

Universidad de las Ciencias Informáticas

Facultad 4



*“Sistema de gestión para las colecciones
marinas del Acuario Nacional Cuba”*

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autores:

*Javier Núñez Rivas
Alejandro Suárez Villar*

Tutor:

Ing. Nilber Barbán Góngora

**Ciudad de La Habana, marzo 2012
“Año 54 de la Revolución”**

Declaración de autoría

Por este medio declaramos ser los únicos autores del trabajo Sistema de gestión para las colecciones marinas del Acuario Nacional de Cuba y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente con el mismo.

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

Autores:

Javier Núñez Rivas

Alejandro Suárez Villar

Tutores

Ing. Nilber Barbán Góngora

RESUMEN

El conjunto de colecciones marinas conservadas hoy en día en el Acuario Nacional Cuba constituyen, sin lugar a dudas, una fuente de consulta para investigadores cubanos y foráneos. En la etapa preliminar al triunfo revolucionario muchos de los científicos al mando de expediciones de recolección optaron por depositar sus hallazgos y resultados en museos y fundaciones especializadas, la mayoría de renombre internacional, y en los restantes casos en sus instituciones de origen. Gran parte del material clasificado se dispersó a través de diversos países lo que dificultó más tarde el libre acceso y la disponibilidad física para posteriores estudios de numerosas especies reunidas. Es necesario resaltar que la catalogación de cada uno de los ejemplares se organiza consecutivamente y se actualiza de forma constante y automatizada; aunque la búsqueda de una especie determinada resulta un proceso complejo, pues su localización depende de un extenso grupo de bases de datos implementadas en Microsoft Access, sin comunicación aparente y en las que figuran un aproximado de sesenta mil ejemplares (60000) en total. En la actualidad no se cuenta con una solución o producto informático que supla las necesidades de expertos e interesados en la profundización de estudios relacionados con la conservación de estas colecciones.

Con la misión de solucionar el problema anterior esta investigación plantea como principal objetivo desarrollar un sistema informático que contribuya a la gestión de las colecciones marinas del Acuario Nacional Cuba.

Como resultado del presente trabajo se generan los principales artefactos propios a las fases que establece la metodología de desarrollo de software Programación Extrema (XP). Se despliegan las diversas funcionalidades para solucionar el problema de la investigación y se valida el sistema a través de los métodos definidos en la investigación.

Palabras clave

Gestión de las colecciones marinas, Acuario Nacional Cuba, ejemplar(es).

INTRODUCCIÓN.....	8
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	12
1.1. INTRODUCCIÓN.....	12
1.2 TÉRMINOS BIOLÓGICOS	12
1.2 SISTEMAS DE INFORMACIÓN	13
1.3 SISTEMAS DE GESTIÓN DE CONTENIDOS.....	15
1.3.1 Características	16
1.3.2 Ventajas.....	16
1.3.3 Resumiendo.....	17
1.4 ESTUDIO DE HOMÓLOGOS.....	17
1.5 TECNOLOGÍAS Y HERRAMIENTAS	20
1.5.1. Selección y análisis del CMS a utilizar	21
1.5.2. Joomla	21
1.5.3. Typo3.....	22
1.5.4. Drupal.....	22
1.5.3. Situación del CMS Drupal	23
1.5.4. Características Generales de Drupal	24
1.6. SISTEMA DE GESTIÓN DE BASE DE DATOS	26
1.6.1. PostgreSQL	26
1.6.2. MySQL.....	26
1.6.3. Selección del sistema de gestión de base de datos	27
1.7. SERVIDORES WEB.....	27
1.7.1. Internet Information Services	27
1.7.2. Apache.....	28
1.7.3. Apache Tomcat.....	28
1.7.4. Selección del servidor web.....	28
1.8. LENGUAJES DE PROGRAMACIÓN EMPLEADOS.....	28
1.8.1. PHP	28
1.8.2. HTML.....	29
1.8.3. CSS	29
1.8.4. XML	30
1.8.5. JavaScript.....	30

