

Universidad de las Ciencias Informáticas

Facultad 3



Portal web taller literario

“Gertrudis Gómez de Avellaneda”

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

Autora:

Leanna Laura Amador Rios

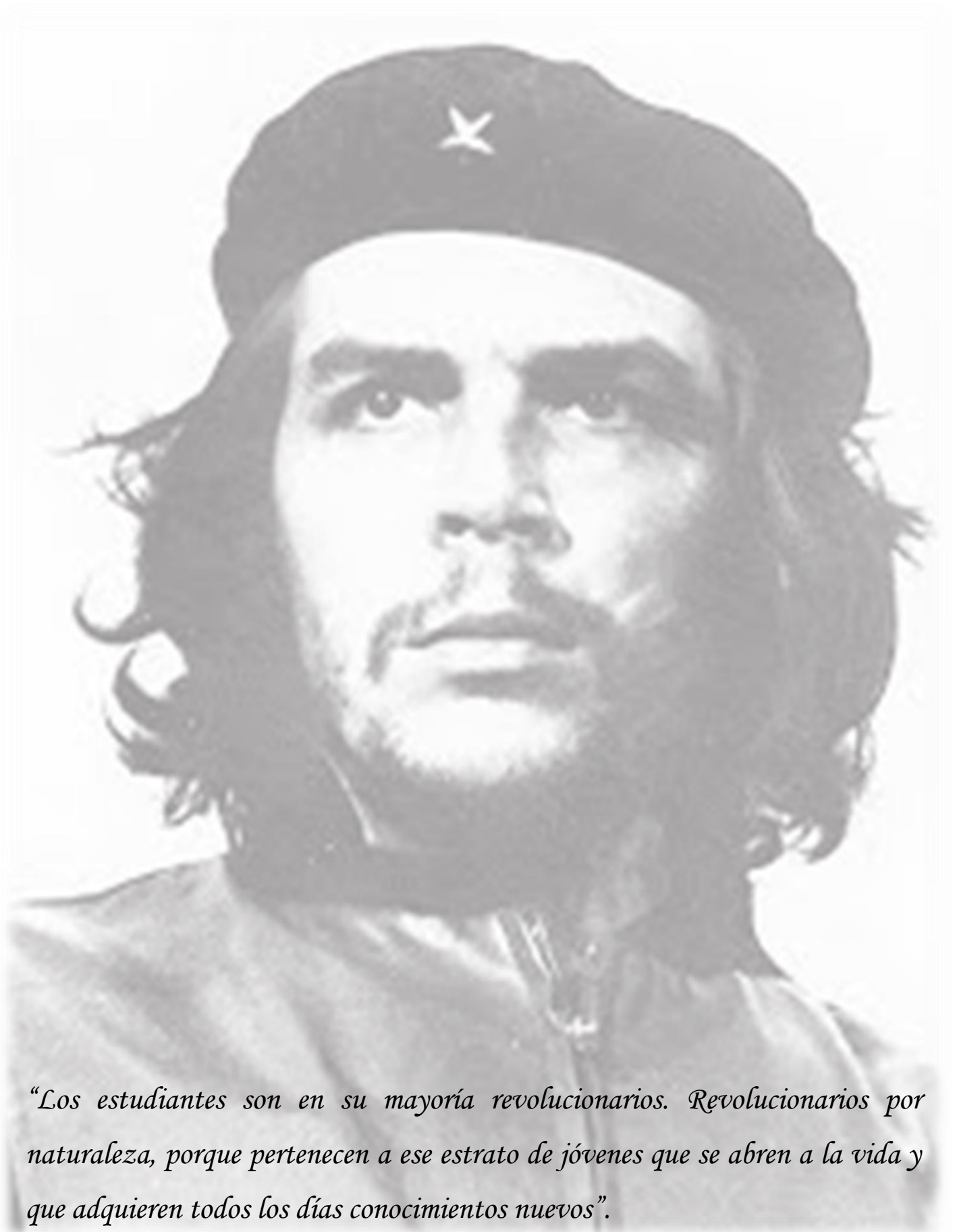
Tutor:

Ing. Abel Andres Irsula Tumbarell

Co-Tutora:

MSc. Susel García Cedeño

La Habana, Junio de 2015



“Los estudiantes son en su mayoría revolucionarios. Revolucionarios por naturaleza, porque pertenecen a ese estrato de jóvenes que se abren a la vida y que adquieren todos los días conocimientos nuevos”.

Ernesto Guevara de la Serna

Agradecimientos

A mi madre por existir y ser la mejor madre del mundo, por ayudarme y apoyarme siempre en todo y especialmente para que terminara mi carrera y llegar a ser la ingeniera que soy y así cumplir nuestro sueño.

A la revolución por darme la oportunidad de vivir esta experiencia que estoy segura será inolvidable.

A mi familia en su totalidad, por creer siempre en mí y no dudar ni por un segundo que este momento llegaría.

A mi novio, mi Leito por todo su apoyo y amor, que en muchas ocasiones me dio fuerza para seguir adelante.

A mis peques Alejandrino, Naty, Nally, Kika, Jordan, Maché, Alexia, Kuki y Kuka por ser mi mayor motivo de inspiración.

A mis abuelos Nancy y Amador por ayudarme y apoyarme en todo momento, son los mejores abuelos del mundo.

A mis tías Briagny y Sailín por su confianza, amor, dedicación y por estar siempre presentes para mí.

A mi tía Iliana, que aunque está muy lejos, siempre me ha querido tanto.

A mi tío Papito que siempre se ha sentido orgulloso de mí y siempre me recordó que yo podía alcanzar mis sueños.

A mi tía Galia, tía Ro, tío Pedri, tío Evelio y mi querido primo Leandri por todo el apoyo y ayuda incondicional, eso nunca lo olvidaré.

A Lilo, Alex y tía Bola, por ayudarme y creer siempre en mí.

A mi padre, que a pesar de la distancia y las circunstancias no dejó nunca de creer en mí.

A Gaby por nunca decirme que no, ser el mejor tío postizo del mundo y al igual que a Erick por darme unas de las alegrías más grandes de mi vida, mis pequeñas que son mis grandes inspiraciones.

A Maiquel por toda la ayuda que me has brindado, te agradezco enormemente todo el esfuerzo y sacrificio que junto a mi mami has hecho para que no me faltase nada.

A mis suegros Rosaida y el Chino y a mi cuñado Duni, por todo el apoyo y la ayuda, pero principalmente por aceptarme en su familia y hacerme sentir que soy parte de ella.

A mi tutor y cotutora por toda la ayuda y por regañarme cuando hizo falta.

Al tribunal, porque gracias a sus comentarios, observaciones y sugerencias esta tesis es lo que es.

A todos mis compañeros de brigada, incluyendo los que no pudieron terminar con nosotros, porque de cada uno de ellos aprendí algo nuevo.

A mis amigas Lore y Ana, porque con ustedes compartí momentos de felicidad, pero también de tristezas y siempre estuvieron ahí para mí sin importar qué.

A Yoandri García Palma, por toda la ayuda incondicional que me ha brindado.

A Rosalina Ibarra González, por ser tan buena persona, profe, bailarina y amiga, de la que aprendí mucho.

A Keylier, Jean Carlos, Jose Alberto y Yoandri por soportarme y aguantar mis malcriadeces.

A Claudina por todo su apoyo y a todo su equipo por ser tan chéveres y verme siempre como una loquita.

A todos los profes que de una forma u otra contribuyeron a mi formación.

A los profesores que me formaron para ingresar en la Universidad de las Ciencias Informáticas Rita, Julio y Raúl.



Este trabajo de diploma está dedicado con todo el amor del mundo a mi madre,

Blanca Rosa Rios Cala.

DECLARACIÓN JURADA DE AUTORÍA

Se declara que Leanna Laura Amador Rios es la única autora del trabajo de diploma Portal web taller literario “Gertrudis Gómez de Avellaneda” y se le reconoce a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste se firma la presente a los ____ días del mes de _____

del año 2015.

Firma de la autora

Leanna Laura Amador Rios

Firma del tutor

Ing. Abel Andres Irsula Tumbarell

Firma de la co-tutora

MSc. Susel García Cedeño

Datos del tutor:

Nombre: Ing. Abel Andres Irsula Tumbarell. Ingeniero en Ciencias Informáticas. 2 años de experiencia laboral. Diseñador de bases de datos del proyecto Sistema de Informatización de Gestión de las Fiscalías fase II. Tiene presentado un trabajo investigativo en la 9na Peña Tecnológica, ha cursado un postgrado de Gestión de Procesos en la escuela de invierno de la Universidad de la Ciencias Informáticas y otros postgrados incluidos como parte del conocimiento a tener para el despliegue nacional del sistema SIGEFII.

Correo electrónico: aairsula@uci.cu

Datos de la co-tutora:

Nombre: MSc. Susel García Cedeño. Licenciada en Psicología. 9 años de experiencia laboral. Jefa de Departamento de Gestión Extensionista. Diplomada en Gerencia Empresarial (2008), en Perfeccionamiento Empresarial (2008), Dirección de Organizaciones (2010), Enseñanza de la Educación Superior (2011), Sistema Diagnóstico Terapéutico de Edward Bach (2011); Master en Dirección (2012). Se ha desempeñado laboralmente como Psicóloga Organizacional, Especialista en Perfeccionamiento Empresarial y Metodóloga de Investigaciones y Metodóloga de Trabajo Educativo. Actualmente se desempeña como Profesor Asistente en la Facultad 2 impartiendo la asignatura de Metodología de la Investigación Científica. Ha participado en más de 10 eventos de carácter provincial, nacional e internacional, con investigaciones donde predominan estudios sobre formación de valores y gestión y desarrollo de habilidades directivas.

Correo electrónico: suselgc@uci.cu

Datos de la autora:

Nombre: Leanna Laura Amador Rios. Estudiante que opta por el título de Ingeniero en ciencias informáticas.

Correo electrónico: llamador@estudiantes.uci.cu

RESUMEN

Con el surgimiento del Centro Cultural Wilfredo Lam la creación de obras literarias aumentó significativamente en la Universidad de las Ciencias Informáticas, surgiendo el taller literario tradicional. Actualmente posee la necesidad de que sus miembros puedan gestionar sus obras literarias de forma personal, así como sus colecciones de obras y poder compartirlas con el profesor y demás miembros, mostrar información sobre eventos y noticias, entre otras actividades e informaciones relevantes para su funcionamiento. Estas acciones no se pueden realizar en el taller de forma tradicional pues el tiempo dedicado es solo de 3 horas semanales, lo cual es poco tiempo. También suelen perder textos revisados que pueden formar parte importante de una colección, esto se debe a que normalmente son almacenados en papeles sueltos.

El presente trabajo persigue como objetivo desarrollar un portal web para la gestión de la información asociada a los talleres literarios de manera que contribuya a elevar su disponibilidad. Para alcanzar este objetivo se investigaron las tendencias actuales para el desarrollo de portales web, se realizó un análisis de las herramientas, tecnologías, lenguajes y metodología a utilizar. Además se definen los artefactos generados a partir de la metodología Programación Extrema, siendo esta la seleccionada para guiar el proceso de desarrollo del software de manera ágil. Como resultado se obtuvo el Portal web que representa al taller literario de la Universidad de las Ciencias Informáticas.

Palabras claves: portal, información, literatura, TL-GGA, taller.

ÍNDICE

| | |
|--|----|
| INTRODUCCIÓN | 1 |
| Capítulo 1. Fundamentación Teórica | 4 |
| 1.1 Introducción..... | 4 |
| 1.2 Conceptos fundamentales..... | 4 |
| 1.2.1 La gestión de la información..... | 4 |
| 1.2.2 Disponibilidad..... | 4 |
| 1.2.3 Portal web | 5 |
| 1.3 Soluciones Similares | 6 |
| 1.4 Metodologías de desarrollo de software | 7 |
| 1.4.1 Metodologías ágiles | 8 |
| 1.4.1.1 Scrum..... | 8 |
| 1.4.1.2 Programación Extrema..... | 9 |
| 1.5 Análisis de las herramientas y técnicas | 10 |
| 1.5.1 Sistemas de Gestión de Contenidos (CMS) | 10 |
| 1.5.1.1 Tipos de CMS: | 10 |
| 1.5.1.2 Drupal 7.14 | 11 |
| 1.5.1.3 Joomla 2.5.4..... | 12 |
| 1.5.1.4 WordPress 4.2.2 | 12 |
| 1.6 Base de Datos..... | 14 |
| 1.6.1 MySQL 5.6.11 | 14 |
| 1.7 Lenguajes de programación | 14 |
| 1.7.1 Lenguaje de programación del lado del servidor | 15 |
| 1.7.1.1 PHP 5.5.3..... | 15 |
| 1.7.2 Lenguaje de programación del lado del cliente..... | 16 |
| 1.7.2.1 HTML 5 | 16 |
| 1.7.2.2 CSS 3..... | 16 |
| 1.7.2.3 Java Script 1.8 | 17 |
| 1.7.2.4 Frameworks de JavaScript JQuery 1.11.0..... | 17 |

| | | |
|---|--|-----------|
| 1.8 | Herramienta para el modelado Visual Parading for UML 8.0 | 17 |
| 1.8.1 | UML 2.0 | 18 |
| 1.9 | Servidores web | 18 |
| 1.9.1 | Apache 2.4.4..... | 18 |
| 1.10 | Apache JMeter 2.9 | 19 |
| 1.11 | Entorno de Desarrollo Integrado (IDE) PhpStorm 8.0..... | 19 |
| 1.12 | Métricas y pruebas para la verificación y la validación del sistema..... | 19 |
| 1.12.1 | Métricas | 19 |
| 1.12.2 | Pruebas..... | 20 |
| 1.12.2.1 | Pruebas funcionales..... | 20 |
| 1.12.2.2 | Pruebas de aceptación..... | 20 |
| 1.13 | Conclusiones parciales | 21 |
| Capítulo 2. Descripción del portal | | 22 |
| 2.1 | Introducción..... | 22 |
| 2.2 | Propuesta del sistema..... | 22 |
| 2.3 | Características no funcionales del sistema..... | 23 |
| 2.3.1 | Usabilidad | 23 |
| 2.3.2 | Funcionalidad..... | 23 |
| 2.3.3 | Mantenibilidad | 23 |
| 2.3.4 | Fiabilidad..... | 23 |
| 2.3.5 | Portabilidad | 24 |
| 2.3.6 | Eficiencia..... | 24 |
| 2.4 | Fase de exploración | 24 |
| 2.4.1 | Involucrados en el sistema | 24 |
| 2.4.2 | Historias de Usuario | 25 |
| 2.5 | Fase de estimación y planificación | 26 |
| 2.5.1 | Estimación de esfuerzo por historias de usuarios..... | 26 |
| 2.5.2 | Plan de iteraciones..... | 27 |
| 2.6 | Tarjetas CRC (Clase – Responsabilidad – Colaborador)..... | 28 |

| | | |
|---|---|-----------|
| 2.7 | Diagrama de despliegue | 28 |
| 2.8 | Mapa de navegación | 29 |
| 2.9 | Patrones de diseño | 30 |
| 2.9.1 | Patrones GRASP | 30 |
| 2.9.2 | Patrones GOF | 34 |
| 2.10 | Conclusiones parciales | 35 |
| Capítulo 3. Implementación y Prueba..... | | 36 |
| 3.1 | Introducción..... | 36 |
| 3.2 | Arquitectura..... | 36 |
| 3.2.1 | Arquitectura Cliente/Servidor..... | 37 |
| 3.3 | Análisis de extensiones existentes | 37 |
| 3.4 | Descripción de nuevos componentes | 38 |
| 3.5 | Temas en WordPress..... | 39 |
| 3.5.1 | Tema para el portal web..... | 39 |
| 3.6 | Seguridad del sistema..... | 39 |
| 3.6.1 | Extensiones para garantizar la seguridad del sistema..... | 40 |
| 3.7 | Fase de implementación | 40 |
| 3.8 | Métricas para requisitos | 41 |
| 3.8.1 | Estabilidad de los requisitos | 41 |
| 3.8.2 | Especificidad de los requisitos | 41 |
| 3.8.3 | Grado de validación de los requisitos | 42 |
| 3.9 | Métricas para diseño | 43 |
| 3.9.1 | Tamaño Operacional de Clase (TOC) | 43 |
| 3.9.2 | Relaciones entre Clases (RC) | 45 |
| 3.10 | Pruebas..... | 47 |
| 3.10.1 | Pruebas funcionales..... | 47 |
| 3.10.1.1 | Resultados de las pruebas funcionales | 49 |
| 3.10.2 | Pruebas de aceptación..... | 49 |
| 3.10.2.1 | Resultados de las pruebas de aceptación..... | 50 |

| | |
|--|-----------|
| 3.10.3 Pruebas de carga y estrés..... | 50 |
| 3.10.3.1 Resultados de las pruebas de carga y estrés | 51 |
| 3.11 Disponibilidad..... | 51 |
| 3.12 Conclusiones parciales | 52 |
| CONCLUSIONES GENERALES | 53 |
| RECOMENDACIONES..... | 54 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 55 |

