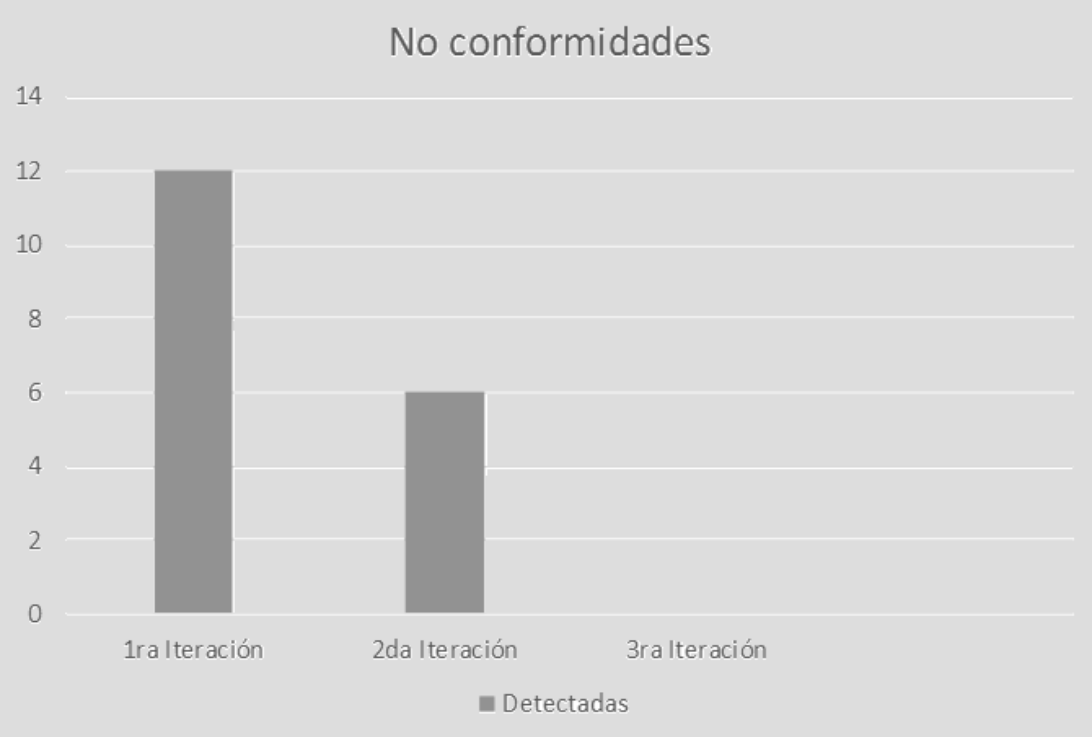


Figura 12: Gráfica de iteraciones de pruebas de aceptación.

Conclusiones del capítulo.

A finalizar las fases de implementación y prueba a modo de conclusión se plantea que:

- Para dar cumplimiento a las historias de usuarios planteadas por el cliente se realizó un total de 17 tareas de ingeniería.
- Se comprobó el correcto funcionamiento y la aceptación del cliente a partir de pruebas unitarias y pruebas de aceptación que permitieron detectar errores y corregirlos posteriormente.

Conclusiones Generales

En consecuencia con el trabajo realizado, se definen como cumplido los objetivos propuestos al comienzo de la investigación arribando a las siguientes conclusiones:

- El análisis de documentación especializada sobre las metodologías, principios y aplicaciones relacionadas con la gestión de componentes gráficos y sonoros permitió definir las responsabilidades generales a cumplir por la aplicación.

- Durante las fases de exploración y planificación se obtuvieron los artefactos necesarios para guiar y facilitar el proceso de desarrollo de la aplicación.

- La implementación de las funcionalidades de gestión para los componentes gráficos y sonoros del proyecto permitió obtener una aplicación que responde a las necesidades del cliente que se definieron en la fase de exploración.

- Las pruebas diseñadas y aplicadas garantizaron el correcto funcionamiento del sistema desarrollado, detectando no conformidades que posteriormente fueron solucionadas.