1.9. METODOLOGÍAS DE DESARROLLO	30
1.9.1. Scrum	33
1.9.2. Proceso Unificado Ágil (Agile Unified Process, AUP)	34
1.9.3. Programación Extrema (XP)	36
1.9.4. Selección de la metodología a utilizar	38
1.10. OTRAS HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO	39
1.10.1. GIMP.....	39
1.10.2. Notepad++	39
1.11. CONCLUSIONES DEL CAPÍTULO	39
CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN	41
2.1. INTRODUCCIÓN.....	41
2.2. FLUJO ACTUAL DE ACTIVIDADES EN EL ACUARIO NACIONAL CUBA.....	41
2.3. PROPUESTA DE SOLUCIÓN	42
2.4. FASES DEFINIDAS POR XP	43
2.5. FASE DE EXPLORACIÓN	43
2.5.1. Historias de usuario	43
2.6. FASE PLANIFICACIÓN.....	45
2.6.1. Estimación de Esfuerzo por Historia de Usuario	46
2.6.2. Plan de iteraciones.....	46
2.6.3. Plan de Duración de Iteraciones	48
2.6.4. Plan de Entregas	48
2.7. PATRÓN ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR	50
2.8. PATRONES DE DISEÑO	50
2.8.1. Programación orientada a objetos en Drupal	50
2.8.2. Patrones	52
2.9. DISEÑO DE LA BASE DE DATOS.....	54
2.9.1. Modelo Entidad-Relación	54
2.10. CONCLUSIONES DEL CAPÍTULO	56
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS	57
3.1. INTRODUCCIÓN.....	57
3.2. DISEÑO DEL SISTEMA	57

Índice de contenido

3.3. MÓDULOS DE DRUPAL	57
3.4. TARJETAS CRC	58
3.5. ESTÁNDARES DE CODIFICACIÓN.....	60
3.5.1. Convenciones de nombres (Funciones, Variables Globales)	60
3.5.2. Indentación y espacios en blanco	61
3.5.3. Comentarios.....	61
3.6. MODELO DE IMPLEMENTACIÓN	62
3.7. FASE DE IMPLEMENTACIÓN	62
3.7.1. Iteración 1	62
3.8. FASE DE PRUEBAS	63
3.8.1. Pruebas unitarias	63
3.8.2. Pruebas de aceptación	64
3.8.3. Pruebas de rendimiento, de carga y estrés del software	65
Pruebas de carga	66
Pruebas de estrés.....	66
Pasos previos a la ejecución.....	66
Ejecución de las pruebas.....	67
3.9. IMPACTO SOCIAL.....	69
3.10. CONCLUSIONES DEL CAPÍTULO	69
CONCLUSIONES GENERALES.....	70
RECOMENDACIONES	71
REFERENCIAS BIBLIOGRÁFICAS	72
BIBLIOGRAFÍA.....	75

Índice de tablas y figuras

Tabla 2.1 Plantilla de Historia de Usuario	44
Tabla 2.2 HU Registrar Usuario	45
Tabla 2.3 HU Gestionar Usuario por Roles	45
Tabla 2.17 Estimación de Esfuerzo por Historias de Usuario	46
Tabla 2.17.1. Estimación de Esfuerzo por Historias de Usuario	47
Tabla 2.17.2. Plan de iteraciones.....	48
Tabla 2.18 Historias de Usuario por Módulos.....	49
Tabla 2.19 Plan de Entrega	49
Tabla 3.21 Tarjeta CRC Usuarios	58
Tabla 3.22 Tarjeta CRC Grupos Taxonómicos.....	59
Tabla 3.23 Tarjeta CRC Noticias.....	59
Tabla 3.24 Tarjeta CRC Mensajes	59
Tabla 3.25 Tarjeta CRC Comentarios	59
Tabla 3.26 Tarjeta CRC Estadísticas	59
Tabla 3.27 Tarjeta CRC Exportar a PDF	60
Tabla 3.28 Tarjeta CRC Búsquedas	60
Tabla 3.29 Módulos abordados en la primera iteración	62
Tabla 3.30 Tarea 1 Módulo Grupos Taxonómicos.....	62
Tabla 3.46 Prueba 1 al Módulo Usuarios	64
Tabla 3.47 Prueba 2 al Módulo Usuarios	65
Figura 1. Comparación entre metodologías ágiles y tradicionales. (43)	32
Figura 2. Ciclo de desarrollo de SCRUM	34
Figura 3. Ciclo de desarrollo de AUP	35
Figura 4. Ciclo de desarrollo de XP	37
Figura 7. Diseño de base de datos	55
Figura 16. Ejemplo de hoja de cálculo para tomar datos durante la ejecución	67
Figura 17. JConsole.....	67
Figura 18. Informe agregado.....	68
Figura 19. Datos registrados en la hoja de cálculo.....	68