ÍNDICE DE TABLAS

| | |
|---|----|
| Tabla 1 Comparación entre las metodologías ágiles y tradicionales | 7 |
| Tabla 2 Involucrados en el sistema..... | 24 |
| Tabla 3 HU Gestionar obra literaria personal | 25 |
| Tabla 4 Estimación de esfuerzo por historias de usuario | 26 |
| Tabla 5 Plan de iteraciones..... | 27 |
| Tabla 6 Tarjeta CRC de la clase Obra_TLGGA | 28 |
| Tabla 7 Tarea 1 de la historia de usuario: Gestionar obra literaria personal..... | 40 |
| Tabla 8 Rango de valores para la evaluación de los atributos de calidad | 43 |
| Tabla 9 Aplicación de la métrica TOC..... | 43 |
| Tabla 10 Resultados de la aplicación de la métrica TOC | 44 |
| Tabla 11 Rango de valores para la evaluación de los atributos de calidad | 45 |
| Tabla 12 Aplicación de la métrica RC | 46 |
| Tabla 13 Resultados de la aplicación de la métrica RC | 47 |
| Tabla 14 Caso de prueba SC Adicionar obra literaria personal..... | 48 |
| Tabla 15 Caso de prueba Gestionar obra literaria personal | 49 |

ÍNDICE DE FIGURAS

| | |
|---|----|
| Figura 1 Propuesta del sistema | 22 |
| Figura 2 Diagrama de despliegue | 29 |
| Figura 3 Mapa de navegación..... | 30 |
| Figura 4 Segmento de la clase user.php..... | 31 |
| Figura 5 Segmento de la clase post.php..... | 31 |
| Figura 6 Segmento de la clase comment.php | 32 |
| Figura 7 Segmento de la clase Obra_TLGGA_Controller | 32 |
| Figura 8 Segmento de la clase Member_TLGGA_Controller | 33 |
| Figura 9 Segmento de la clase functions.php | 33 |
| Figura 10 Segmento del archivo uci-ldap.php | 34 |
| Figura 11 Segmento del archivo category.php..... | 35 |
| Figura 12 Estructura de archivos de WordPress | 36 |
| Figura 13 Arquitectura Cliente- Servidor | 37 |
| Figura 14 Resultados de las pruebas funcionales..... | 49 |
| Figura 15 Resultados de las pruebas de aceptación..... | 50 |
| Figura 16 Resultados de las pruebas de carga y estrés..... | 51 |

INTRODUCCIÓN

Desde el surgimiento de la clase obrera, las casas de cultura o centros culturales jugaron un papel muy importante en la democratización y actualización de los valores culturales y como centro de confrontación de las ideas. Desde su creación tienen el objetivo de formar valores culturales en la sociedad, además se convirtieron en el centro irradiador de la cultura en los municipios, representando un apoyo especial para el desarrollo del Movimiento de Artistas Aficionados (MAA). En la actualidad existen más de 300 casas de cultura en el país y con su creación se logra una amplia incorporación de la población y en especial de los jóvenes en el ámbito cultural (1). Entre las actividades que ofrecen se encuentra el funcionamiento de talleres literarios.

Un taller literario es una vía tradicional que ofrecen las casas de cultura por la que se puede acceder a actividades de formación literaria e información, se revisan obras literarias, se comparte a criterio de profesores y otros integrantes, es un espacio para fomentar la creación de libros, colecciones que puedan ser presentadas en eventos de cualquier ámbito, están orientados al aprendizaje, la creación y el intercambio y no son unidireccionales ni meramente expositivos (2). Las personas van a un taller a compartir lo que escriben, a conocer a otras personas que escriben, a recibir comentarios y a comentar. Un taller literario debe ayudar a cada participante a forjar una voz propia y a ser escritor o escritora desde su contexto cotidiano. Los talleres literarios existentes en el país para los aficionados no son suficientes, la información que se suele buscar no siempre se puede conseguir en el momento porque el taller está sujeto a un horario fijo en la semana.

El mundo de hoy, está inmerso en una nueva revolución tecnológica basada en la informática, que encuentra su principal impulso en el acceso y en la capacidad de procesamiento de información sobre todos los temas y sectores de la actividad humana. Ha contribuido a que culturas y sociedades se transformen aceleradamente tanto económica, como social y políticamente, con el objetivo fundamental de alcanzar con plenitud sus potencialidades. La influencia de la informática en la vida se puede ver en el cine, la música y también en la literatura, que forman parte de nuestra cultura.

En la Universidad de las Ciencias Informáticas (UCI) se han implementado diferentes portales web que brindan soluciones para manejar información y crear un ambiente de divulgación de diferentes temas que son de interés de la comunidad universitaria. Aun en un entorno tan pequeño como este, está claramente expuesta la diversidad de procesos de desarrollo, tecnologías y herramientas para la creación de los mismos.

También se cuenta con un centro cultural llamado “Wilfredo Lam” el cual ofrece un taller literario tradicional cuyo funcionamiento no puede ofrecer una gestión rápida e interactiva de sus actividades, debido a:

1. La información que ofrece (noticias, eventos, técnicas literarias) no toda puede ser encontrada con prontitud.
2. El tiempo reducido de su funcionamiento impide que grandes cantidades de obras sean compartidas entre autor-profesor, autor y otros integrantes.
3. La creación de colecciones de novelas, poesías, ensayos y cuentos agrupadas en libros no se puede realizar con el seguimiento y ayuda de profesores del taller.
4. Los autores aficionados del taller que buscan crear diferentes grupos de colecciones de obras, suelen en muchos casos perder textos revisados en el taller, los cuales pueden convertirse en parte importante de una colección, esto se debe a que por lo general su almacenamiento es en papeles sueltos.

De la situación antes planteada se deriva al siguiente **problema a resolver**: ¿cómo gestionar los talleres literarios de manera que contribuya a elevar la disponibilidad de la información asociada a los mismos?

Enmarcado en el **objeto de estudio**: los talleres literarios en internet.

Del problema descrito anteriormente se identifica como **objetivo general**, desarrollar un portal web para la gestión de la información asociada a los talleres literarios de manera que contribuya a elevar su disponibilidad.

Centrado en el **campo de acción**: sistemas de gestión de contenidos (CMS) literarios en internet.

Para darle cumplimiento al objetivo general propuesto se han definido los siguientes **objetivos específicos**:

- Determinar los referentes teóricos, herramientas y tecnologías a utilizar.
- Realizar análisis y diseño del portal para definir los elementos a implementar.
- Implementar un portal web de información literaria con funcionalidades de un taller literario.
- Validar la implementación del portal web para elevar la disponibilidad de la información asociada a talleres literarios.

Para realizar la investigación se utilizaron los siguientes **métodos de investigación**:

Métodos teóricos:

- **Histórico-lógico:** se empleó para analizar la trayectoria y evolución de los talleres literarios en internet.
- **Analítico-sintético:** permitió el procesamiento de la información y arribar a las conclusiones prácticas y teóricas de la investigación, así como precisar las herramientas utilizadas para el diseño y la implementación del portal web.
- **Modelación:** se utilizó para la creación de abstracciones que explican la realidad, este método es un instrumento de la investigación que permite la creación de modelos, descubrir y estudiar nuevas relaciones y cualidades del objeto de estudio.

Métodos empíricos:

- **Entrevista:** permitió obtener información valiosa sobre las necesidades del cliente, para identificar los requisitos del portal web.

El presente trabajo estará conformado por la siguiente estructura: introducción, tres capítulos, conclusiones, recomendaciones, bibliografía general y anexos. Los capítulos abordan los siguientes temas:

Capítulo 1. Fundamentación Teórica. En este capítulo se refleja el estado del arte del tema en cuestión en el ámbito internacional y nacional, se explican los principales conceptos relacionados con el objeto de estudio, los elementos teóricos que constituyen la base de la investigación tales como el de portal web para la comprensión de la naturaleza específica del trabajo y el CMS como la principal herramienta utilizada para realizarlo. Se estudian y analizan el uso de las tecnologías, metodologías, técnicas y herramientas utilizadas en trabajos anteriores relacionados con el objeto de estudio para seleccionar las idóneas para la solución al problema en cuestión de la investigación.

Capítulo 2. Descripción del portal. Se presentan las características del portal web a desarrollar así como la explicación de cómo está estructurada la arquitectura del portal. Provee además un conjunto de artefactos obtenidos a raíz de la aplicación de la metodología de desarrollo seleccionada.

Capítulo 3. Implementación y prueba. Se realiza el plan de iteraciones y se expresan las tareas de programación implementando todas las funcionalidades identificadas. Finalmente se realizan las pruebas donde estas son detalladas con el objetivo de asegurar el correcto funcionamiento del sistema.

Capítulo 1. Fundamentación Teórica

1.1 Introducción

En la actualidad para una empresa es indispensable contar con un sitio web, ya que la red es el medio más utilizado por todos los usuarios para buscar información y llegar hasta las empresas a las cuales se les va a comprar un producto o contratar un servicio (3). Los portales web han tenido una significativa demanda, ya que brindan a un grupo de usuarios, acceso a diversa cantidad de recursos y servicios informáticos de forma integrada y sencilla, constituyen un instrumento primordial en la socialización de la información.

1.2 Conceptos fundamentales

1.2.1 La gestión de la información

La gestión de la información es definida por varios autores como el conjunto de actividades realizadas con el fin de controlar, almacenar y posteriormente, recuperar adecuadamente la información reproducida, recibida o retenida por cualquier organización en el desarrollo de sus actividades (4) (5).

Otros autores, teniendo en cuenta las nuevas tecnologías definen gestión de la información, como un proceso que incluye operaciones como extracción, manipulación, tratamiento, depuración, conservación, acceso y/o colaboración de la información adquirida por una organización a través de diferentes fuentes y que gestiona el acceso y el derecho de los usuarios sobre la misma (6) (7) (8).

Teniendo en cuenta las definiciones anteriores se puede decir que la gestión de la información consiste en el almacenamiento, recolección y manipulación de datos que puede llegar a incluir operaciones como organización, extracción, tratamiento, depuración y conservación de los mismos. Actualmente existen varios productos de software que permiten la gestión de la información, dentro de los cuales se encuentran los portales web.

1.2.2 Disponibilidad

La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos fuera de línea, *downtime* u *offline*),

sino incluso con la percepción de caída desde el punto de vista del usuario: cualquier circunstancia que nos impida trabajar productivamente con el sistema desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles, es considerada como un factor de baja disponibilidad (9).

1.2.3 Portal web

El término portal tiene como significado puerta grande. Un portal de Internet es un sitio web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, documentos, aplicaciones y compra electrónica. Un portal tiene como objetivo conseguir que los internautas, cuando accedan a internet lo hagan siempre a través de una web determinada (10).

Los portales web pueden ser clasificados en diferentes modalidades en dependencia del público al cual están dirigidos, los servicios que brindan y a los objetivos del propio portal. Dentro de la primera clasificación se encuentran los siguientes:

- **Portales horizontales o genéricos:** también conocidos como mega portales, cuyo contenido está dirigido a todo tipo de usuario, en el que se brinda cierta cantidad de información de forma estructurada, sin obligar a los usuarios a navegar por muchos sitios para obtenerla, en otras palabras diseñado para usuarios con poca experiencia en el internet (11). Incluyen servicios como: chats, comunidades virtuales, e-mail, foros de discusión, noticias e información (12).
- **Portales verticales o especializados:** es un sitio web que provee de información y servicios a un sector en particular, con contenidos concretos y centrados en un tema como puede ser un portal de música, empleo, inmobiliario, un portal de finanzas, arte o deportes (12). Acotar que en muchos de los casos el acceso no es gratuito y se trata de portales corporativos que dentro de una Intranet Corporativa, provee a sus miembros de información interna y externa relacionada con la propia entidad (11).
- **Portales diagonales:** también conocidos como híbridos, están formados por el portal horizontal y el vertical, abordando temas variados, de naturalezas diferentes, así como profundidad y alcance (11). Utilizan redes sociales o aplicaciones generalistas (12).

- **Portales móviles:** son sitios que permiten la conexión de los usuarios tanto desde Internet como de un teléfono móvil (12). Este es similar a un sitio web normal, pero diseñado para pantallas pequeñas, de tal forma que las imágenes y los contenidos se optimizan para que se ajusten a las pantallas de dispositivos móviles. Cada vez más se abren caminos a niveles empresariales para realizar negocios ofreciendo productos o servicios a los usuarios (11).

La diferencia radical entre los portales horizontales y verticales, es que los horizontales están buscando ofrecer sus servicios para los usuarios en general, sin importar cuál será el contenido, mientras que los verticales enfocan sus servicios para alguna institución determinada mediante la implementación de Intranets, Extranets o portales verticales (13).

El tipo a implementar es vertical porque se diseñará e implementará para una institución en específico, el centro cultural “Wilfredo Lam” de la UCI y específicamente para el taller literario de dicho centro.

1.3 Soluciones Similares

Estudios realizados develaron que en la actualidad existen en el mundo diversos procesos informatizados gracias al desarrollo y avance de las TIC, tal es el caso de los talleres literarios vía internet y de portales con funciones similares a la que se pretende alcanzar, lo cual es muy útil por la complejidad de funcionamiento de los talleres literarios tradicionales expuestos en la introducción del presente trabajo.

Algunos de los ejemplos son:

- Taller de escritura “El desván de la memoria” (ANEXO #1)
- Talleres de escritura creativa “Fuentetaja” (ANEXO #2)
- Taller literario “Palabras” (ANEXO #3)
- El taller de creación literaria “Escritores” (ANEXO #4)

En Cuba, los portales web también se emplean frecuentemente con el objetivo de divulgar las principales noticias y actividades que tienen lugar en la isla, o en una entidad en particular.

Ejemplo de estos portales es:

- La editorial electrónica “cubaLiteraria” (ANEXO #5)

Capítulo 1. Fundamentación Teórica

Entre las características comunes que presentan los portales web antes mencionados se encuentran que todos permiten a los inscritos publicar artículos, además tienen una especie de biblioteca o librería donde los usuarios pueden acceder a los artículos que hayan sido publicados por los inscritos, también cuentan con una sesión acerca de los escritores del taller y contienen vínculos a las redes sociales. La presente solución cuenta con una sesión donde los inscritos pueden visualizar consejos de literatura y podrán gestionar sus obras y colecciones literarias de forma personal.

1.4 Metodologías de desarrollo de software

Un proceso de software detallado y completo suele denominarse “metodología”. Las metodologías se basan en una combinación de los modelos de proceso genéricos (cascada, evolutivo, incremental, espiral) (14).

El análisis de la bibliografía consultada, muestra que existen dos tipos de metodologías: ágiles y tradicionales (14). En las metodologías tradicionales o pesadas están incluidas MSF (*Microsoft Solution Framework*), RUP (*Proceso Unificado de Desarrollo*) y sus híbridos; estas fueron diseñadas para grandes proyectos, se centran en la definición detallada de los procesos, tareas y herramientas a utilizar; y por esto generan una extensa documentación. Además, al no demandar un contacto sistemático con el cliente son descartadas en la presente investigación y se propone el empleo de una metodología ágil (15). A continuación se muestra una tabla comparativa de las metodologías ágiles y tradicionales (16).

Tabla 1 Comparación entre las metodologías ágiles y tradicionales

| Metodologías Ágiles | Metodologías Tradicionales |
|--|--|
| Basadas en heurísticas provenientes de prácticas de producción de código | Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo |
| Especialmente preparados para cambios durante el proyecto | Cierta resistencia a los cambios |
| Impuestas internamente (por el equipo de desarrollo) | Impuestas externamente |
| Proceso menos controlado, con pocos principios | Proceso mucho más controlado, con numerosas políticas/normas |
| No existe contrato tradicional o al menos es bastante flexible | Existe un contrato prefijado |

Capítulo 1. Fundamentación Teórica

| | |
|--|--|
| El cliente es parte del equipo de desarrollo | El cliente interactúa con el equipo de desarrollo mediante reuniones |
| Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos |
| Pocos artefactos | Más artefactos |
| Pocos roles | Más roles |
| Menos énfasis en la arquitectura del software | La arquitectura del software es esencial y se expresa mediante modelos |

1.4.1 Metodologías ágiles

Las metodologías ágiles son lo más flexibles posibles, puesto que son utilizadas en proyectos de corta duración y generan poca documentación (17). Dentro de esta clasificación se encuentran: SCRUM, Crystal, DSDM (Método de desarrollo de sistema dinámico) y Programación Extrema (XP) (18). En lo adelante se profundizará en las metodologías SCRUM y XP por ser metodologías ágiles que se basan en el trabajo orientado directamente al objetivo.