Recomendaciones

- Continuar extendiendo la colección de componentes a gestionar por medio del gestor de contenidos a los scripts de programación generados con los motores gráficos Unity 3D y Shiva 3D.

- Continuar perfeccionando el esquema de información de la aplicación, incorporando los metadatos propios incluidos en los formatos de los componentes obtenidos con cada una de las herramientas adoptadas por el proyecto (.blend, .dae, .omg, .wav).

Referencias

1. Portal de la Universidad de las Ciencias Informáticas . [En línea] [Citado el: 14 de 03 de 2014.] http://www.uci.cu/?q=mision.

2. GARCÍA, N. E. y JAROSZCZUK. *Objetos digitales: una experiencia de representación con metadatos.* s.l. : Dublin Core, 2008. ISBN 978-987-9350-84-3.

3. TRAMULLAS, J. *Gestión de contenidos 2.0.* 2008. ISSN 1886-6344.

4. GONZÁLEZ, S. D. *El gestor de contenidos en la sociedad de la información.* Univ. Salamanca de España : s.n., 2010. pág. 59.

5. Brun, Ricardo Eíto. hypertext.net. [En línea] 01 de 2009. http://www.hipertext.net/web/pag256.htm#Definición%20de%20gestión%20de%20contenidos..

6. *Modelo de gestión de objetos digitales para la gestión de soluciones tecnológicas.* Dayni Pérez-HernándezI, Martha Dunia Delgado-DapenaII. 1, La Habana, Cuba : s.n., 2013, Vol. 34. ISSN 1815-5936.

7. TurboSquid. [En línea] [Citado el: 20 de 03 de 2014.] http://www.turbosquid.com.search/robot.

8. SoundSnap. [En línea] [Citado el: 20 de 3 de 2014.] http://www.soundsnap.com/browse.

9. John Freddy Vega, Christian Van Der Henst. Guía HTML5. El presente de la web. HTML5, css3 y javascript. [En línea] 1 de junio de 2011. [Citado el: 10 de 03 de 2014.] http://mlw.io/guia-html5/.

10. Pérez, Javier Eguíluz. www.librosweb.es. [En línea] 25 de marzo de 2009. [Citado el: 11 de marzo de 2014.] http://www.librosweb.es/javascript.

11. librosweb.es. [En línea] [Citado el: 04 de 03 de 2014.] http://librosweb.es/javascript/capitulo_1.html.

12. Mateu, Carles. *Desarrollo de aplicaciones web.* Barcelona : Eureca Media, SL, 2004. ISBN: 84-9788-118-4.

13. ecured.cu. [En línea] 05 de 03 de 2014. http://www.ecured.cu/index.php/PHP.

14. ecured.cu. [En línea] [Citado el: 05 de 03 de 2014.] http://www.ecured.cu/index.php/JSP.

15. oracle.com. [En línea] [Citado el: 05 de 03 de 2014.] http://www.oracle.com/technetwork/java/javaee/jsp/index.html.

16. Alvarez, Miguel Angel. DesarrolloWeb.com. [En línea] [Citado el: 12 de mayo de 2014.] http://www.desarrolloweb.com/manuales/manual-jquery.html.

17. Jesús González Barahona, Joaquín Seoane Pascual, Gregorio Robles. *Introducción al software libre.* Madrid : s.n., 2003.

18. Potencier, Fabien. *Manual de Symfony2.* 2011. pág. 633.

19. GitHub. [En línea] [Citado el: 21 de 03 de 2014.] https://github.com/.

20. symfony.es. [En línea] [Citado el: 06 de 03 de 2014.] http://symfony.es/que-es-symfony.

21. framework.zend.com. [En línea] [Citado el: 06 de 03 de 2014.] http://framework.zend.com/about/.

22. ecured.cu. [En línea] [Citado el: 10 de 03 de 2014.] http://www.ecured.cu/index.php/Servidor_web.

23. news.netcraft.com. [En línea] [Citado el: 10 de 03 de 2014.] http://news.netcraft.com/archives/2014/05/07/may-2014-web-server-survey.html.