INTRODUCCIÓN

Las principales colecciones marinas en Cuba comenzaron a desarrollarse en el Instituto de Oceanología (IDO) desde su fundación en 1965 como resultado de numerosas investigaciones sistemáticas y ecológicas de la plataforma cubana. Desde 2005 se encuentran depositadas en el departamento de colecciones marinas del Acuario Nacional Cuba (ANC), donde curadores y técnicos especializados en los distintos grupos zoológicos garantizan el manejo de las colecciones, mantienen las condiciones climáticas adecuadas, chequeando y controlando diariamente la temperatura y la humedad del local donde se almacenan.

La excelente conservación, el mantenimiento y la organización de las colecciones garantizan la consulta de sus ejemplares entre los especialistas cubanos y de diferentes instituciones del mundo interesados en la sistemática, la taxonomía y la ecología de las distintas especies marinas.

Las colecciones que se conservan hoy en el Acuario Nacional Cuba y que comenzaron a formarse en el Instituto de Oceanología de la Academia de Ciencias de Cuba, hoy Ministerio de Ciencias Tecnología y Medio Ambiente (CITMA) resultan de las más consultadas actualmente en nuestra región, tanto por investigadores cubanos como extranjeros y se consolidaron a través de un proyecto cuyo objetivo principal era el estudio detallado de la fauna marina de Cuba. Dicho proyecto fue iniciado por los investigadores Vassil Zlatarski de la Academia de Ciencias de Bulgaria y Nereida Martínez Estalella perteneciente al IDO.

El elevado número de ejemplares recolectados permite realizar comparaciones morfométricas dentro de una misma especie, variaciones entre diferente ecoregiones y variabilidad de la distribución de las especies. Dicha característica distingue a estas colecciones de otras de referencia que existen en el área; sin embargo desde la década del 70, la sistemática de estos grupos zoológicos ha cambiado en la medida en que esta ciencia se ha desarrollado, por lo tanto nuevas tendencias hacen que las especies descritas se organicen de diferentes formas. Esto ha motivado la formación de una nueva colección siguiendo las tendencias sistemáticas más actuales.

En estas colecciones ocupa un lugar especial las colecciones Tipo con 228 especies de invertebrados y peces encontrados en localidades cubanas, dichos ejemplares han sido de gran utilidad e imprescindibles en cualquier trabajo sistemático ya sea la revisión de un taxón determinado, el registro de nuevas especies para la ciencia o la determinación de una especie en trabajo de inventarios ecológicos o poblacionales.

Sin embargo antes del Triunfo de la Revolución, científicos foráneos que condujeron varias expediciones de colecta en Cuba depositaron el material clasificado en sus instituciones de origen o en aquellas de prestigio internacional. Ello trajo como consecuencia que la mayor parte de los ejemplares recolectados y los ejemplares Tipo de las especies descritas se

encuentran depositados hoy en colecciones de diversas partes del mundo con un limitado acceso, lo cual imposibilita en ocasiones la ejecución de estudios e investigaciones a partir de los mismos, además gran cantidad de referencias donde aparecen las descripciones y las imágenes de estas especies no están a disposición de los especialistas para una consulta, ya sea por antigüedad de las mismas o por el difícil acceso a una revista determinada. Por otra parte está el deterioro de las irremplazables colecciones Tipo debido a su uso continuo.

Vale la pena destacar que cada ejemplar está catalogado de forma consecutiva y la información que lo acompaña se mantiene actualizada en catálogos, en papel y de forma automatizada, pero realizar la búsqueda de una especie, incluyendo los Tipo, es un proceso engorroso, pues hay que localizar esta especie en diferentes bases de datos desarrolladas en Microsoft Office Access, que no se comunican entre sí y donde existen aproximadamente sesenta mil (60000) ejemplares. Por último es necesario destacar que actualmente no existe un producto informático que permita a investigadores e interesados en el tema estudiar y profundizar sus conocimientos, lo cual influye también, en la conservación de las colecciones ya que las consultas a los ejemplares por lo general se tienen que realizar de manera física. Es por ello que la presente investigación forma parte y se encuentra respaldada por el proyecto Solución Integral para el Acuario Nacional Cuba (SIANC), centrado en la tramitación de una urgente respuesta que satisfaga la creciente demanda actual de la institución. A partir de la situación problemática descrita, se propone en la presente investigación el siguiente **problema a resolver**: ¿Cómo contribuir a la gestión de las colecciones marinas del Acuario Nacional Cuba?