1.4.1.1 Scrum

Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos. El desarrollo de software se realiza mediante iteraciones, denominadas sprint, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La segunda característica importante son las reuniones a lo largo del proyecto. Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración (18).

Scrum reduce al máximo la burocracia y las actividades no orientadas a producir software. Se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad. Se basa en la adaptación continua a las circunstancias de la evolución del proyecto. Es un modo de desarrollo de carácter adaptable más que predictivo, orientado a las personas más que a los procesos, emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones. Gestiona su evolución a través de reuniones breves diarias en las que todo el equipo revisa el trabajo realizado el día anterior y el previsto para el día siguiente (19).

1.4.1.2 Programación Extrema

Programación Extrema se centra 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 (20). Se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (21). Se define para proyectos con requisitos imprecisos, muy cambiantes y donde existe un alto riesgo técnico. Considera que los cambios de requisitos es un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos (20).

Cree que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos (22). La Programación Extrema cuenta con las siguientes fases: planificación, diseño, codificación y pruebas. Parte fundamental de esta metodología es que no es necesario implementar todas y cada una de sus prácticas (23).

Como clave para lograr el éxito se centra en cuatro variables para guiar el desarrollo: costo, tiempo, calidad y alcance. Su base es la comunicación continua entre el cliente y el equipo de desarrollo, existiendo fluidez entre todos los participantes (24). Incluye 12 prácticas fundamentales destacando entregas pequeñas, refactorización constante, diseño simple y un desarrollo orientado a las pruebas (25).

Para la selección de la metodología de desarrollo a utilizar se compararon las metodologías antes expuestas arrojando como resultado, que se descarta Scrum porque sus iteraciones son más extensas que en XP, además cuando se finaliza una iteración, las tareas implementadas en ella y que el cliente ha mostrado su conformidad, no se vuelven a tocar más, por tanto, debido a que el equipo de desarrollo es pequeño, a que el cliente forma parte de dicho equipo, a los continuos cambios de los requisitos del sistema, a que se acordó con el cliente realizar pequeñas entregas y realizar pruebas en producción y además no contar con un contrato tradicional con requisitos de implementación bien definidos, se decidió utilizar como metodología para guiar el proceso de desarrollo la metodología ágil XP.

1.5 Análisis de las herramientas y técnicas

1.5.1 Sistemas de Gestión de Contenidos (CMS)

Los CMS se han convertido en una tendencia general adoptada por los desarrolladores de aplicaciones que manipulan contenido web (26). Estos sistemas de gestión de contenidos presentan ventajas que se explican a continuación:

- Generan, gestionan y publican la información con la máxima eficiencia y productividad y administran la información del sitio sin conocimiento alguno de HTML (por sus siglas en inglés *Hypertext Markup Lenguaje* o Lenguaje de Marcas de Hipertexto) (26).
- Permite la integración de contenidos de múltiples fuentes y formatos (textos, imágenes, audio, video) (27).
- Sencilla administración de la web o portal pues la curva de aprendizaje es extremadamente baja (28).
- Solución adaptable mediante módulos independientes y los numerosos componentes que pueden combinarse para obtener la solución específica para una necesidad concreta (28).

1.5.1.1 Tipos de CMS:

Existe una gran variedad de CMS, los cuales se pueden agrupar según el tipo de sitio que permiten gestionar, entre los fundamentales se encuentran (29):

- **Genéricos:** ofrecen la plataforma necesaria para desarrollar e implementar aplicaciones que den solución a necesidades específicas, para soluciones de comercio electrónico, blogs, portales.
- **Específicos para Organizaciones No Gubernamentales (ONG):** nacen para cubrir las necesidades de las ONG, ofreciendo una plataforma de servicios de Internet que en ocasiones incluye además del CMS herramientas para la recaudación de fondos.
- **Foros:** sitio que permite la discusión en línea donde los usuarios pueden reunirse y discutir temas en los que están interesados.

Capítulo 1. Fundamentación Teórica

- **Blogs:** sitio que permite la publicación de noticias o artículos en orden cronológico con espacio para comentarios y discusión.
- **Wikis:** sitio web dónde todos los usuarios pueden colaborar en los artículos, aportando información o reescribiéndola.
- **Publicaciones digitales:** son plataformas especialmente diseñadas teniendo en cuenta las necesidades de las publicaciones digitales, tales como periódicos, revistas.
- **Portal:** sitio web con contenido y funcionalidad diversa que sirve como fuente de información o como soporte a una comunidad.

Según el tipo de licencia escogido, los CMS se pueden clasificar en: CMS de código abierto u Open Source y CMS propietarios o privativos.

CMS propietarios o privativos: se denomina de esta manera a los sistemas de gestión de contenido comercializados por empresas que consideran el código fuente un activo más que tienen que mantener en propiedad y que no permiten que terceros tengan acceso (27).

CMS de código abierto u Open Source: son los desarrollados por individuos, grupos o empresas que permiten el acceso libre y la modificación del código fuente. Utilizar un CMS de código abierto tiene una ventaja que hace decidirse a la mayoría de los usuarios: su costo. Usualmente el software de código abierto es de acceso libre, por lo que la licencia es gratis (27). Algunos de los principales CMS de código abierto son WordPress, Drupal y Joomla (27):

1.5.1.2 Drupal 7.14

Drupal es un sistema de gestión de contenido muy configurable. Es de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Gracias a su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales de los que dispone, hace que sea adecuado para la realización de muchos sitios web. Permite publicar artículos, imágenes, otros archivos y servicios añadidos como foros, encuestas, votaciones, blogs, administración de usuarios y permisos (30).

La curva de aprendizaje de Drupal es demasiado pronunciada, prácticamente un muro de aprendizaje, por lo que puede costar mucho acostumbrarse a este CMS, su interfaz

por defecto es poco amigable, el menú de navegación es jerárquico, con dos o tres niveles, lo que confunde y dificulta el acceso rápido a la página que se busca. Drupal consume más recursos que otros CMS, además tiene la base de datos más grande por lo que el tiempo de respuesta suele ser mayor en comparación con otros sistemas y también su panel de administración no es lo suficientemente amigable (31).

1.5.1.3 Joomla 2.5.4

Es usado por muchas personas para crear desde sitios simples hasta complejas aplicaciones corporativas. Joomla es fácil de instalar, simple de manejar y confiable. Es un CMS que permite crear sitios web de alta interactividad, profesionalidad y eficiencia. Está programado en lenguaje PHP y SQL. Este sistema ofrece una interfaz de administración avanzada (32).

A pesar de estos beneficios, el trabajo diario con el sitio no es muy fácil, el tiempo de respuesta del sistema es muy largo, las operaciones simples toman enorme cantidad de tiempo y sus diseños son a menudo descuidados. No es fácil crear tipos de contenido propio y poder mostrarlos luego en diferentes vistas o páginas. Si no se encuentra o no existe el módulo o componente que soporte lo que se desea hacer, resulta casi imposible adaptarlo al CMS. No ofrece un modo sencillo de crear un flujo de trabajo basado en secciones o tipos de contenidos. Muchas de las plantillas que brindan una mejor distribución de sus componentes, y que por ende, proporciona mayor usabilidad y funcionalidades desde el punto de vista del usuario final, se encuentran bajo licencias privativas. Cuando hospeda mucha información se notan afectaciones de rendimiento. Deficiente planificación de la interfaz administrativa. Limitación en las opciones para personalizar rangos de usuarios. Dependencia excesiva del JavaScript en su panel de administración (33).

1.5.1.4 WordPress 4.2.2

WordPress es un sistema de gestión de contenido enfocado a la creación de blogs (sitios web periódicamente actualizados). Desarrollado en PHP y MySQL, bajo licencia GPL¹ y código modificable. Es un sistema que provee una plataforma fácil de usar para sus publicaciones personales con un enfoque en lo estético, estándares manejados en la web y en la usabilidad. Las causas de su enorme crecimiento son su licencia, su facilidad de uso, sus características como gestor de contenidos y su enorme comunidad de

¹ GPL o Licencia Pública general (*General Public License*) es una licencia que permite modificar el programa para adaptarlo a las propias necesidades de los usuarios ya que este viene acompañado de su código fuente. Además, podrás distribuir el programa GPL tal cual o después de haberlo modificado.

Capítulo 1. Fundamentación Teórica

desarrolladores y diseñadores que se encargan de crear extensiones y temas para la comunidad (34). Es una avanzada plataforma semántica de publicación personal orientada a la estética, los estándares web y la usabilidad, es libre y, al mismo tiempo, gratuito (35).

Características principales de WordPress:

- Permite la creación de múltiples blogs con un solo registro y de categorías y subcategorías para las entradas y además la posibilidad de agregar al sitio widgets con diversas funcionalidades (36).
- Posee capacidad de almacenamiento gratuito de hasta 3GB y exportación automática de la información de un blog (entradas, páginas y archivos) en un archivo XML² (37).
- Cuenta con una interfaz en varios idiomas en el modo diseño del blog y con estadísticas de la actividad de los lectores del blog, también permite la creación de páginas web (38).
- Dispone de decenas de plantillas listas para utilizar e importación automática de entradas, comentarios, páginas, archivos, categorías y etiquetas, provenientes de otros blogs (39)
- Permite la adición de comentarios de los lectores en entradas y páginas y posee una librería multimedia (archivos tipo jpg, jpeg, png, gif, pdf, doc, ppt y odt) (39).

Dispone actualmente de extensiones para prácticamente cualquier necesidad que se pueda tener en un sitio web. Posee una sencilla administración, es decir fácil de usar, por lo que es un sistema de publicación potente y con una buena gestión en general. Cuenta con un sistema de temas estructurados y organizados, donde además se pueden reutilizar ficheros de estos temas en otros, solo cambiando los estilos y clases CSS, con lo que se ahorra tiempo y se aprovechan los trabajos para un cliente como posibles soluciones para otros (35).

² Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones (105).

Capítulo 1. Fundamentación Teórica

Después de un detallado estudio de los principales CMS de código abierto se decidió presentar la propuesta de emplear el CMS WordPress en su versión 4.2.2 al cliente, por todas las características que el mismo presenta.

1.6 Base de Datos

Una base de datos es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos. Es un conjunto exhaustivo no redundante de datos estructurados organizados independientemente de su utilización y su implementación en máquina accesibles en tiempo real y compatibles con usuarios concurrentes con necesidad de información diferente y no predicable en tiempo (40).

1.6.1 MySQL 5.6.11

MySQL es un sistema de gestión de base de datos relacional, lo que permite velocidad y flexibilidad, en el cual los datos están almacenados en tablas separadas enlazadas entre sí. Tiene licencia dual, pudiéndose usar de forma gratuita bajo licencia GNU/GPL (Licencia Pública General) o bien adquiriendo licencias comerciales de MySQL AB en el caso de no desear estar sujeto a los términos de la licencia GPL. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente (41).

Las principales características de este gestor de bases de datos son las siguientes:

- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo y posee vistas actualizables (42).
- Agrupación de transacciones, reuniendo múltiples transacciones de varias conexiones para incrementar el número de transacciones por segundo (42).
- Dispone de una Interfaz de Programación de Aplicaciones (API's) en gran cantidad de lenguajes (C, C++, Java, PHP) (43).
- Posee una gestión de usuarios y contraseñas, manteniendo muy buen nivel de seguridad en los datos y tiene soporte a multiplataforma y a SSL (43).

1.7 Lenguajes de programación

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar

instrucciones a un equipo (44). En esta sección se especifican los diferentes lenguajes de programación empleados en la realización del producto.

1.7.1 Lenguaje de programación del lado del servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. Los lenguajes de lado servidor más ampliamente utilizados para el desarrollo de páginas dinámicas son el ASP, JSP, PERL y PHP (45).

1.7.1.1 PHP 5.5.3

PHP es un acrónimo recursivo que significa *PHP Hypertext Pre-processor*. Es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Este lenguaje es publicado bajo la licencia PHP, compatible con la Licencia Publica General de GNU. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión php (46). Este es el lenguaje de programación en el que está creado WordPress, por lo cual es necesario instalarlo para que pueda ser interpretado por el servidor.

Entre sus ventajas podemos encontrar:

- Tiene capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL que es el que se utiliza en el desarrollo del sistema propuesto (46).
- Es un lenguaje multiplataforma, con capacidad de expandir su potencial utilizando módulos (llamados ext's o extensiones) y además no requiere definición de tipos de variables (43).
- Es fácil de aprender, posee una amplia documentación en su página oficial (www.php.net), entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda y también tiene manejo de excepciones (42).

- Se caracteriza por ser un lenguaje muy rápido y permite las técnicas de Programación Orientada a Objetos (47).
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y cuenta con una biblioteca nativa de funciones sumamente amplia e incluida (43).

1.7.2 Lenguaje de programación del lado del cliente

Los lenguajes de programación del lado del cliente se ejecutan en el navegador del usuario. Las páginas del cliente son muy dependientes del sistema donde se están ejecutando y esa es su principal desventaja. Cada navegador tiene sus propias características y lo que puede funcionar en un navegador puede no funcionar en otro. Como ventaja se puede decir que ofrecen respuestas inmediatas a las acciones del usuario y permiten la utilización de recursos de la máquina local (48).

1.7.2.1 HTML 5

Es un lenguaje sencillo, fácil de entender y utilizar. La versión 5 de HTML no se limita sólo a crear nuevas etiquetas, atributos y eliminar las marcas que están en desuso o se utilizan inadecuadamente; sino que pretende proporcionar una plataforma para desarrollar aplicaciones web más parecidas a las aplicaciones de escritorio (49).

A continuación se exponen algunas de las ventajas:

- Ofrece la posibilidad de obtener un código más limpio y fácil de depurar, que los códigos de los estándares anteriores (50).
- Inclusión de las etiquetas video y audio. Soporta de manera eficiente y estable cualquier opción de ejecución de video y audio, sin generar errores o añadir código flash en el sitio web (50).
- Nueva estructura de etiquetas mejorada. Permite definir por separado el encabezado, la barra de navegación, las secciones de la página web, los textos del sitio, los diálogos y el pie de página de los sitios web (50).

1.7.2.2 CSS 3

Las hojas de estilo en cascada (*Cascading Style Sheets*, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML y por extensión en XHTML. Las hojas de estilo representan un avance importante para los diseñadores de páginas web, al darles un mayor rango de posibilidades para

mejorar la apariencia de sus páginas. El objetivo del desarrollo de CSS es separar la estructura de un documento de su presentación. Las hojas de estilo en CSS están compuestas de reglas (51).

1.7.2.3 Java Script 1.8

Con Java Script podemos crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones Java Script y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador (52).

Permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, Java Script pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente (53).

1.7.2.4 Frameworks de JavaScript JQuery 1.11.0

Consiste en un único fichero JavaScript que contiene las funcionalidades comunes del DOM³, eventos, efectos y AJAX⁴, permitiendo simplificar la manera de interactuar con los documentos HTML. Al igual que otras librerías, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código. Es decir, con las funciones propias de esta librería se logran resultados en menos tiempo de desarrollo (54).

1.8 Herramienta para el modelado Visual Parading for UML 8.0

Visual Paradigm es una herramienta CASE⁵ que utiliza UML como lenguaje de modelado. Está diseñada para una amplia gama de usuarios interesados en construir sistemas de software fiables con el uso del paradigma orientado a objetos, incluyendo actividades como ingeniería de software, análisis de sistemas y análisis de negocios (55).

³ *Document Object Model* (Modelo de Objetos del Documento). Define cómo los objetos de una página Web son representados, qué atributos tienen y cómo se manipulan. DOM es una interfaz para que aplicaciones y scripts puedan acceder a un documento pudiendo modificar su estructura y contenido (106).

⁴ AJAX (*Asynchronous JavaScript and XML*): Es una técnica de desarrollo web para crear aplicaciones interactivas (107).

⁵ CASE (Ingeniería de Software Asistida por Computador) (108).

Emplea las últimas notaciones de UML, ingeniería inversa, generación de código, exportación e importación XML. Soporta aplicaciones web, exporta en formato HTML, está disponible en varios idiomas, es fácil de instalar y fácil de actualizar. Los desarrolladores pueden diseñar la documentación del sistema con plantillas de diseño. El analista de sistema puede estimar las consecuencias de los cambios con los diagramas de análisis de impacto, como la matriz y el diagrama de análisis (55).

1.8.1 UML 2.0

Lenguaje de Modelado Unificado (UML), por sus siglas en inglés, (*Unified Modeling Language*): es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el Object Management Group (OMG). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. Ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables (56).

1.9 Servidores web

Un servidor web es un programa que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de Internet. Se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Básicamente consta de un intérprete HTTP el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe en pantalla (57).