24. Drupal. [En línea] [Citado el: 20 de 03 de 2014.] http://drupal.org/node/3854.

25. EcuRed. [En línea] [Citado el: 21 de 03 de 2014.] http://www.ecured.cu/index.php/IIS.

26. Windows Server. [En línea] 2 de 2012. [Citado el: 25 de 06 de 2014.] http://technet.microsoft.com/es-es/library/hh831725.aspx.

27. GallegoVázquez, José Antonio. *Desarrollo Web con PHP y MySQL.* Madrid : s.n., 2003. ISBN: 84-415-1525-5..

28. Pecos, Daniel. Danielpecos.com. [En línea] 04 de 12 de 2010. [Citado el: 22 de 03 de 2014.] http://danielpecos.com/docs/mysql_postgres/x57.html.

29. postgresql.org. [En línea] [Citado el: 10 de 03 de 2014.] http://www.postgresql.org/about/.

30. netbeans.org. [En línea] [Citado el: 12 de 03 de 2014.] https://netbeans.org/features/index.html#.

31. netbeans.org. [En línea] [Citado el: 12 de 3 de 2014.] https://netbeans.org/features/php/index.html.

32. slideshare.net. [En línea] [Citado el: 12 de 03 de 2014.] http://www.slideshare.net/MagaLasic/presentacion-eclipse-grupo-6.

33. Rodrigo Pérez González, Isaías Carrillo Pérez, Aureliano David Rodríguez Martín. *Metodología de Desarrollo del Software.* 2008.

34. Menéndez, Rafael. Informática Aplicada a la Gestión Pública. s.l. : Departamento Informática y Sistemas. Universidad de Murcia, 13 de 10 de 2011.

35. Jacobson, Ivar, Booch, Grady y Rumbaugh, James. *El Proceso Unificado de Desarrollo de Software.* Madrid : s.n., 2000. 84-7829-036-2.

36. Pressman, Roger. *Ingeniería del Software: Un Enfoque Práctico.* s.l. : McGraw-Hill, 2005.

37. Diego Cantor, Brandon Jones. *WebGL Beginner's Guide.* s.l. : Packt Publishing Ltd, 2012. ISBN 978-1-84969-172-7.

38. AWAY3D. [En línea] 2010. [Citado el: 24 de 03 de 2014.] http:www.away3d.com/documentation.

39. Eguilus, Javier. *Desarrollo web ágil con Symfony 2.3.* [En línea] [Citado el: 18 de 03 de 2014.] http://sunshine.prod.uci.cu/search/Desarrollo%20web%20%C3%A1gil%20con%20Symfony%202.3.

40. [En línea] 18 de 03 de 2014. http://programacionextrema.tripod.com/fases.htm.

41. Joskowicz, José. *Reglas y Prácticas en eXtreme Programming.* 2008.

42. Agut, Raúl Monferrer. *Especificación de Requisitos Software según el estándar IEEE 830*.[En línea] http://www.google.com.cu/url?sa=t&rct=j&q=Especificaci%C3%B3n+de+Requisitos+Software+seg%C3%BAn+el+est%C3%A1ndar+IEEE+830&source=web&cd=2&ved=0CFUQFjAB&url=http%3A%2F%2Fsiml.googlecode.com%2Ffiles%2FERS.pdf&ei=40vET_qWN8zogAfGybS_CQ&usg=AFQjCNFj2Xj8OMaKkue4Qcg5SHl5w4TYUg&cad=rja

43. procesosoftware.wikispaces.com. [En línea] [Citado el: 19 de 03 de 2014.] http://procesosoftware.wikispaces.com/M%C3%A9todo+%C3%81gil+XP#Planificacion.

44. Joskowicz, José. [En línea] http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf.

45. Luis, Pedro. borrowbits.com. [En línea] http://borrowbits.com/2013/04/que-es-como-funciona-symfony2-conceptos-claves/.

46. Larman, Craig. *UML y Patrones.* s.l. : Prentice Hall.

47. sparxsystems. [En línea] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.