Con la misión de solucionar el problema anterior se plantea como **objetivo general**: Desarrollar un sistema informático que contribuya a la gestión de las colecciones marinas del ANC.

El **objeto de estudio** de la investigación lo componen: Los sistemas de gestión de contenidos.

El **campo de acción** se enfoca en un: Sistema de gestión de contenidos para el ANC.

Para guiar la investigación se traza la siguiente **idea a defender**: El desarrollo del sistema de gestión de contenidos para el ANC permitirá la localización de ejemplares en las colecciones marinas y colecciones Tipo con mayor facilidad y rapidez.

A partir del objetivo general expuesto se trazaron los siguientes **objetivos específicos**:

- ✓ Realizar el estudio del estado del arte del objeto de la investigación científica.
- ✓ Realizar el análisis y diseño de la aplicación.
- ✓ Implementar la aplicación.
- ✓ Validar la propuesta presentada.

Como **posibles resultados** al concluir el trabajo de diploma se espera obtener: Un sistema

cuyas funcionalidades respondan a la efectiva gestión de los contenidos correspondientes a las colecciones marinas del Acuario Nacional Cuba, y que además facilite la accesibilidad a las mismas.

Para el desarrollo de la aplicación en correspondencia con los objetivos específicos se han propuesto las siguientes **tareas de investigación**:

- Realización del diseño teórico y metodológico de la investigación con el objetivo de definir el sustento teórico del presente trabajo.
- Realización de un diagnóstico del estado actual del objeto de estudio a partir de la elaboración de entrevistas a clientes y especialistas en el tema.
- Determinación de las características y diseño del sistema.
- Implementación de la propuesta de solución.
- Validación de la propuesta de solución.

MÉTODOS CIENTÍFICOS

Para garantizar un desarrollo óptimo de la investigación se hizo necesario el empleo de métodos científicos. El desarrollo de este trabajo se basó en los métodos siguientes:

Métodos teóricos: permiten estudiar las características del objeto de investigación que no se observan directamente, además crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad. Se aplican en calidad de enfoque general como estrategia. Los utilizados fueron:

- Analítico-Sintético: se utilizó en el estudio de la literatura especializada relacionada con el tema y la exploración de resultados de investigaciones afines, lo que permitió adoptar posiciones teóricas relacionadas con el objeto de investigación.
- Histórico-Lógico: se empleó para el estudio y profundización de la evolución de las aplicaciones, lo que facilitó la indagación de soluciones al problema planteado.

Métodos empíricos: estos métodos explican las características fenomenológicas del objeto, con ellos es posible representar un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. Se aplican como procedimiento en el proceso de investigación. De los métodos empíricos existentes se utilizaron:

- Entrevistas: se realizaron a especialistas para obtener información relacionada con la aplicación y los diferentes conceptos biológicos y zoológicos a utilizar, lo que contribuyó al perfeccionamiento de la propuesta realizada.
- Cuestionarios: se consultó a personal capacitado y con experiencia en el trabajo con las distintas herramientas a utilizar en el proceso de desarrollo del producto.

ESTRUCTURA CAPITULAR

Capítulo I Fundamentación teórica: se realizará un análisis detallado del marco conceptual, donde se tratarán los conceptos principales relacionados con el tema en cuestión. Se analizarán sistemas similares, además del estado del arte de las tecnologías y herramientas a utilizar en el desarrollo de la aplicación.

Capítulo II Exploración y planificación: se describe la propuesta del sistema, se detallan los artefactos generados en la fase Exploración y Planificación, prestando especial atención a las historias de usuario y al plan de iteración. Además, se exponen los aspectos relacionados con el diseño del sistema donde se detallan los artefactos propuestos en esta fase, entre los que se encuentran las Tarjetas CRC.