1.9.1 Apache 2.4.4

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Es altamente configurable, admite bases de datos de autenticación y negociado de contenido, aunque carece de una interfaz gráfica que ayude en su configuración. Soporta PHP como lenguaje de programación y con los módulos adecuados puede soportar también ASP (58).

Es multiplataforma y además es extensible gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado

del servidor (59). Está basado en Hebras en la versión 2.0 y es modular, ya que puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API⁶ de programación de módulos, para el desarrollo de módulos específicos (60). Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos (61).

1.10 Apache JMeter 2.9

Apache JMeter es una herramienta de carga diseñada para realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web. Está diseñado para desarrollar diferentes tipos de test; permitiendo diseñar tanto sencillos tests que soliciten simples páginas web, como complejas secuencias de requisiciones que permitan evaluar el comportamiento de una aplicación o como la capacidad de carga máxima que pueda tener una aplicación en un servidor (62).

1.11 Entorno de Desarrollo Integrado (IDE) PhpStorm 8.0

PhpStorm es un IDE de programación desarrollado por JetBrains. Es uno de los entornos de programación más completos de la actualidad, permite editar código no sólo del lenguaje de programación php como lo indica su nombre (63). Permite la gestión de proyectos fácilmente y proporciona un fácil autocompletado de código. Además soporta el trabajo con PHP 5.5 y tiene una sintaxis abreviada (64). PhpStorm tiene un ambiente de desarrollo para WordPress que utiliza el estilo de código de WordPress. Esto permite integrar el WordPress en el PhpStorm para editar o crear nuevos plugins. Es compatible con las funcionalidades y características relacionadas con los hooks⁷ de WordPress (63).

1.12 Métricas y pruebas para la verificación y la validación del sistema

Para la verificación y la validación del sistema se hizo necesaria la aplicación de métricas y pruebas. La aplicación de estas garantizó que el software desarrollado cumpliera con los requisitos propuestos por el cliente.

1.12.1 Métricas

Un elemento clave de cualquier proceso de ingeniería es la medición. Se emplean medidas para valorar la calidad de los sistemas que se construyen. Una métrica es un

⁶ API Abreviatura de *Application Programming Interface*. Conjunto de convenciones internacionales que definen cómo debe invocarse una determinada función de un programa desde una aplicación.

⁷ Conjunto de funcionalidades predefinidas en WordPress.

instrumento que cuantifica un criterio (65). Las métricas de software tienen un papel decisivo en la obtención de un producto de alta calidad, porque determinan mediante estadísticas basadas en la experiencia, el avance del software y el cumplimiento de parámetros requeridos (66).

Métricas para requisitos

Para la validación de los requisitos se tienen en cuenta las métricas de estabilidad, especificidad y grado de validez de los requisitos; las cuales brindan información sobre la comprensibilidad y validez de los mismos (67).

Métricas para diseño

Para medir el diseño se utilizan las métricas básicas inspiradas en el estudio de la calidad del diseño orientado a objeto que propone Pressman. Las métricas escogidas para la validación del diseño fueron las métricas orientadas a las clases: Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC) (68).

1.12.2 Pruebas

La aplicación de pruebas es actualmente una herramienta importante para llevar el control de la realización de un proyecto de software de calidad. Cada vez son más rigurosas estas pruebas por el hecho de la gran competitividad creciente en el mercado (69).

1.12.2.1 Pruebas funcionales

Se denominan pruebas funcionales (Functional Testing), a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados cumplan con las funciones específicas para los cuales han sido creados, es común que sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales. Al realizarlas se pretende ponerse en los pies del usuario, usar el sistema como él lo usaría, sin embargo el analista de pruebas debe ir más allá que cualquier usuario, generalmente se requiere apoyo de los usuarios finales ya que ellos pueden aportar mucho en el desarrollo de casos de prueba complejos (70).

1.12.2.2 Pruebas de aceptación

Las pruebas de aceptación se ponen en práctica a través de los casos de prueba de aceptación que propone la metodología de desarrollo XP y con ellos se puede confirmar que cada Historia de Usuario ha sido implementada correctamente. Estas pruebas son

destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida por el cliente final (71).

1.13 Conclusiones parciales

Para documentar el proceso de desarrollo se seleccionó un subconjunto de prácticas y artefactos de la metodología ágil XP, que son representados utilizando Visual Paradigm como herramienta CASE. El estudio del estado del arte arrojó como resultado que las tendencias actuales de desarrollo de portales web utilizan los CMS. Entre estos se destaca el uso de WordPress como CMS líder en la actualidad. La selección de WordPress como sistema base para el desarrollo de la solución condicionó la selección de MySQL como gestor de base de datos al ser el soporte nativo del CMS y presentar además la mejor integración con casi todas las extensiones desarrolladas y los motores relacionales existentes. WordPress utiliza como lenguaje de programación PHP el cual pasó a ser el lenguaje para el desarrollo de la solución. Además se seleccionaron las métricas y pruebas a realizar para la validación de los requisitos funcionales y del diseño del sistema.

Capítulo 2. Descripción del portal

2.1 Introducción

En el presente capítulo se analizan las características con que cuenta el sistema desarrollado. Además se abordan los temas relacionados con las fases de exploración, planificación y diseño que define la metodología de desarrollo XP. También se tratan los principales artefactos generados así como la planificación del tiempo y el esfuerzo de las fases posteriores.

2.2 Propuesta del sistema

La propuesta de solución que se plantea pretende desarrollar una plataforma que permite la publicación de información relevante para el taller literario “Gertrudis Gómez de Avellaneda”, ya sean noticias, artículos o documentación de interés para el usuario. Para la solución se propone clasificar el contenido mediante categorías logrando una mayor organización de la información publicada. A continuación se muestra una imagen representativa del sistema (elaboración propia).



Figura 1 Propuesta del sistema

En la página principal o página de inicio se ubicaron los cinco últimos artículos publicados posibilitando acceder al resto de los mismos mediante un paginado al final de la página. Dicha página tiene un menú de navegación principal en el que se encuentra

información sobre las obras, las colecciones de obras, noticias, eventos e historia del taller.

Además de las funcionalidades anteriores el portal brinda la posibilidad de buscar artículos de interés mediante un buscador general y permite la autenticación contra el servicio de directorio LDAP de la universidad.

2.3 Características no funcionales del sistema

Las características no funcionales del sistema son aquellos requisitos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de éste como la fiabilidad, usabilidad, el tiempo de respuesta y la capacidad de almacenamiento. De acuerdo con la Norma ISO 9126 se definen un conjunto de requisitos no funcionales que debe cumplir el sistema (72).

2.3.1 Usabilidad

La solución propuesta será usada por personas que pueden o no tener conocimientos básicos de informática, por lo que la interfaz gráfica del sistema debió concebirse con un diseño sobrio y profesional, brindando una navegación fácil e intuitiva para los usuarios, es por ello que cuenta con un mapa de navegación visible desde todas las páginas y secciones del portal, además es visible en los principales navegadores web.

2.3.2 Funcionalidad

La solución propuesta debe poseer políticas de accesibilidad a las diferentes funcionalidades del mismo en dependencia del nivel de autorización que presente un usuario determinado. El sistema debe cumplir con las actividades correspondientes a un taller literario.

2.3.3 Mantenibilidad

El sistema está implementado con la característica principal de desarrollo y utilización de extensiones, lo que permite integrar, modificar y eliminar funcionalidades sin que se interrumpa su ejecución, esto facilita su modificación y cambio de apariencia. Además de poder adaptarse a las actualizaciones que el CMS distribuya.

2.3.4 Fiabilidad

Para garantizar la fiabilidad del software se fueron entregando N versiones del sistema, cada una de ellas más completas en especificaciones y requerimientos. Se realizaron

Capítulo 2. Descripción del portal

pruebas de aceptación a cada una de estas versiones, para comprobar la tolerancia a fallos del programa. También se hace uso de la extensión BackUpWordPress, que realiza una copia de seguridad de todo el sitio en un horario definido por el cliente, incluyendo la base de datos y todos sus archivos.

2.3.5 Portabilidad

El sistema podrá ser utilizado bajo cualquier distribución Linux o Windows, convirtiéndolo así en multiplataforma. Es posible su instalación en cualquiera de estos dos sistemas con todas sus características y funcionalidades.

2.3.6 Eficiencia

Con el fin de garantizar la eficiencia el sistema demanda en el cliente una computadora con 256MB de RAM como mínimo, con velocidad de procesamiento igual o mayor a 1.3 GHz y una tarjeta de red. El servidor debe tener 4GB RAM y 250 GB de capacidad en disco duro como mínimo. Para un mejor rendimiento del sistema cuenta con un tiempo de respuesta por transacción entre 1 y 10 segundos, además de poseer una mínima utilización de memoria física.

2.4 Fase de exploración

La metodología de desarrollo XP comienza con la fase de exploración, en esta fase el cliente define lo que necesita mediante las historia de usuarios. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán para el desarrollo del sistema, siendo además en esta fase donde los programadores estiman el tiempo de desarrollo.

2.4.1 Involucrados en el sistema

Se definieron como involucrados en el sistema todos aquellos que realizan una función o interactúan con él de una forma u otra. Todos los usuarios que interactúen en el sistema en excepción del Invitado serán usuarios autenticados en el sistema.

Tabla 2 Involucrados en el sistema

| Persona relacionada con el sistema | Justificación |
|---|--|
| Invitado | Es la persona que navega por el sistema sin estar autenticado. Fluye dentro de este sin privilegios de |

Capítulo 2. Descripción del portal

| | |
|---------------------|---|
| | acceder a los recursos que el portal define para usuarios autenticados. |
| Usuario autenticado | Es la persona que se encuentra registrada en el sistema y al acceder a él tendrá el privilegio de disfrutar de los servicios y contenidos que el portal brinda a los usuarios autenticados. |
| Editor | Es la persona que puede gestionar los contenidos que han sido publicados en el portal sin acceder a las funcionalidades administrativas del portal. |
| Redactor | Es la persona que tiene permisos de publicar artículos en el portal. |
| Administrador | Es la persona autorizada para la gestión general del sistema. Encargado de actualizar, modificar, eliminar e insertar toda la información del sitio. |

2.4.2 Historias de Usuario

Las historias de usuario (HU) es la técnica que utiliza XP para especificar los requisitos del software, realizándose una por cada característica fundamental del sistema; tienen el mismo propósito que los casos de uso en las metodologías de desarrollo de software pesadas, aunque no son lo mismo ya que son escritas por los propios clientes desde su perspectiva del sistema, por lo que serán descripciones cortas y escritas en el lenguaje del cliente (73).

Para definir las historias de usuario se utilizó el siguiente prototipo, que contiene todos los datos necesarios para desarrollar la funcionalidad descrita.

Durante la fase de exploración se identificaron 13 HU para el desarrollo del Portal web taller literario “Gertrudis Gómez de Avellaneda”, a continuación se presenta la HU Gestionar obra literaria personal, el resto se encuentran en el acápite Anexo II:

Tabla 3 HU Gestionar obra literaria personal

| Historia de usuario | |
|--|--|
| Número: 7 | Nombre: Gestionar obra literaria personal |
| Usuario: usuario autenticado, redactor, editor, administrador | |

| | |
|--|-----------------------------------|
| Prioridad en negocio: alta | Riesgo en desarrollo: alto |
| Iteración asignada: 2 | |
| Descripción: permite al usuario desde la administración del portal crear, modificar y eliminar las obras literarias personales. | |
| Observaciones: | |

2.5 Fase de estimación y planificación

La fase de planificación de la metodología de XP como bien indica su nombre, se encarga de planificar el proceso de desarrollo de una aplicación de software, en esta se realiza una estimación del esfuerzo por historia de usuario. Para la realización de la estimación en XP las métricas son libres, por lo que puede utilizarse cualquier criterio para medir el desempeño de un proyecto (18).

2.5.1 Estimación de esfuerzo por historias de usuarios

Para el desarrollo de la aplicación propuesta se realizó una estimación de esfuerzo en días por cada una de las historias de usuarios identificadas, los resultados se muestran a continuación.

Tabla 4 Estimación de esfuerzo por historias de usuario

| No | Historia de usuario | Puntos estimados | Puntos reales |
|----|-----------------------------------|------------------|---------------|
| 1 | Autenticar usuario | 2 | 2 |
| 2 | Gestionar artículo | 3 | 2 |
| 3 | Mostrar artículo en portada | 5 | 4 |
| 4 | Comentar artículo | 2 | 2 |
| 5 | Gestionar miembro | 3 | 3 |
| 6 | Adjuntar archivo | 3 | 3 |
| 7 | Gestionar obra literaria personal | 4 | 3 |
| 8 | Historia del taller | 2 | 2 |
| 9 | Mapa de navegación | 2 | 2 |

Capítulo 2. Descripción del portal

| | | | |
|----|-----------------------|---|---|
| 10 | Gestionar colección | 3 | 3 |
| 11 | Consejo de literatura | 2 | 2 |
| 12 | Enlace externo | 2 | 2 |
| 13 | Buscar artículo | 2 | 2 |

2.5.2 Plan de iteraciones

Después de ser identificadas las historias de usuario y estimado el esfuerzo dedicado a la realización de cada una de estas se procedió a la realización de la planificación de la etapa de desarrollo del Portal web taller literario “Gertrudis Gómez de Avellaneda”. Para un mejor desempeño en el desarrollo del portal se estableció la implementación del mismo en 3 iteraciones, donde una iteración no es más que un mini-proyecto en el que se obtiene un resultado en software con un valor para el cliente. Cada iteración está conformada por las historias de usuarios seleccionadas por el cliente a implementar así como por la estimación completa de dicha iteración.

Tabla 5 Plan de iteraciones

| Iteración | Orden de las historias de usuario a implementar | Duración |
|-------------|---|-----------|
| Iteración 1 | Autenticar usuario Gestionar artículo Mostrar artículos en portada Comentar artículo | 3 semanas |
| Iteración 2 | Gestionar miembro Adjuntar archivo Gestionar obra literaria personal Historia del taller Mapa de navegación | 3 semanas |
| Iteración 3 | Gestionar colección Consejo de literatura Enlace externo Buscar artículo | 2 semanas |

2.6 Tarjetas CRC (Clase – Responsabilidad – Colaborador)

Las tarjetas CRC son en la práctica pequeñas tarjetas de cartón que se elaboran para ser mostradas al cliente, de manera que se pueda llegar a un acuerdo sobre la validez de las abstracciones propuestas, lo que ayuda al equipo durante el diseño del sistema (74).

En la Programación Orientada a Objetos (POO), cada tarjeta representa una clase y define sus responsabilidades (lo que hace) y las colaboraciones con las otras clases (cómo se comunica con ellas), por lo que frecuentemente cada tarjeta CRC se convierte en un objeto, sus responsabilidades en métodos públicos y sus colaboradores en llamadas a otras clases. Por esta razón es posible afirmar que las tarjetas CRC se utilizan para llevar la realidad al modelo orientado a objetos (como si fuera un diagrama UML) (23).

A continuación se describe una de las tarjetas CRC correspondiente a la funcionalidad Gestionar obra, el resto de las tarjetas CRC se encuentran visibles en el acápite Anexo II:

Tabla 6 Tarjeta CRC de la clase Obra_TLGGA

| | |
|--|-----------------------|
| Clase: Obra_TLGGA | |
| Descripción: clase encargada de adicionar el menú Gestionar Obra del taller literario “Gertrudis Gómez de Avellaneda” a la administración del portal. | |
| Responsabilidades | Colaboradores |
| Adicionar menú en la administración Insertar dinámicamente la hoja de estilos | Obra_TLGGA_Controller |

2.7 Diagrama de despliegue

El diagrama de despliegue permite visualizar la relación física de los componentes, dispositivos, hardware y software que se desplegará en el ambiente seleccionado. El diagrama de despliegue del Portal web taller literario “Gertrudis Gómez de Avellaneda” se muestra a continuación.

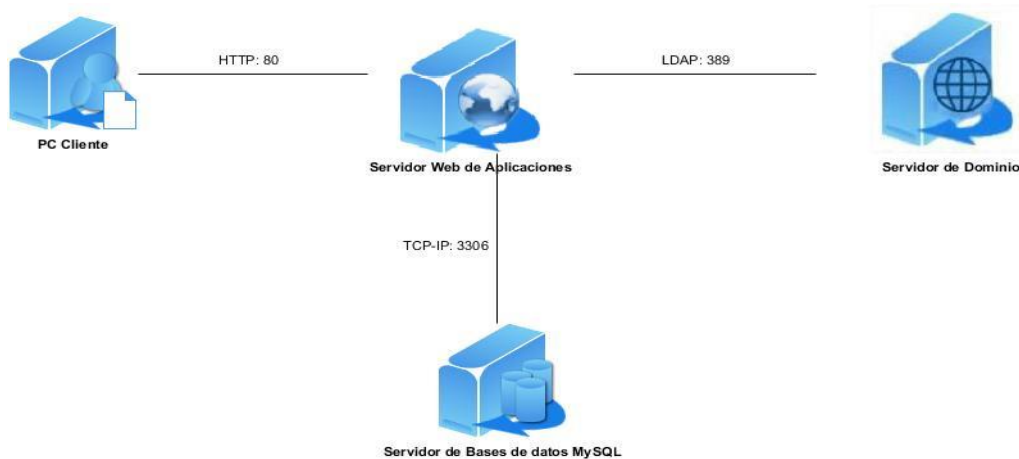


Figura 2 Diagrama de despliegue

En el diagrama de despliegue se evidencia la presencia de 4 nodos: PC Cliente, Servidor Web de aplicaciones, Servidor de BD MySQL y Servidor de dominio. La relación entre estos inicia cuando la PC Cliente solicita mediante el protocolo HTTP (puerto 80) una petición al Servidor Web, este a su vez se comunica mediante TCP-IP (puerto 3306) con el Servidor de BD MySQL. De ser necesaria una comunicación con el servidor de dominio esta se realiza mediante el protocolo LDAP (puerto 389) entre el Servidor de Dominio y el Servidor Web, el servidor de Dominio le envía una respuesta al Servidor Web y este a su vez finaliza la petición enviando la respuesta a la PC Cliente donde se muestra la información solicitada en el navegador del usuario.

2.8 Mapa de navegación

Los mapas de navegación proporcionan una representación esquemática de la estructura del hipertexto, indicando los principales conceptos incluidos en el espacio de la información y las interrelaciones que existen entre ellos. El mapa de navegación refleja la estructura de la web por medio de enlaces a los nodos principales, y estos también pueden desarrollarse para mostrar los subnodos (75). A continuación se muestra el mapa de navegación del Portal web taller literario “Gertrudis Gómez de Avellaneda”.

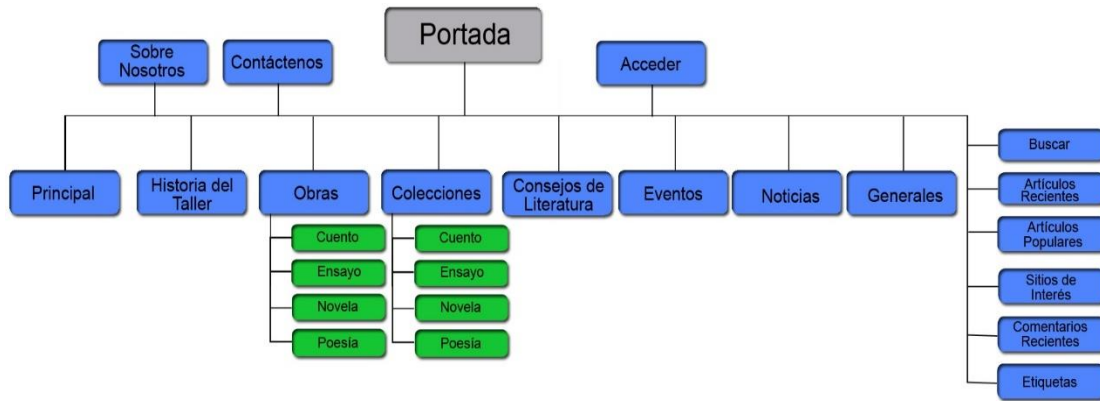


Figura 3 Mapa de navegación

2.9 Patrones de diseño

Un patrón de diseño es una solución general que se aplica a un problema de diseño de cualquier sistema informático. Además de poder ser aplicable a distintos problemas en diferentes circunstancias, también debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores (76).

El CMS WordPress no define propiamente patrones de diseño para guiar el desarrollo de nuevos componentes o funcionalidades propias, como nuevas extensiones (*plugins*) y temas (*themes*), pero si se puede observar el uso de patrones de diseño orientado a objetos en la implementación de las funcionalidades que conforman el núcleo del sistema, como son los patrones GRASP y los GOF.

2.9.1 Patrones GRASP

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos. Sirven para normalizar quien hace que cosa en cada momento y como debe realizarse (74).

Experto

La función del patrón experto es asignar una responsabilidad al experto en información, es decir, una clase contiene toda la información necesaria para cumplir la responsabilidad que le ha sido encomendada (74). Todas las clases en WordPress son expertas en información, teniendo cada una toda la información referente a ellas y a su comportamiento, de esta manera se logra que realicen las tareas que tienen asignadas en el sistema sin verse afectada otras clases. Ejemplo es la clase Usuario (*user.php*), esta posee toda la información referente a los usuarios.

```
function edit_user( $user_id = 0 ) {
    global $wp_roles;
    $user = new stdClass;
    if ( $user_id ) {
        $update = true;
        $user->ID = (int) $user_id;
        $userdata = get_userdata( $user_id );
        $user->user_login = wp_slash( $userdata->user_login );
    } else {
        $update = false;
    }
}
```

Figura 4 Segmento de la clase user.php

Bajo acoplamiento

El acoplamiento de una clase es el conjunto de dependencias que tiene esta con otras clases. Bajo acoplamiento significa que una clase no posee dependencias de muchas clases, cuanto menor sea el acoplamiento o las dependencias entre ellas, menor influencia tendrán los cambios efectuados (74). En WordPress se evidencia en sus clases el uso de este patrón, por ejemplo al modificar la clase Comentario (comment.php) no se afecta la clase Artículos (post.php) y así se puede observar respectivamente con las demás.

```
function edit_post( $post_data = null ) {
    global $wpdb;
    if ( empty($post_data) )
        $post_data = &$_POST;
    unset( $post_data['filter'] );
    $post_ID = (int) $post_data['post_ID'];
    $post = get_post( $post_ID );
    $post_data['post_type'] = $post->post_type;
    $post_data['post_mime_type'] = $post->post_mime_type;
}
```

Figura 5 Segmento de la clase post.php

```
function edit_comment() {
    if ( ! current_user_can( 'edit_comment', (int) $_POST['comment_ID'] ) )
        wp_die ( __( 'You are not allowed to edit comments on this post.' ) );
    if ( isset( $_POST['newcomment_author'] ) )
        $_POST['comment_author'] = $_POST['newcomment_author'];
    if ( isset( $_POST['newcomment_author_email'] ) )
        $_POST['comment_author_email'] = $_POST['newcomment_author_email'];
    if ( isset( $_POST['newcomment_author_url'] ) )
        $_POST['comment_author_url'] = $_POST['newcomment_author_url'];
    if ( isset( $_POST['comment_status'] ) )
        $_POST['comment_approved'] = $_POST['comment_status'];
    if ( isset( $_POST['content'] ) )
        $_POST['comment_content'] = $_POST['content'];
    if ( isset( $_POST['comment_ID'] ) )
        $_POST['comment_ID'] = (int) $_POST['comment_ID'];
}
```

Figura 6 Segmento de la clase comment.php

Alta cohesión

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Es necesario tener en cuenta que al asignar responsabilidades en un diseño, se deben buscar soluciones que asignen los métodos a las clases de forma coherente, completa y relacionada, de esta forma se obtendrán clases cohesionadas (74). El diseño del sistema se realizó de forma tal que todas las clases tienen una responsabilidad única en el diseño y todas las clases trabajan juntas para propiciar el funcionamiento del sistema, por ejemplo la clase `Obra_TLGGA_Controller` fue creada para gestionar las obras y es quien lo hace, pero necesita de la clase complementaria `Obra_TLGGA_View`.

```
require_once 'class_obra_tlgga_view.php';

class Obra_TLGGA_Controller {
    private $view;

    function __construct()
    {
        $this->view = new Obra_TLGGA_View();
    }
}
```

Figura 7 Segmento de la clase Obra_TLGGA_Controller

Creador

El patrón creador ayuda a identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Una de las consecuencias de usar este

patrón es la visibilidad entre la clase creada y la clase creador. Una ventaja es el bajo acoplamiento, lo cual supone facilidad de mantenimiento y reutilización. La creación de instancias es una de las actividades más comunes en un sistema orientado a objetos. En consecuencia es útil contar con un principio general para la asignación de las responsabilidades de creación. Si se asignan bien, el diseño puede soportar un bajo acoplamiento, mayor claridad, encapsulación y reutilización (74). Este patrón se evidencia en la clase Member_TLGGA_Controller.

```
require_once 'class-members-tlgga-view.php';

class Member_TLGGA_Controller{
    private $view;

    function __construct(){
        $this->view = new Members_TLGGA_View();
    }
}
```

Figura 8 Segmento de la clase Member_TLGGA_Controller

Controlador

El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control. Se recomienda dividir los eventos del sistema en el mayor número de controladores para poder aumentar la cohesión y disminuir el acoplamiento (74). Ejemplo de este patrón se evidencia en la clase functions.php.

```
function alx_load() {
    // Load theme languages
    load_theme_textdomain( 'TLGGA', get_template_directory().'/languages' );
    // Load Coleccion, memeber, Obras
    load_template( get_template_directory() . '/functions/admin/load.php' );
    // Load theme options and meta boxes
    load_template( get_template_directory() . '/functions/theme-options.php' );
    load_template( get_template_directory() . '/functions/meta-boxes.php' );
}
```

Figura 9 Segmento de la clase functions.php

2.9.2 Patrones GOF

Los patrones GOF son los patrones más conocidos y usados en la actualidad en el campo del Diseño Orientado a Objetos, se dividen en creacionales, estructurales y de comportamiento (77).

Creacionales: abstraen el proceso de creación de instancias y ocultan los detalles de cómo los objetos son creados (78).

- **Instancia única**

Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos (79). En el portal se evidencia en la clase user.php, debido a que en cualquier parte que se llame dentro del proyecto se pueden usar sus atributos, sin necesidad de crear nuevas instancias de dicho objeto, por ejemplo en el archivo uci-ldap.php perteneciente al plugin UCI-LDAP.

```
$userData = array(  
    'user_pass' => $password,  
    'user_login' => $datos_usuario->Usuario,  
    'user_nicename' => $datos_usuario->Nombres,  
    'user_email' => $datos_usuario->Correo,  
    'display_name' => $datos_usuario->Nombres,  
    'first_name' => $datos_usuario->Nombres,  
    'last_name' => $datos_usuario->Apellidos,  
    'role' => $role  
);
```

Figura 10 Segmento del archivo uci-ldap.php

Comportamiento: ofrecen soluciones respecto a la interacción y responsabilidades entre clases y objetos, así como los algoritmos que encapsulan (80).

- **Observador**

El patrón Observador define una dependencia de uno a muchos entre objetos, de forma tal que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él (79). Este patrón se evidencia en WordPress al cambiar el nombre de una categoría, pues la categoría a la que pertenecen los artículos que están publicados bajo la misma categoría también cambia.

```
function _make_cat_compat( &$category ) {
    if ( is_object( $category ) ) {
        $category->cat_ID = &$category->term_id;
        $category->category_count = &$category->count;
        $category->category_description = &$category->description;
        $category->cat_name = &$category->name;
        $category->category_nicename = &$category->slug;
        $category->category_parent = &$category->parent;
    } elseif ( is_array( $category ) && isset( $category['term_id'] ) ) {
        $category['cat_ID'] = &$category['term_id'];
        $category['category_count'] = &$category['count'];
        $category['category_description'] = &$category['description'];
        $category['cat_name'] = &$category['name'];
        $category['category_nicename'] = &$category['slug'];
        $category['category_parent'] = &$category['parent'];
    }
}
```

Figura 11 Segmento del archivo category.php

2.10 Conclusiones parciales

Se detallaron las funcionalidades del sistema y se identificaron las personas que intervendrán en él. Se definieron 3 iteraciones que abarcan un total de 13 historias de usuario, que describen las funcionalidades principales con que contará el sistema. Asociado a estas historias de usuarios se construyó el plan de iteraciones y la estimación de esfuerzo por historias de usuarios. Los artefactos necesarios para la implementación de la solución cierran el capítulo.

Capítulo 3. Implementación y Prueba

3.1 Introducción

En el capítulo se describen las fases de implementación y pruebas, siendo estas propias de la metodología de desarrollo XP y entre las más importantes en el ciclo de vida de cualquier software. En estas fases se materializó el código mediante su codificación y se realizaron las pruebas necesarias para comprobar el correcto funcionamiento del sistema.

3.2 Arquitectura

La arquitectura del sistema está fundamentada por el CMS WordPress, este propone una estructura propia para los sistemas desarrollados sobre él. Trabaja con los archivos de manera independiente, de manera que cada archivo se puede modificar y adaptarse a las necesidades del administrador y/o desarrollador del sistema. Luego, cada uno de estos archivos podrá llamarse desde un archivo principal, donde este se encarga de incluir cada una de las partes, como la cabecera, el contenido, la barra lateral y el pie de página para todas las páginas que conformen el tema utilizado en el sistema (81).

La configuración estándar de este CMS comprende todos los archivos de configuración del sistema de contenidos, además de un conjunto de carpetas principales que en conjunto con las funciones que se encuentran dentro de estas y que están agrupadas en archivos con extensión PHP conforman el sistema de archivos de WordPress (82).

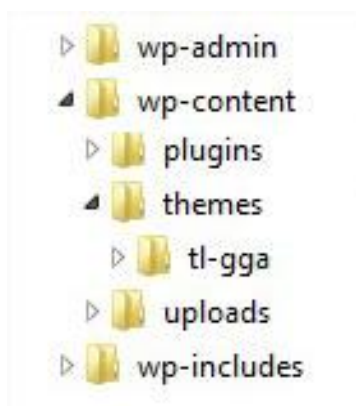


Figura 12 Estructura de archivos de WordPress

Estas carpetas principales son:

- **wp-admin:** esta carpeta contiene todos los archivos y funciones que se utilizan en la administración de WordPress (83).

- **wp-content:** esta carpeta está compuesta aparte de otros archivos PHP por tres carpetas más, la carpeta de las extensiones (*plugins*), la carpeta de los temas (*themes*) y la carpeta de los archivos subidos al servidor (*uploads*) (84).
- **wp-includes:** en esta carpeta se encuentran las funciones que manejan las entradas, los comentarios, las taxonomías como las etiquetas y las categorías (83).

3.2.1 Arquitectura Cliente/Servidor

La arquitectura Cliente/Servidor fue la utilizada para el desarrollo de la aplicación. Según IBM, es la arquitectura que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas (85).

La utilización de los diferentes servicios y aplicaciones existentes en internet se lleva a cabo respondiendo a este modelo, que es una forma de especializar terminales y programas para que las actividades y tareas se ejecuten con la mayor eficiencia posible. Mediante esta arquitectura el usuario puede acceder a la información sin tener en cuenta su ubicación física y donde puede estar alojada la misma (86).

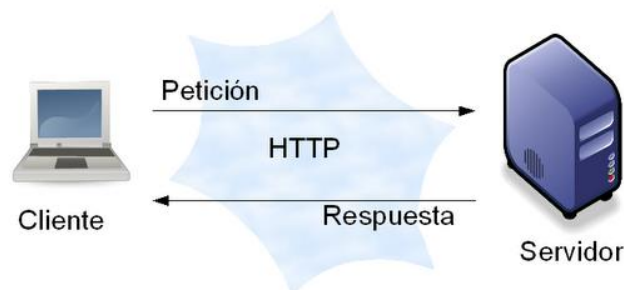


Figura 13 Arquitectura Cliente- Servidor

3.3 Análisis de extensiones existentes

La reutilización de código se refiere al comportamiento y a las técnicas que garantizan que una parte o la totalidad de un programa informático existente se puedan emplear en la construcción de otro programa. De esta forma se aprovecha el trabajo anterior, se economiza tiempo y se reduce la redundancia (87).

La forma más común de reutilizar código es poner el código en un único lugar y llamarlo desde los diferentes. Este proceso se conoce como abstracción y puede verse claramente en el archivo *functions.php* donde en este es posible implementar diferentes

Capítulo 3. Implementación y Prueba

funcionalidades para luego ser llamadas cuando se necesiten, evitando de esta manera que el desarrollador tenga que duplicar el código en los archivos de las vistas finales que necesitan la funcionalidad requerida (87).

En el desarrollo del Portal web taller literario “Gertrudis Gómez de Avellaneda” se utilizaron los beneficios que ofrece la reutilización de código, debido a que se hizo uso de extensiones implementadas con anterioridad que facilitan el trabajo a los desarrolladores, a continuación se mencionan y explican las que se consideran más importantes.

WP-PageNavi: funcionalidad que implementa la paginación de todas las páginas del sistema (88).

Disable Google Fonts: esta funcionalidad detiene la carga de las fuentes de Google Fonts y otras fuentes utilizadas por WordPress directamente desde internet (89).