Capítulo III Implementación y pruebas: se orienta a la implementación del sistema de gestión y los componentes y módulos utilizados. Además, se describen las tareas relacionadas en la fase de implementación. Se exponen los aspectos relacionados con la fase Pruebas. Se detallan los artefactos propuestos en esta fase y se llevan a cabo las pruebas de aceptación solicitadas por el cliente.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. INTRODUCCIÓN

En el presente capítulo, se hace referencia al marco conceptual, citando los principales conceptos relacionados con el tema propuesto. Se realizó un análisis detallado del estado del arte de la gestión de la información, ejemplificado con instituciones nacionales e internacionales que utilizan sistemas de gestión de contenidos. Además de un estudio de las herramientas y las tecnologías posibles a utilizar en el desarrollo de la aplicación. Lo cual posibilitó arribar a conclusiones respecto a lo que fuese más factible utilizar para dar cumplimiento a los objetivos generales de la investigación y solución al problema planteado.

1.2 TÉRMINOS BIOLÓGICOS

Taxón

En biología, un taxón es un grupo de organismos emparentados, que en una clasificación dada han sido agrupados, asignándole al grupo un nombre en latín, una descripción, y un «tipo», de forma que el taxón de una especie es un espécimen o ejemplar concreto. Cada descripción formal de un taxón es asociada al nombre del autor o autores que la realizan, los cuales se hacen figurar detrás del nombre. La ciencia que define a los taxones se llama taxonomía.

La finalidad de clasificar los organismos en taxones formalmente definidos en lugar de grupos informales, es la de proveer grupos cuya circunscripción (esto es, de qué organismos están compuestos) sea estricta y cuya denominación tenga valor universal, independientemente de la lengua utilizada para la comunicación. Nótese que los taxones existen dentro de una clasificación dada, sujeta a cambios y sobre la que pueden presentarse discrepancias; lo que obliga, respecto a ciertas denominaciones problemáticas, a especificar en el sentido de qué autor se está usando el nombre. (1)

Invertebrados

Se denomina invertebrados a todos aquellos animales (reino Animalia) que no se encuadran dentro del subfilo de los vertebrados (Vertebrata) del filo cordados (Chordata). El nombre alude a que, a diferencia de estos últimos, carecen de columna vertebral o notocorda y de esqueleto interno articulado. Agrupa al 95% de todas las especies animales.

Están subdivididos en diez clases: moluscos, cirrípedos, anélidos, cangrejos, arañas, insectos, gusanos, equinodermos, pólipos e infusorios. (2)

Especies Tipo

En taxonomía, el tipo nomenclatural o, simplemente, Tipo es un ejemplar de una especie dada, sobre el que se ha realizado la descripción de la misma y que, de ese modo, valida la publicación de un nombre científico para dicha especie.

El Tipo del nombre de una especie es, por lo general, el espécimen a partir del cual se ha perfilado la descripción que valida el nombre, y que se conserva permanentemente en alguna institución (museo, herbario, centro de investigación), perfectamente siglado y localizable. El Tipo del nombre de un género es la especie sobre la cual se basó la descripción original que validaba dicho nombre. El Tipo del nombre de una familia es el género sobre el cual fue basada la descripción original válida de la familia. En los nombres de taxones de rango superior al de familia no se aplica el principio de tipificación. (3)

En biología se llama Tipo a la especie, género o familia que se usa para definir un grupo taxonómico de mayor rango, como un género, familia u orden, según corresponda. A partir de este tipo se hace la descripción y caracterización del grupo. (3)

Grupos zoológicos

Se dividen en 3 grandes grupos:

Grupo monofilético: es aquel grupo de especies que descienden todas de una misma especie y que incluye todas las especies derivadas de esa especie. Los grupos monofiléticos se basan en una serie de caracteres que se denominan apomorfías o sinapomorfías.

Grupo parafilético: definen los caracteres que forman los grupos parafiléticos. Son los caracteres plesiomórficos.

Grupo polifilético: cada grupo de especies derivan de unas líneas. Son el resultado de un proceso de convergencia. (48)

1.2 SISTEMAS DE INFORMACIÓN

El desarrollo de una institución es, en gran medida, el producto de la precisión y efectividad del sistema de información (SI) que soporte su funcionamiento. Un sistema con información veraz, confiable, precisa y bien estructurada garantizará el éxito de las decisiones de los usuarios. Para obtener estos resultados, es preciso un flujo eficiente e ininterrumpido de información que sustente los procesos institucionales para que estos fluyan con libertad, a partir del quehacer de un personal dotado de los conocimientos necesarios para el desarrollo de sus actividades.