UCI-LDAP: esta funcionalidad permite integrar la autenticación del sistema con el servicio de directorio LDAP de la UCI (90).

BackUpWordPress: esta funcionalidad hace una copia de seguridad en un horario definido como el conveniente del sitio entero, incluyendo la base de datos y todos los archivos (91).

3.4 Descripción de nuevos componentes

Para satisfacer el diseño propuesto se decidió desarrollar un tema de WordPress para el Portal web taller literario “Gertrudis Gómez de Avellaneda”. Los nuevos tipos de contenidos generan la necesidad de implementar nuevas funcionalidades con el fin de responder a la correcta visualización de las páginas del sistema. Entre las vistas del sistema se encuentran:

Gestionar Obras: se encarga de mostrar todas las obras del taller, organizadas según su categoría.

Gestionar Colecciones: se encarga de mostrar todas las colecciones de obras del taller, organizadas según su categoría.

Miembros TL-GGA: se encarga de mostrar todos los miembros del taller.

Cada una de las vistas anteriores corresponde a las funcionalidades que se necesitan en el portal por parte del cliente. Estas funcionalidades fueron desarrolladas en su totalidad por el equipo de desarrollo.

3.5 Temas en WordPress

Los temas (o plantillas) son archivos y estilos que funcionan en conjunto para crear la presentación de un sitio. Cada tema puede ser diferente, ofreciendo opciones para mejorar la experiencia de los usuarios o para cambiar de forma instantánea el aspecto visual del sitio. Los temas pueden incluir archivos de plantilla personalizados, archivos de imagen, hojas de estilos, páginas personalizadas, así como cualquier otro archivo de código necesario. Es posible crear un nuevo tema desarrollándolo desde cero o partiendo de uno ya creado (92).

3.5.1 Tema para el portal web

Por las especificaciones y necesidades del cliente se partió del tema *hueman* disponible en el sitio oficial de WordPress, las características de este tema son semejantes a las definidas para el Portal web taller literario “Gertrudis Gómez de Avellaneda”.

Lo primero para hacer coincidir el tema seleccionado con el diseño previsto es seleccionar el nombre, utilizado en este caso *tl-gga*, y ubicarlo en el directorio *wp-content/themes/tl-gga*. Luego se creó un archivo con nombre *style.css* con la información básica tal como el nombre, versión, autor. Esta información se declara en la cabecera del documento *style.css* en forma de comentario. Finalmente se creó el archivo *index.php* y el tema ya es reconocido en la administración del portal. A partir de aquí se crean e implementan los archivos en correspondencia con las características y funcionalidades con que contará el portal.

3.6 Seguridad del sistema

La seguridad en las aplicaciones web es uno de los factores que no puede ser olvidado por el equipo de desarrollo. Las mayores amenazas a la seguridad provienen de los datos que se introducen en los formularios de los sistemas informáticos, esto se refiere a scripts que traspasan los permisos asignados a los usuarios finales y la inyección de código SQL. Respecto a estos ataques PHP ofrece a los desarrolladores la posibilidad de filtrar los datos y validarlos (93).

Capítulo 3. Implementación y Prueba

Previendo posibles ataques al sistema se utilizó para el desarrollo la función `mysql_real_escape_string ()` a la hora de utilizar variables en sentencias SQL, esta función se encarga de escapar caracteres especiales que contengan las variables.

3.6.1 Extensiones para garantizar la seguridad del sistema

Aunque los desarrolladores validen los datos de entrada del sistema, nunca es demasiado si en cuanto a seguridad se trata, a continuación se explican dos extensiones a utilizar para garantizar la seguridad del sistema.

- **Ultimate Security Checker:** extensión de seguridad que analiza tu sitio y lo puntúa dependiendo del grado de seguridad en que se encuentra tu plataforma. Muestra los errores para que puedas repararlos o te ayuda a que los resuelvas automáticamente (94).
- **Limit Login Attempts:** establece un límite de números de intentos de conexión posible en la autenticación para acceder al sistema (95).

3.7 Fase de implementación

XP como metodología de desarrollo de software define al inicio de cada iteración la revisión del plan de iteraciones y se expresan las tareas de ingeniería, donde a cada una de ellas le es asignada los programadores responsables. Se describieron un total de 19 tareas de ingeniería, a continuación se muestra una de las pertenecientes a la historia de usuario Gestionar obra literaria personal, el resto se encuentran en el acápite Anexo II:

Tabla 7 Tarea 1 de la historia de usuario: Gestionar obra literaria personal

| Tarea de Ingeniería | |
|--|--|
| Número de Tarea: 1 | Número Historia de Usuario: 7, Gestionar obras literaria personal. |
| Nombre Tarea: Diseñar la interfaz de Gestionar obra literaria personal. | |
| Tipo de Tarea: desarrollo | Puntos Estimados (días): 2 |
| Fecha Inicio: 05 de marzo de 2015 | Fecha Fin: 07 de marzo de 2015 |
| Programador Responsable: Leanna L. Amador Rios – Abel A. Irsula Tumbarell | |

Descripción: se diseña la interfaz que permite gestionar la información de las obras literarias personales.

3.8 Métricas para requisitos

Una vez definidos los requisitos del sistema, estos deben ser validados para asegurar que los requisitos que se determinaron especifican realmente el sistema que el cliente desea obtener.

3.8.1 Estabilidad de los requisitos

El objetivo de esta métrica es medir la estabilidad de los requisitos para asegurar su adecuación antes de pasar al próximo flujo de trabajo. Se considera que los requisitos son estables cuando no existen adiciones o supresiones en ellos que impliquen modificaciones en las funcionalidades principales de la aplicación (66).

$$ETR = \left[\frac{RT - RM}{RT} \right] \times 100$$

Donde:

1. ETR: valor de la estabilidad de los requisitos
2. RT: total de requisitos definidos
3. RM: número de requisitos modificados, que se obtienen como la sumatoria de los requisitos insertados, modificados y eliminados.

Esta métrica ofrece valores entre 0 y 100. El mejor valor de ETR es el más cercano a 100 porque mostrará que no se están realizando cambios sobre los requisitos, son estables y por tanto es confiable trabajar el análisis y diseño sobre ellos.

Se identificaron un total de 13 requisitos funcionales, de los cuales 2 fueron modificados, por tanto: $ETR = \left[\frac{13-2}{13} \right] \times 100 = 85$

Como resultado se obtuvo un valor de 85, este número demuestra que no se han realizado cambios significativos sobre los requisitos, son estables y por tanto, es confiable el diseño efectuado sobre ellos.

3.8.2 Especificidad de los requisitos

El objetivo de esta métrica es cuantificar la especificidad o falta de ambigüedad en la definición de los requisitos. Para calcular esta métrica deben contarse los requisitos que tuvieron igual interpretación por los revisores y compararlos con el total de requisitos definidos. El valor de esta métrica debe estar siempre entre 0 y 1. Mientras más cerca

de 1 esté el valor de la especificidad de los requisitos mayor será la consistencia de la interpretación de los revisores para cada requisito y menor será la ambigüedad en la especificación de los requisitos (96).

$$ER = n_{ui}/n_r$$

Donde:

- ✓ ER: grado de especificidad de los requisitos
- ✓ n_{ui} : número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas.
- ✓ n_r : cantidad de requisitos en una especificación

Se obtiene como resultado: $ER = 13/13 = 1$

Como el valor que se obtuvo de ER es 1, se puede interpretar que la especificación de los requisitos no posee ambigüedad. Esta característica garantiza mayor calidad en el proceso de especificación.

3.8.3 Grado de validación de los requisitos

Los requisitos deben ser posibles de validar. La validación de los requisitos se realiza en consenso del equipo de desarrollo al contrastar lo que desea el cliente con la posibilidad real de implementarlo. El grado de validación de los requisitos mide la corrección en la definición de los requisitos (66).

$$VR = n_c / (n_c + n_{nv})$$

Donde:

- ✓ VR: grado de validación de los requisitos.
- ✓ n_c : número de requisitos que se han validado como correctos.
- ✓ n_{nv} : número de requisitos no validados aún.

El resultado de esta métrica está siempre entre 0 y 1. El valor óptimo de esta métrica es el más cercano a 1 e indica un alto nivel de corrección en la definición de los requisitos.

Se obtiene como resultado: $VR = 13 / (13 + 0) = 13/13 = 1$

La aplicación de esta métrica dio como resultado 1, por lo tanto se concluye que la definición de los requisitos es correcta.

3.9 Métricas para diseño

Para medir el diseño se utilizan las métricas orientadas a clases: Tamaño Operacional de Clase (TOC) y Relaciones entre Clases (RC)

3.9.1 Tamaño Operacional de Clase (TOC)

Consiste en medir el tamaño general de una clase tomando el valor de la cantidad de operaciones que están encapsuladas dentro de dicha clase. Si el resultado obtenido indica valores grandes, significa que la clase posee un alto grado de responsabilidad, debido a esto se reducirá la reutilización, se hará mucho más difícil la implementación y la realización de pruebas de dicha clase. Por tanto mientras menor sea el valor para el TOC se hará mucho más fácil la reutilización de dicha clase dentro del sistema (68).

Tabla 8 Rango de valores para la evaluación de los atributos de calidad

| Atributo | Categoría | Criterio |
|--------------------------------------|-----------|---|
| Responsabilidad | Baja | Cantidad de procedimientos \leq Promedio |
| | Media | Promedio \leq Cantidad de procedimientos $\leq 2 * \text{Promedio}$ |
| | Alta | Cantidad de procedimientos $> 2 * \text{Promedio}$ |
| Complejidad de Implementación | Baja | Cantidad de procedimientos \leq Promedio |
| | Media | Promedio \leq Cantidad de procedimientos $\leq 2 * \text{Promedio}$ |
| | Alta | Cantidad de procedimientos $> 2 * \text{Promedio}$ |
| Reutilización | Baja | Cantidad de procedimientos $> 2 * \text{Promedio}$ |
| | Media | Promedio \leq Cantidad de procedimientos $\leq 2 * \text{Promedio}$ |
| | Alta | Cantidad de procedimientos \leq Promedio |

Tabla 9 Aplicación de la métrica TOC

| No | Clase | Cant. Proc | Resp | Complej | Reutil |
|----|-----------------------|------------|------|---------|--------|
| 1 | Obra_TLGGA_Controller | 2 | Baja | Baja | Alta |
| 2 | Obra_TLGGA_List_Table | 3 | Baja | Baja | Alta |

Capítulo 3. Implementación y Prueba

| | | | | | |
|----|----------------------------|---|-------|-------|-------|
| 3 | Obra_TLGGA_View | 7 | Media | Media | Media |
| 4 | Obras_TLGGA | 6 | Media | Media | Media |
| 5 | Coleccion_TLGGA_Controller | 2 | Baja | Baja | Alta |
| 6 | Coleccion_TLGGA_List_Table | 3 | Baja | Baja | Alta |
| 7 | Coleccion_TLGGA_View | 7 | Media | Media | Media |
| 8 | Coleccion_TLGGA | 6 | Media | Media | Media |
| 9 | Member_TLGGA_Controller | 2 | Baja | Baja | Alta |
| 10 | Members_TLGGA_List_Table | 3 | Baja | Baja | Alta |
| 11 | Members_TLGGA_View | 7 | Media | Media | Media |
| 12 | Members_TLGGA | 6 | Media | Media | Media |

Donde:

- ✓ Cant. Proc: cantidad de procedimientos
- ✓ Resp: responsabilidad
- ✓ Complej: complejidad
- ✓ Reutil: reutilización

Se trabajó con un total de 12 clases y un promedio de procedimientos de 4.5, obteniéndose como resultados los datos que a continuación se muestran:

Tabla 10 Resultados de la aplicación de la métrica TOC

| Cantidad de clases: 12 | Baja | Media | Alta |
|--------------------------------------|-------------|--------------|-------------|
| Responsabilidad | 50 % | 50 % | 0 % |
| Complejidad de implementación | 50 % | 50 % | 0 % |

Capítulo 3. Implementación y Prueba

| | | | |
|---------------|-----|------|------|
| Reutilización | 0 % | 50 % | 50 % |
|---------------|-----|------|------|

Luego de aplicar la métrica TOC y de alcanzar los resultados anteriores se llega a la conclusión de que las clases que forman parte del sistema están dentro de la categoría baja y media para los atributos responsabilidad y complejidad de implementación con un 50 % del total cada una y además se determina que existe una alta y media reutilización en el diseño propuesto. Por lo tanto, se concluye que los resultados obtenidos según esta métrica son positivos.

3.9.2 Relaciones entre Clases (RC)

Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad (68):

Acoplamiento: un aumento del RC implica un aumento del Acoplamiento de la clase.

Complejidad de mantenimiento: un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.

Cantidad de pruebas: un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Tabla 11 Rango de valores para la evaluación de los atributos de calidad

| Atributo | Categoría | Criterio |
|--------------------------------------|-----------|--|
| Acoplamiento | Ninguno | 0 |
| | Bajo | 1 |
| | Medio | 2 |
| | Alto | >2 |
| Complejidad del Mantenimiento | Baja | Cantidad de relaciones de uso \leq Promedio |
| | Media | Promedio \leq Cantidad de relaciones de uso \leq 2* Promedio |
| | Alta | Cantidad de relaciones de uso $>$ 2* Promedio |
| Reutilización | Baja | Cantidad de relaciones de uso $>$ 2* Promedio |
| | Media | Promedio \leq Cantidad de relaciones de uso \leq 2* Promedio |
| | Alta | Cantidad de relaciones de uso \leq Promedio |
| | Baja | Cantidad de relaciones de uso \leq Promedio |

Capítulo 3. Implementación y Prueba

| | | |
|----------------------------|-------|--|
| Cantidad de Pruebas | Media | Promedio \leq Cantidad de relaciones de uso \leq 2* Promedio |
| | Alta | Cantidad de relaciones de uso $>$ 2* Promedio |

Tabla 12 Aplicación de la métrica RC

| No | Clase | Cant. Relac | Acop | Complej. Mant | Reutil | Cant. Prueb |
|----|----------------------------|-------------|-------|---------------|--------|-------------|
| 1 | Obra_TLGGA_Controller | 2 | Medio | Media | Media | Media |
| 2 | Obra_TLGGA_List_Table | 1 | Bajo | Baja | Alta | Baja |
| 3 | Obra_TLGGA_View | 2 | Medio | Media | Media | Media |
| 4 | Obras_TLGGA | 1 | Bajo | Baja | Alta | Baja |
| 5 | Coleccion_TLGGA_Controller | 2 | Medio | Media | Media | Media |
| 6 | Coleccion_TLGGA_List_Table | 1 | Bajo | Baja | Alta | Baja |
| 7 | Coleccion_TLGGA_View | 2 | Medio | Media | Media | Media |
| 8 | Coleccion_TLGGA | 1 | Bajo | Baja | Alta | Baja |
| 9 | Member_TLGGA_Controller | 2 | Medio | Media | Media | Media |
| 10 | Members_TLGGA_List_Table | 1 | Bajo | Baja | Alta | Baja |
| 11 | Members_TLGGA_View | 2 | Medio | Media | Media | Media |
| 12 | Members_TLGGA | 1 | Bajo | Baja | Alta | Baja |

Donde:

- ✓ Cant. Relac: cantidad de relaciones de uso
- ✓ Acop: acoplamiento
- ✓ Complej. Mant: complejidad de mantenimiento
- ✓ Reutil: reutilización
- ✓ Cant. Prueb: cantidad de pruebas

Capítulo 3. Implementación y Prueba

Se trabajó con un total de 12 clases y un promedio de asociaciones de uso de 1.5, obteniéndose como resultados los datos que a continuación se muestran:

Tabla 13 Resultados de la aplicación de la métrica RC

| Cantidad de clases: 12 | Baja | Media | Alta |
|-------------------------------------|-------------|--------------|-------------|
| Acoplamiento | 50 % | 50 % | 0 % |
| Complejidad de mantenimiento | 50 % | 50 % | 0 % |
| Reutilización | 0 % | 50 % | 50 % |
| Cantidad de pruebas | 50 % | 50 % | 0 % |

Los resultados obtenidos tras la aplicación de la métrica RC demuestran que las clases del diseño poseen un bajo y medio acoplamiento con un 50 % del total cada una. Los atributos complejidad de mantenimiento y cantidad de pruebas, sumaron un 100 % en las categorías baja y media, lo que demuestra que no es necesario un elevado esfuerzo a la hora de realizar cambios, correcciones y pruebas al software, y además se determina que existe un alta y media reutilización en el diseño propuesto. Por lo tanto, se concluye que los resultados obtenidos según esta métrica son positivos.

3.10 Pruebas

Con las pruebas se pretende verificar y validar que el software cumple con todas las expectativas y requisitos definidos por el cliente. Se decidió realizar pruebas funcionales y pruebas de aceptación por cada historia de usuario definida. Además se realizaron las pruebas de carga y estrés sobre varias vistas del sistema.

3.10.1 Pruebas funcionales

En el nivel de unidad se realizaron pruebas funcionales a la aplicación con el método pruebas de caja negra y la técnica partición de equivalencia, estas fueron realizadas para verificar que el sistema cumple con los requisitos definidos. Para su ejecución se diseñan casos de prueba basados en los requisitos del sistema y para cada caso se definen los posibles escenarios a probar, que pueden ser que la entrada de los datos sea correcta, que existan campos obligatorios vacíos o que en la entrada de los datos haya campos incorrectos.



Tabla 14 Caso de prueba SC Adicionar obra literaria personal.

| Escenario | Descripción | Título | Descripción | Categoría | Autor | Compartir | Adjuntar archivo | Respuesta del sistema | Flujo central |
|---|--|-------------------------|---|-------------|-----------|-------------------|--|---|---|
| EC 1.1 Adicionar obra literaria personal correctamente. | Permite adicionar una obra literaria personal con sus datos correspondientes. | V El ratoncito Pérez | V Érase una vez Pepito Pérez, que era un pequeño ratoncito | V Cuento | V Lali | V Obra pública | V C:\xampp\htdocs\tallerliterario\wp-content\uploads\cuento-ratn-prez-1 | El sistema permite adicionar una obra literaria personal correctamente. | Acceder a la administración del portal y en el panel de administración ir al |
| EC 1.2 Adicionar obra literaria personal con campos incompletos. | El sistema no permite adicionar una obra literaria personal porque existen campos obligatorios vacíos. | I | V Érase una vez Pepito Pérez, que era un pequeño ratoncito | V Cuento | V Lali | V Obra pública | V C:\xampp\htdocs\tallerliterario\wp-content\uploads\cuento-ratn-prez-1 | El sistema muestra un mensaje de error informando que existen campos que son obligatorios vacíos. | menú Gestionar Obras, Crear nueva Obra Literaria, llenar los campos para crear la nueva obra y dar clic en el botón Publicar. |
| | | V El ratoncito Pérez | I | V Cuento | V Lali | V Obra pública | I C:\xampp\htdocs\tallerliterario\wp-content\uploads\cuento-ratn-prez-1 | | |

3.10.1.1 Resultados de las pruebas funcionales

Luego de una primera iteración de las pruebas se detectaron un total de 14 no conformidades (NC), las cuales fueron solucionadas por el equipo de desarrollo resultando una segunda iteración sin NC.

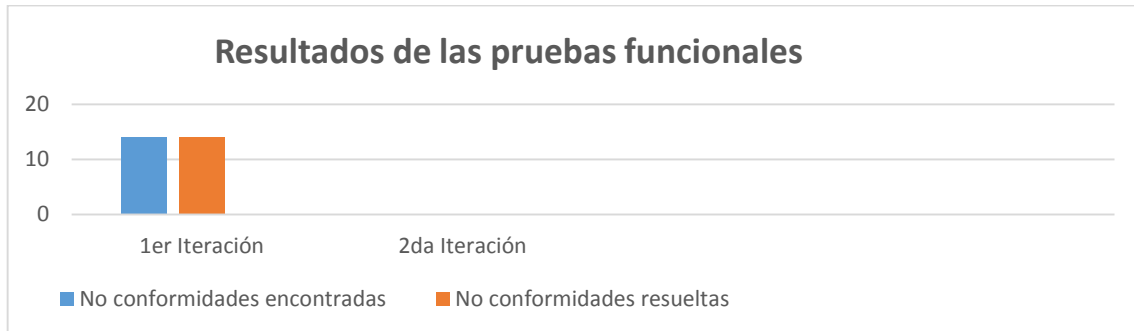


Figura 14 Resultados de las pruebas funcionales

3.10.2 Pruebas de aceptación

En el nivel de aceptación se realizaron pruebas de aceptación, las que aseguran el comportamiento del sistema y especifican los aspectos a probar cuando una historia de usuario ha sido correctamente implementada. Son realizadas para validar que los requisitos implementados satisfagan al cliente. Este tipo de prueba debería ser definida por el cliente, aunque en la mayoría de los casos y por cuestiones técnicas, toma un papel relevante el equipo de desarrollo (68).

Tabla 15 Caso de prueba Gestionar obra literaria personal

| Caso de prueba de aceptación | |
|---|---|
| Código caso de prueba: HU7_01 | HU: 7, Gestionar obra literaria personal |
| Descripción: se debe validar que en el sistema sea posible adicionar obras literarias personales. | |
| Condiciones de ejecución: el usuario debe de estar autenticado y tener permisos para adicionar obras literarias. | |
| Entradas/ Pasos de ejecución: - Título – Contenido - Imagen destacada - Categoría (marcar) | |
| Resultado esperado: se ha validado correctamente la adición de obras en el sistema. | |
| Evaluación: satisfactorio | |

3.10.2.1 Resultados de las pruebas de aceptación

Al finalizar la etapa de las pruebas se detectaron un total de 24 no conformidades que fueron resueltas en cada una de las iteraciones según correspondía. De los resultados alcanzados y los datos visualizados en el gráfico se puede inferir que las pruebas de aceptación fueron exitosas.

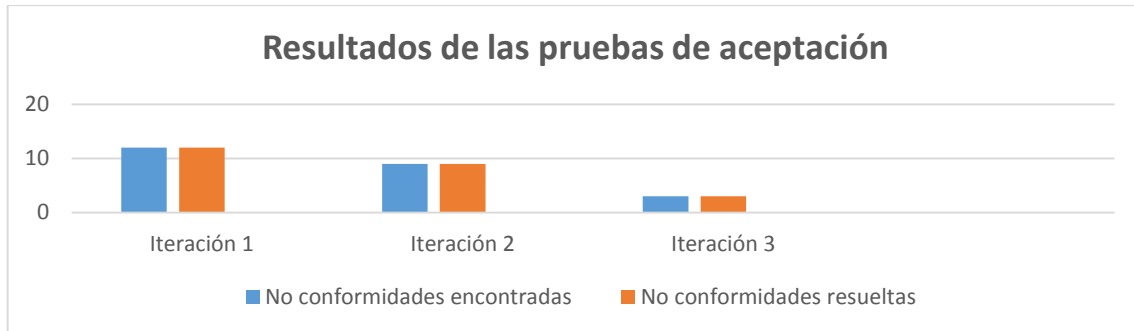


Figura 15 Resultados de las pruebas de aceptación

3.10.3 Pruebas de carga y estrés

En el nivel de sistema se realizaron las pruebas de carga y estrés con el objetivo de verificar el comportamiento del sistema ante altas cantidades de peticiones, permitiendo conocer el tiempo de respuesta por transacción. La herramienta utilizada para la realización de estas pruebas es JMeter⁸ en su versión 2.9, se utiliza para realizar pruebas de rendimiento y pruebas funcionales sobre aplicaciones web.

Entorno de pruebas

Para la realización de las pruebas se utilizaron dos ordenadores, uno servidor y el otro cliente, los cuales cumplen con las siguientes características:

PC servidor

- Microprocesador AMD A8-4500M a 1.90GHz
- Memoria RAM de 4GB
- Disco duro de 640GB
- Sistema operativo Ubuntu 13.10

PC cliente

⁸ Aplicación de escritorio, es un software de código abierto, diseñado para cargar el comportamiento de pruebas funcionales y medir el rendimiento (104).

- Microprocesador AMD Single Core E-240
- Memoria RAM de 512MB
- Disco duro de 500GB
- Sistema operativo Ubuntu 12.10

3.10.3.1 Resultados de las pruebas de carga y estrés

Las pruebas fueron realizadas sobre diferentes vistas del sistema, accediendo a cada una 1000 usuarios simultáneamente, arrojando un error de 0.7% para 3000 conexiones de forma concurrente. A continuación se muestran los resultados:

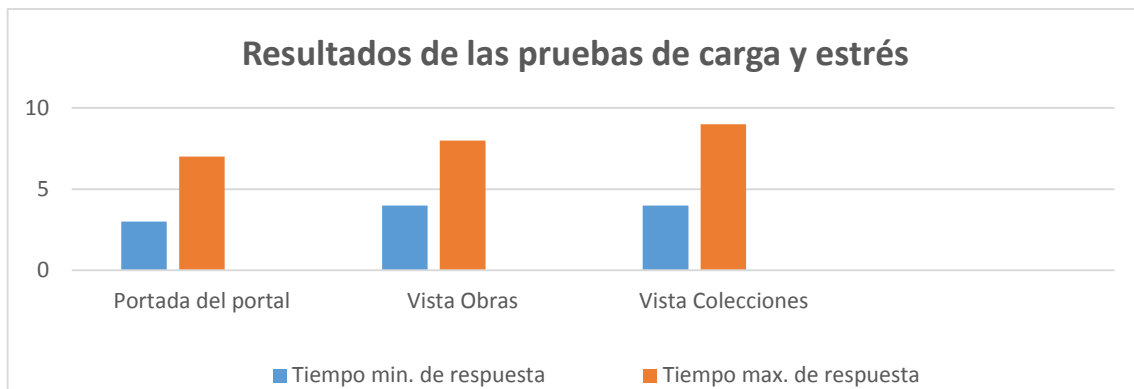


Figura 16 Resultados de las pruebas de carga y estrés

Las pruebas realizadas se consideran exitosas, teniendo en cuenta el entorno de trabajo donde fueron realizadas y los resultados obtenidos. Para 3000 conexiones de forma concurrentes los tiempos de respuestas fueron menores de 10 segundos.

3.11 Disponibilidad

Para evaluar en qué medida aumenta la disponibilidad de la información referente al taller literario con la solución propuesta, se realiza una comparación del comportamiento de dicha disponibilidad en el taller literario tradicional y el Portal web taller literario “Gertrudis Gómez de Avellaneda”.

Para calcular la disponibilidad se aplicará la siguiente fórmula (9):

$$D = \left(\frac{A - B}{A} \right) * 100$$

Donde:

D= Disponibilidad

Capítulo 3. Implementación y Prueba

A = Horas comprometidas de disponibilidad

B = Número de horas que la información no está disponible (fuera de línea).

Para el taller literario tradicional:

$$A = 3 * 42 = 126 \text{ horas/ año}$$

$$B = 6 \text{ horas por días feriados} + 6 \text{ horas por problemas personales del profesor del taller} \\ = 12 \text{ horas/ año}$$

$$D = \left(\frac{126 - 12}{126} \right) * 100 = 90.48\%$$

Para el Portal web taller literario “Gertrudis Gómez de Avellaneda”:

$$A = 24 * 365 = 8,760 \text{ horas/ año}$$

$$B = 24 \text{ horas por fallas eléctricas}$$

$$D = \left(\frac{8,760 - 24}{8,760} \right) * 100 = 99.73\%$$

Con estos resultados se puede apreciar un aumento de la disponibilidad de la información asociada a los talleres literarios y todo previsto con el aumento de la variable horas comprometidas de disponibilidad, pues se pasa de 3 horas a la semana durante todo el año (excluyendo las vacaciones de verano y fin de año) a 24 horas todos los días del año.

3.12 Conclusiones parciales

Se describió la arquitectura de la tecnología utilizada para la solución propuesta y la arquitectura Cliente/ Servidor. También se analizaron las extensiones necesarias para la implementación de la solución, así como para garantizar la seguridad y se detallaron las funciones, vistas y el tema a utilizar. Además se describieron las tareas de ingeniería para dar cumplimiento a los requisitos identificados. Se aplicaron las métricas para la validación de los requisitos y del diseño, se comprobó que la variable disponibilidad presente en el problema planteado se eleva con la solución propuesta y por último se realizaron pruebas al sistema para comprobar su correcto funcionamiento, dando como resultando un sistema satisfactorio.

CONCLUSIONES GENERALES

Una vez desarrollado el Portal web taller literario “Gertrudis Gómez de Avellaneda” se puede plantear las siguientes conclusiones:

- Los conceptos y aspectos teóricos abordados en la primera etapa de la investigación hicieron posible un mejor entendimiento de las herramientas y tecnologías utilizadas en la actualidad para el desarrollo de portales web.
- El uso del CMS WordPress acompañado de la metodología XP aportó gran rapidez, flexibilidad y organización en el proceso de desarrollo.
- El diseño e implementación del Portal permitió la gestión centralizada de toda la información referente al taller literario “Gertrudis Gómez de Avellaneda”.
- Las pruebas de aceptación y pruebas al sistema comprobaron el buen funcionamiento del mismo de acuerdo con los requisitos planteados.



RECOMENDACIONES

Luego de concluido el presente trabajo se recomienda:

- Mantener actualizado el CMS Wordpress con la última actualización disponible con el fin de eliminar fallas de seguridad.
- Extender el funcionamiento del Portal web taller literario “Gertrudis Gómez de Avellaneda” a todo el país.

REFERENCIAS BIBLIOGRÁFICAS

1. **Medina, Juan G. Mendoza.** Radio Cadena Agramonte. [En línea] 10 de 02 de 2011. [Citado el: 10 de 11 de 2014.] <http://www.cadenagramonte.cu/articulos/ver/12688:casas-de-cultura-tres-decadas-en-defensa-del-arte-y-la-revolucion>.
2. **Coordinación de talleres literarios – Didáctica de la escritura creativa.** Coordinación de talleres literarios – Didáctica de la escritura creativa. [En línea] [Citado el: 10 de 11 de 2014.] <https://uacmtalleresliterarios.wordpress.com/que-es-un-taller-literario/>.
3. **Aliaga Benavides, Adriel Alejandro.** Serie Científica de la Universidad de las Ciencias Informáticas. [En línea] 04 de 05 de 2011. [Citado el: 28 de 04 de 2015.] <http://publicaciones.uci.cu/index.php/SC/article/view/369/396>.
4. *Gestión documental en las empresas : una aproximación práctica.* **Bustelo Ruesta, Carlota.** Bilbao : FESABID, 2000, Jornadas Españolas de Documentación, Vol. 7, págs. 189-195.
5. **Aja Quiroga, , Lic. Lourdes.** Scielo. [En línea] 05 de 10 de 2002. [Citado el: 27 de 04 de 2015.] http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352002000500004.
6. **Delgado, M.C. Gabino Estevez y Ing. Eduardo Ochoa Hernández.** . Universidad Michoacana de San Nicolás de Hidalgo. [En línea] 2001. [Citado el: 14 de 11 de 2014.] <http://dieumsnh.qfb.umich.mx/gesinfo/>.
7. *Gestión documental, gestión de información y gestión del conocimiento: evolución y sinergias.* **Ponjuán-Dante, Gloria.** 3, La Habana : Instituto de Información Científica y Tecnológica, 2005, Ciencias de la Información, Vol. 36.
8. **Yáñez, María Rebeca y Villatoro S., Pablo.** *Las nuevas tecnologías de la información y de la comunicación (TIC) y la institucionalidad social. Hacia una gestión basada en el conocimiento.* Chile : s.n., 2004.
9. **Everac99.** Everac99. [En línea] 19 de 08 de 2008. [Citado el: 10 de 05 de 2015.] <https://everac99.wordpress.com/2008/08/19/alta-disponibilidad-que-es-y-como-se-logra/>.

Referencias Bibliográficas

10. **A. Voos, Ing. Javier, Gonzalez, Ing. Eduardo y Cagnolo, Ing. Fernando.** *Portal de Aplicaciones Médicas.* Argentina : Universidad Tecnológica Nacional Facultad Regional Córdoba, 2012.
11. **Figueroa, Valls.** *Evaluación y análisis de la calidad percibida en destinos turísticos sol y playa.* La Habana : CUJAE, 2006.
12. **Buyto.** Buyto. [En línea] [Citado el: 28 de 04 de 2015.] <http://www.buyto.es/general-diseno-web/que-es-un-portal-web>.
13. **Vargas, Iván.** WarriorsBlue. [En línea] [Citado el: 28 de 04 de 2015.] <http://warriorsblue15.tripod.com/id1.html>.
14. **Equipo del Producto CMMI.** *CMMI para Desarrollo, Versión 1.3.* s.l. : Software Engineering Institute, 2010.
15. **Blanco Encinosa, Lázaro.** *La informática en la dirección de empresas.* La Habana : Félix Varela, 2011. ISBN 978-959-07-1629-4.
16. **Programación Extrema.** Programación Extrema. [En línea] [Citado el: 29 de 01 de 2015.] <http://www.programacionextrema.org>.
17. **Vera, Macarena.** Intelligence to Business. [En línea] 2014. [Citado el: 2 de 02 de 2015.] <http://www.i2btech.com/blog-i2b/tech-deployment/los-beneficios-de-implementar-la-metodologia-agil/>.
18. **Letelier, Patricio.** *Proceso de desarrollo de software.* Valencia : Universidad Politécnica de Valencia., 2009.
19. **Palacio, Juan.** *Flexibilidad con Scrum.* 2007.
20. **Ingeniería de Software.** Ingeniería de Software. [En línea] [Citado el: 4 de 02 de 2015.] http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
21. **de la Vega García, Erik.** Serie Científica Universidad de las Ciencias Informáticas. [En línea] 2008. [Citado el: 5 de 02 de 2015.] <http://publicaciones.uci.cu/index.php/SC/article/view/73/74>.
22. **Escribano Fernández, Gerardo.** Universidad de Mendoza. [En línea] 2002. [Citado el: 2 de 02 de 2015.] <http://www.um.edu.ar/catedras/claroline/backends/download.php?url=L01ldG9kb3NfQ>

Referencias Bibliográficas

WdpbGVzL1Byb2dyYW1hY2lvbl9FeHRyZW1hLVhQLnBkZg%3D%3D&cidReset=true
&cidReq=II0162004.