Un SI es la composición de elementos que operan unidos en orden de capturar, procesar, almacenar y distribuir información. Esta información se utiliza generalmente para tomar decisiones, la coordinación, el control y el análisis en una organización. Frecuentemente, el propósito básico del sistema es la gestión de la información. (5)

Para que un SI pueda realizar las tareas planteadas, es necesario que se identifique, antes de su implantación (6):

- ✓ Las necesidades institucionales que justifican su creación.
- ✓ Las fortalezas y debilidades que puedan hacer que el sistema triunfe o fracase.
- ✓ La información que circulará.

- ✓ Las herramientas capaces de capturar, almacenar y recuperar la información.
- ✓ La forma de intercambio de ideas.
- ✓ El personal que se ocupará de su mantenimiento.

Los sistemas de información:

- ✓ Gestionan los datos que almacenan.
- ✓ Gestionan los usuarios que utilizan la información, que además pueden agregarla.
- ✓ Poseen una interfaz en correspondencia con la información que contienen.
- ✓ Integran no sólo datos e información, sino también programas y otros sistemas de información.

Los SI no sólo almacenan datos, sino que estos datos deben procesarse y distribuirse. Para su distribución, deben seguir una representación clara y cómoda para el usuario final.

Este usuario no sólo es consumidor, además puede actuar como productor de información, por lo que el sistema debe poseer un registro de la producción de sus usuarios y en este entorno pudiera integrar varios programas que ayudan al desarrollo del sistema de información, incluso puede un sistema de información ser la conjunción de varios micro sistemas que se integran en uno general, con un formato de presentación común que lo identifique.

El punto de la interfaz es muy importante para la presentación ante el usuario, porque esta puede ayudar o dañar el contacto. Una interfaz amigable, discreta, fácil de usar y de navegación simple tiene muchas ventajas frente a una cargada de información mal delimitada, mal ordenada y con navegación enmarañada y confusa.

La información incluida en un SI debe procesarse por sus responsables antes de su incorporación. Los encargados del funcionamiento del sistema deben recibir la información, así como verificar su procedencia y los programas utilizados para su confección, en aras de incorporarla para su posterior uso.

El hecho de que los SI automatizados dependen en gran medida de personas que procesan la información, autorizan su inclusión y de otros que la publican, esto comenzó a conspirar contra su efectividad y rápida evolución, porque debido a estos procesos, la información se demoraba en estar a disposición de los usuarios finales, al tener que circular por varios especialistas; además, ocurría también que luego de su caducidad, continuaba a la vista en el sitio. Un SI que pierde el hilo del desarrollo y el flujo de trabajo de la institución a la que pertenece, puede considerarse mediocre o inservible.

Los procesos de integración descansan en la idea de utilizar la información como el recurso común a todas las funciones y departamentos y por tanto, la base para crear un flujo de datos que vincule a todas las áreas y niveles de la organización.

La aparición de herramientas de gestión de contenidos vinieron a enmendar los problemas que

presentaban los SI y llevaron a que las instituciones basen sus sistemas de información en sistemas de gestión de contenidos o “content management system” (CMS) por sus siglas en inglés; es por ello, que puede decirse que los sistemas de gestión de contenidos deben su surgimiento, básicamente, a la necesidad de suplir las faltas que se enunciaron antes, junto a otros aspectos como el hecho de que las instituciones comenzaron a ver como insuficientes las soluciones que brindaban los SI basados en páginas web estáticas, necesitadas de especialistas dedicados a su desarrollo. Es por eso, que entre las principales bondades de los CMS está permitir que cualquier persona, sin conocimientos avanzados sobre informática, pueda colocar, modificar o eliminar contenidos del sitio. Esto, sin renunciar a los necesarios controles de calidad que exige un sistema de información que se respete. (6)

1.3 SISTEMAS DE GESTIÓN DE CONTENIDOS.

La idea básica que existe detrás de un CMS es separar la gestión del contenido de su diseño. El diseño de las páginas se almacena en plantillas mientras que el contenido puede almacenarse en bases de datos o ficheros separados.