23. **Programación Extrema.** Programación Extrema. [En línea] [Citado el: 5 de 02 de 2015.] <http://programacionextrema.tripod.com/fases.htm>.

24. **R. Villafuerte, Victor.** Extreme Programming. [En línea] [Citado el: 2 de 02 de 2015.] <http://extremeprogramming.host56.com/PRINCIPIOS.php>.

25. **H. Canós, José, Letelier, Patricio y Penadé, M^a Carmen .** *Métodologías Ágiles en el Desarrollo de Software*. Valencia : DSIC -Universidad Politécnica de Valencia, 2012.

26. **Centro de Apoyo Tecnológico a Emprendedores, Fundación Parque Científico y.** *Estudios de los sistemas de gestión de contenidos web*. Castilla-La Mancha : Junta de Comunidades de Castilla-La Mancha, 2012.

27. **Pérez Formantín, Yusleidys y Peraza Sierra, Indira.** Repositorio Institucional. [En línea] 2007. [Citado el: 19 de 11 de 2015.] http://repositorio_institucional.uci.cu/jspui/handle/ident/TD_0912_07.

28. **Hidalgo Urbino, Ing. Rafael Jacobo .** Conferencia. CMS Wordpress. Buenas practicas en la web. [En línea] 26 de 10 de 2011. [Citado el: 28 de 04 de 2015.] <http://www.recreacomunicacion.com/blog/ventajas-e-inconvenientes-de-usar-cms-de-codigo-abierto-wordpress-drupal-o-joomla/>.

29. **Aliaga Benavides, Adriel Alejandro.** Serie Científica de la Universidad de las Ciencias Informáticas. [En línea] 04 de 05 de 2011. [Citado el: 23 de 11 de 2015.] <http://publicaciones.uci.cu/index.php/SC>.

30. **Drupal.org.** Drupal.org. [En línea] 11 de 04 de 2005. [Citado el: 6 de 02 de 2015.] <http://drupal.org.es/drupal>.

31. **Somoza, J. M.** Pyme Networks. [En línea] 14 de 09 de 2013. [Citado el: 26 de 01 de 2015.] <http://www.pymenetworks.es/blog/ventajas-y-desventajas-de-los-cms-iii-drupal>.

32. **Graf, Hagen.** Cocoate. [En línea] 03 de 01 de 2012. [Citado el: 5 de 12 de 2015.] <http://cocoate.com/15/02/joomla-development-beginners-guide>.

33. **Web Design.** Web Design. [En línea] 23 de 04 de 2014. [Citado el: 5 de 12 de 2015.] <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>.

Referencias Bibliográficas

34. **Portal para dar consejos útiles para desarrollo web.** *Professional Word Press , 2nd Edition.* s.l. : John Wiley & Sons, 2012.
35. **Wordpress.org.** Wordpress.org. [En línea] [Citado el: 8 de 04 de 2015.] <https://es.wordpress.org/>.
36. **Web empresa.** Web empresa. [En línea] [Citado el: 28 de 04 de 2015.] <http://www.webempresa.com/caracteristicas-generales-de-los-packs-wordpress.html>.
37. **Esfera hosting.** Esfera hosting. [En línea] [Citado el: 28 de 04 de 2015.] <http://www.esferahosting.com/wordpress>.
38. **Brazell, Aaron.** *WordPress Bible.* Indiana : Mark Jaquith, 2010.
39. **Un web para todos.** Un web para todos. [En línea] [Citado el: 28 de 04 de 2015.] <http://www.unwebparatodos.net/que-es-wordpress>.
40. **MATO GARCIA, LIC. ROSA MARIA.** *DISEÑO de BASES DE DATOS.* 1999.
41. **Gilfillan, Ian.** *La Biblia De Mysql.* s.l. : ANAYA MULTIMEDIA, 2006.
42. **Rodríguez, José Antonio.** *Tutorial de PHP y My SQL completo.* 2005.
43. **Gallego Vázquez, José Antonio.** *Desarrollo Web con PHP y My SQL.* s.l. : ANAYA MULTIMEDIA, 2006.
44. **Terán Pomier, Jorge Humberto.** *Fundamentos de la programación.* Bolivia : Universidad Mayor de San, 2007.
45. **Programacion Web.** Programacion Web. [En línea] Instituto Tecnológico de Matehuala. [Citado el: 28 de 04 de 2015.] <https://programacionwebisc.wordpress.com/2-3-lenguajes-de-programacion-del-lado-del-servidor/>.
46. **PHP.** PHP. [En línea] [Citado el: 19 de 01 de 2015.] <http://php.net/manual/es/intro-what-is.php>.
47. **Marley , Jimi.** Programación en castellano. [En línea] [Citado el: 28 de 04 de 2015.] http://programacion.net/articulo/por_que_elegir_php_143.
48. **de la Torre, Aníbal.** Adelat. [En línea] 2006. [Citado el: 20 de 01 de 2015.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.

Referencias Bibliográficas

49. **Der Henst S, Christian Van y Vega, John Freddy** . *Guía HTML5- El presente de la web. HTML5, css3 y javascript*. s.l. : CRISTALAB, 2011.
50. **Eguiluz, Javier**. *Desarrollo web ágil con Symfony 2.3*. 2013.
51. **Navajas Ojeda, Antonio**. Etnassoft. [En línea] 2012. [Citado el: 27 de 01 de 2015.] <http://www.etnassoft.com/biblioteca/guia-completa-de-css3/>.
52. **Eguíluz Pérez, Javier** . *Introducción javascript*. 2009.
53. **Rodríguez Trujillo, Isyed De La Caridad**. *Javascript y la programación orientada a objetos*. s.l. : Serie Científica de la Universidad de las Ciencias Informáticas, 2011.
54. **JQuery**. JQuery. [En línea] 2013. [Citado el: 29 de 01 de 2015.] <http://jquery.com/>.
55. **Pompa T, Enrique Roberto y Cabrera González, Lianet**. *Extensión de Visual Paradigm for UML para el desarrollo dirigido por modelos de aplicaciones de gestión de información*. s.l. : Serie Científica de la Universidad de las Ciencias Informáticas, 2012.
56. **Unified Modeling Language**. Unified Modeling Language. [En línea] [Citado el: 26 de 04 de 2015.] <http://www.uml.org/>.
57. **Sierra , Manuel**. Aprende a programar. [En línea] [Citado el: 30 de 04 de 2015.] http://aprenderaprogramar.com/index.php?option=com_content&view=article&id=542:que-es-un-servidor-y-cuales-son-los-principales-tipos-de-servidores-proxydns-webftppop3-y-smtp-dhcp&catid=57:herramientas-informaticas&Itemid=179.
58. **Libros Web**. Silex y los servidores web. [aut. libro] Fabien Potencier Igor Wiedler. *Silex, el manual oficial*.
59. **J. Kabir, Mohammend**. *La Biblia Server Apache*. s.l. : ANAYA MULTIMEDIA, 2004.
60. **Apache web**. Apache web. [En línea] [Citado el: 28 de 04 de 2015.] http://ldc.usb.ve/~emilio/Portafolio/Software/REDES3/G5b/Presenta_Site/caracteristica_s_pyc.htm.
61. **Apache-openSUSE**. Apache-openSUSE. [En línea] [Citado el: 28 de 04 de 2015.] <https://es.opensuse.org/Apache>.
62. **The Apache Software Foundation**. The Apache Software Foundation. [En línea] [Citado el: 15 de 05 de 2015.] <http://jmeter.apache.org/>.

Referencias Bibliográficas

63. **Chaudhary, Mukund y Kumar, Ankur.** *Php Storm Cookbook*. s.l. : Packy Publishing, 2015.
64. **JetBrains.** JetBrains. [En línea] [Citado el: 30 de 04 de 2015.] <https://www.jetbrains.com/phpstorm/>.
65. **Pérez, Jorge, y otros.** Fundamentos de ingeniería. [En línea] 06 de 03 de 2012. [Citado el: 1 de 05 de 2015.] <http://fundamentos-ing-requisitos.blogspot.com/2012/03/metricas-y-herramientas-para-la.html>.
66. *Propuesta de una guía de métricas para evaluar el desarrollo de los Sistemas de Información Geográfica.* **Sánchez Fornaris.** 2010, Vinculando.
67. *Vinculando.* **Universidad de las Ciencias Informáticas.** 04 de 01 de 2010.
68. **S. Pressman, Roger.** Capítulo 15 Métricas del producto para el software. *Ingeniería de Software. Un Enfoque Práctico. 5ta Edición.*
69. **Ril Gil, Yohandri y Toll Palma,, Yuniet del Carmen.** Serie Científica. [En línea] 2008. [Citado el: 29 de 04 de 2015.] <http://publicaciones.uci.cu/index.php/SC/article/view/81/82>.
70. **Oré B, Ing. Alexander.** Calidad y Software. [En línea] [Citado el: 29 de 04 de 2015.] http://www.calidadyssoftware.com/testing/pruebas_funcionales.php.
71. **Pruebas de software.** Pruebas de software. [En línea] [Citado el: 2 de 05 de 2015.] <http://pruebasdesoftware.com/pruebadeacceptacion.htm>.
72. **S. Pressman, Roger.** Factores de calidad ISO. *Ingeniería de Software. Un Enfoque Práctico. 5ta Edición.* s.l. : Mc Graw Hill, 2010.
73. **Díaz Gutiérrez, Jorge Antonio, Lorente Rodríguez, Abel Ernesto y Ferrer Obregón, Roberto.** *Desarrollo de un IDE libre y multiplataforma para la creación de componentes visuales de ActionScript para Software Educativo : codeDraw.* 2009.
74. **Larman, Craig.** *UML y Patrones.* s.l. : Pearson, 2004.
75. **Cheers, Martin Dodge.** An Atlas of CyberSpaces. [En línea] 2007. [Citado el: 25 de 02 de 2015.] http://personalpages.manchester.ac.uk/staff/m.dodge/cybergeography//atlas/web_sites.html.

Referencias Bibliográficas

76. **Tedeschi, Nicolás.** Microsoft Developer Network. [En línea] [Citado el: 5 de 05 de 2015.] <https://msdn.microsoft.com/es-es/library/bb972240.aspx>.
77. **Guerrero, Carlos A, Suárez, Johanna M y Gutiérrez, Luz E.** Scielo. [En línea] 2013. [Citado el: 2 de 05 de 2015.] http://www.scielo.cl/scielo.php?pid=S0718-07642013000300012&script=sci_arttext.
78. **CristaLab.** CristaLab. [En línea] 17 de 08 de 2011. [Citado el: 15 de 05 de 2015.] <http://www.cristalab.com/tutoriales/patrones-de-diseno-creacionales-c99932/>.
79. **Pérez Mariñán, Martín.** Patrones de Diseño. [En línea] 3 de 03 de 2012. [Citado el: 28 de 02 de 2015.] <http://www.buenastareas.com/ensayos/Patrones -De-Dise%C3%B1o/2552392.html>.
80. **Cortez, Alberto, Garis, Ana Gabriela y Riesco, Daniel Eduardo.** Sedici. [En línea] 05 de 2011. [Citado el: 18 de 05 de 2015.] <http://sedici.unlp.edu.ar/handle/10915/20072>.
81. **WordPress.org.** WordPress.org. [En línea] [Citado el: 2 de 05 de 2015.] https://codex.wordpress.org/Site_Architecture_1.5.
82. **WordPress.org.** WordPress.org. [En línea] [Citado el: 06 de 05 de 2015.] https://codex.wordpress.org/Stepping_into_Templates.
83. **WordPress.org.** WordPress.org. [En línea] [Citado el: 2 de 05 de 2015.] http://codex.wordpress.org/Hardening_WordPress.
84. **WordPress.org.** WordPress.org. [En línea] [Citado el: 2 de 05 de 2015.] https://codex.wordpress.org/Determining_Plugin_and_Content_Directories.
85. **De La Torre Llorente, César, y otros.** *Guía de Arquitectura N Capas orientada al Dominio con NET 4.0.* s.l. : Krasis Press, 2010.
86. **Herramientas web para la enseñanzade proteocolos de comunicación.** Herramientas web para la enseñanzade proteocolos de comunicación. [En línea] [Citado el: 05 de 05 de 2015.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>.
87. **Doria, Mario.** Scribd. [En línea] [Citado el: 26 de 03 de 2015.] <http://es.scribd.com/doc/106486385/REUTILIZACION-DE-CODIGOS>.
88. **WordPress.org.** WordPress.org. [En línea] [Citado el: 3 de 05 de 2015.] <https://wordpress.org/plugins/wp-pagenavi/>.

Referencias Bibliográficas

89. **WordPress.org.** WordPress.org. [En línea] [Citado el: 03 de 05 de 2015.] <https://wordpress.org/plugins/disable-google-fonts/>.
90. **Pérez Cuello, Yosján.** *Presentando el plugin UCI All, para Wordpress.* 2014.
91. **WordPress.org.** WordPress.org. [En línea] [Citado el: 03 de 05 de 2015.] <https://wordpress.org/plugins/backupwordpress/>.
92. **Wordpress.org.** Wordpress.org. [En línea] [Citado el: 03 de 03 de 2015.] http://codex.wordpress.org/es:Theme_Development..
93. **Coordinación de seguridad de la información.** Coordinación de seguridad de la información. [En línea] [Citado el: 05 de 05 de 2015.] <http://www.seguridad.unam.mx/documento/?id=17>.
94. **Wordpress.org.** Wordpress.org. [En línea] [Citado el: 05 de 05 de 2015.] <https://wordpress.org/plugins/ultimate-security-checker/>.
95. **WordPress.org.** WordPress.org. [En línea] [Citado el: 25 de Abril de 2015.] <https://wordpress.org/plugins/limit-login-attempts/>.
96. **Pressman, Roger S.** *Ingeniería de Software. Un enfoque práctico. 5ta edición. s.l.* s.l. : McGraw-Hill, 2002. ISBN: 8448132149.
97. **El desván de la memoria.** El desván de la memoria. [En línea] [Citado el: 21 de Noviembre de 2014.] www.tallerliterario.net.
98. **Fuentetaja.** Fuentetaja. [En línea] [Citado el: 20 de Noviembre de 2014.] www.fuentetaja.com.
99. **Palabras.** Palabras. [En línea] [Citado el: 22 de Noviembre de 2014.] www.tallerpalabras.com.
100. **Escritores.org.** Escritores.org. [En línea] [Citado el: 24 de Noviembre de 2014.] www.escritores.org.
101. **Cuba Literaria.** Cuba Literaria. [En línea] [Citado el: 23 de Noviembre de 2014.] www.cubaliteraria.cu.
102. **Bustos Salinas, Ariel .** Ingeniería Civil en Computación e Informática, Metodologías Modernas del Software, Ciclo de vida XP. [En línea] 21 de 04 de 2010.

Referencias Bibliográficas

[Citado el: 4 de 02 de 2015.] http://ingeniero-arielbustos.blogspot.com/2010_04_01_archive.html.

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

104. **The Apache Software Foundation.** The Apache Software Foundation. [En línea] [Citado el: 06 de 05 de 2015.] <http://jmeter.apache.org/>.

105. **W3C.** W3C. [En línea] [Citado el: 06 de 04 de 2015.] <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>.

106. **w3schools.com.** w3schools.com. [En línea] [Citado el: 27 de Abril de 2015.] http://www.w3schools.com/js/js_htmldom.asp.

107. **Libros Web.** Libros Web. [En línea] [Citado el: 9 de Abril de 2015.] <https://librosweb.es/libro/ajax/>.

108. **Pérez Mallea, Ivan y Iznaga Benitez, Arsenio M.** Fundamentos de la Grafica por Computadora. [En línea] 2006. [Citado el: 16 de Abril de 201.] <http://sunshine.prod.uci.cu/book/53221cf80571742cb4000008/>.