En este entorno, los usuarios crean sus páginas, con diseños específicos y los administradores de los sitios, en caso de que necesiten cambiar la apariencia del sitio, sólo deben trabajar en las plantillas, sin necesidad de alterar el contenido ni los diseños individuales. Una de las características más importantes de los sistemas de gestión de contenidos es el hecho de que se pueda mantener el sistema, con la colaboración de múltiples usuarios, es decir, que siempre habrá elementos que consultar, sin esperar a que los especialistas publiquen la información, según sus posibilidades y con la prioridad que entiendan, cuando puede que no sea la requerida.

Un sistema de gestión de contenidos es la confluencia de la captación, procesamiento, actualización, representación y consulta de información en una institución, para la toma de decisiones en ambiente web; estas tareas se realizan con la colaboración de todos los usuarios, y el sistema no depende sólo de información aislada, sino también de su dinámica, que tiene carácter continuo, que convierte el proceso en un ciclo productivo, resultante de su retroalimentación. (6)

En la definición de CMS, pueden encontrarse varios puntos coincidentes con los de SI: capturar, actualizar y analizar información pertinente para una institución. Aunque en este sentido los procesos en los SI engloban varios de los puntos de incidencia de los CMS.

Se puede resumir que los sistemas de gestión de contenidos se emplean para crear, procesar, compartir y organizar contenido en forma cooperativa entre los autores y el sistema, donde este último es capaz de ofrecer posibilidades de integrar los contenidos, con independencia de su formato o procedencia, así como guiar su flujo dentro del mismo, y ello ofrece la oportunidad de

que las instituciones naveguen satisfactoriamente mediante la gestión de la información hacia el aprendizaje institucional, como vía invaluable en el cumplimiento de su misión y sus objetivos estratégicos, así como para la reducción de los costos de fricción de este proceso.

Los costos de fricción comprenden los costos directos e indirectos asociados a la implementación y puesta en funcionamiento de un sistema o tecnología sea este informático o de otro tipo. (6)

1.3.1 Características

Entre las características más importantes de los CMS y que constituyen una evolución de los SI tradicionales, está el hecho de que sean manejables por todos los usuarios, todos pueden subir información, manipularla, representarla. Esto podría verse como un problema, en materia de la veracidad y confiabilidad de la información que circula en el sistema, pero lo cierto es que los CMS permiten distribuir funciones a sus usuarios, así siempre existirán individuos encargados de comprobar que se incluye y quién lo incluye, con la opción de rechazar o solicitar la modificación del material.

Este proceso se realiza de forma dinámica, porque cuando un autor o contribuyente envía un material para publicar, los revisores y administradores, al acceder al sistema, este muestra ante ellos una lista con los trabajos pendientes, ellos sólo deben revisarlo y tomar la acción que se requiere. Además, al existir varias personas con estas posibilidades, los contenidos no se agrupan y envejecen sin publicarse. También es posible someter la información a debate para luego mejorarla, sin necesidad de que las personas se reúnan y conversen sobre ello, sino desde su puesto de trabajo y con la facilidad de que cada criterio o comentario realizado quede grabado y perfectamente visible. Esto se resume en la denominada capacidad de implementación de flujos de trabajo propios de este tipo de sistemas. (6)

Una de las características de los CMS y que además constituye una ventaja con respecto a su uso, es el hecho de posibilitar la inmediatez de la publicación y edición de los contenidos, con ello, se obtiene entonces la agilidad necesaria para la divulgación de los contenidos en una institución.

1.3.2 Ventajas

- ✓ Los gestores de contenidos facilitan el acceso a la publicación de contenidos a un rango mayor de usuarios.
- ✓ Permiten que sin conocimientos de programación ni maquetación cualquier usuario pueda añadir contenido en el portal web.
- ✓ Además permiten la gestión dinámica de usuarios y permisos, la colaboración de varios usuarios en el mismo trabajo y la interacción mediante herramientas de comunicación.

- ✓ Los costes de gestión de la información son mucho menor ya que se elimina un eslabón de la cadena de publicación, el maquetador. La maquetación es hecha al inicio del proceso de implantación del gestor de contenidos.
- ✓ La actualización, backup y reestructuración del portal son mucho más sencillas al tener todos los datos vitales del portal y los contenidos en una base de datos estructurada en el servidor.
- ✓ Facilita la capacidad de manejar eficientemente gran cantidad de páginas web.
- ✓ Posibilita trabajar en un ambiente de páginas web interactivas.
- ✓ Permite controlar el acceso de los usuarios al sistema, no sólo mediante su contraseña, sino mediante los permisos asignados a cada uno y la información que incluye, tanto en calidad como en cantidad, posibilitando el crecimiento y desempeño del sistema. (7)

1.3.3 Resumiendo

Para la gestión, administración y publicación basada en páginas web de forma dinámica de un sistema de gestión de contenidos, existen varias herramientas cuya función principal es: brindarle al desarrollador la posibilidad de crear un sitio web capaz de separar el contenido del diseño y almacenarlo en una base de datos.

A través del estudio de varios sistemas, reflejado anteriormente, se concluye que el uso de un SI tradicional basado en páginas web estáticas y necesitadas de especialistas dedicados a su desarrollo no respondería efectivamente a las exigencias planteadas por el cliente, producto a las deficiencias descritas que presentan los mismo, a lo cual se antepone como solución el empleo de un CMS para la gestión, administración y publicación de contenidos basados en páginas web de forma dinámica, posibilitándole a los especialistas del ANC un manejo más cómodo y sencillo de toda la información que se administrará en la aplicación.

1.4 ESTUDIO DE HOMÓLOGOS

Alrededor del mundo existen varias instituciones de muchísimo prestigio internacional que son referencia en los estudios más importantes de botánica, zoología y paleontología, que albergan en sus instalaciones las más completas colecciones y que son en sí mismas fuentes de obligada consulta para dar credibilidad a estudios que trasciendan al reconocimiento internacional, pero a su vez cuentan estos centros, en muchos casos, con políticas privativas empleadas para proteger el mal uso de su información, la violación de los derechos de sus autores y cuidar el prestigio de las instalaciones mismas; políticas, que en ocasiones limitan el acceso a estudios importantes e impiden que centros con menos recursos o prestigio hagan un uso eficiente de la información.

Instituciones internacionales:

Museo de Zoología Comparada de la Universidad de Harvard (MCZ)

Es un centro de investigación y educación centrada en las relaciones comparativas de la vida animal. El museo sirvió como un campo de entrenamiento para los zoólogos profesionales e influyó en muchos museos de Estados Unidos en ciernes a través de esta nueva generación.

Las Investigaciones en el MCZ continúan para iluminar las estructuras de los seres vivos, su clasificación natural y la relación con el entorno, sin dejar de ser una institución líder en investigación zoológica moderna. Sus colecciones actuales se componen de aproximadamente 21 millones de invertebrados fósiles y existentes. (26)

Acuerdo del usuario (algunos ejemplos de condiciones de uso)

De la propiedad intelectual: Harvard se reserva todos los derechos, incluidos los derechos de autor, en el contenido, sujeto a la limitación de que los derechos de autor y otros derechos de propiedad pueden ser ocupados por personas o entidades distintas, o además de, la Universidad de Harvard.

Licencia limitada general: usted no va a reproducir, mostrar, distribuir, ejecutar, o hacer cualquier otro uso de MCZ Contenido de la Revista, sin obtener primero el permiso específico del Museo de Zoología Comparada.

El uso del contenido de tercero: Harvard no garantiza que cualquier uso de contenido de terceros no infrinja los derechos de los propietarios de contenido.

Indemnización: usted se compromete a indemnizar y mantener indemne a la Universidad de Harvard y sus filiales, agentes, oficiales, directores, y empleados de y contra cualquier y toda responsabilidad, pérdidas, reclamaciones, daños, costos y / o acciones (incluyendo honorarios de abogados) que surjan del uso del contenido, el sitio web MCZ, o el Museo de Zoología Comparada de las colecciones.

Declaración de privacidad: el sitio web de MCZ contiene enlaces a otros sitios web que funcionan independientemente de la red de la Universidad de Harvard, y en algunos sitios fuera de la "mcz.harvard.edu" de dominio. El MCZ no se hace responsable por las prácticas de privacidad o el contenido de dichos sitios web.(26)

Museo de Historia Natural de Londres (NHM)

El Museo de Historia Natural es uno de los tres grandes museos de Londres, Inglaterra. El museo es una organización benéfica exenta, y un organismo público no ministerial auspiciada por el Departamento de Cultura, Medios y Deportes.

El museo es el hogar de la vida y las muestras de ciencias de la tierra comprenden unos 70 millones de artículos dentro de cinco colecciones principales: Botánica, Entomología, Mineralogía, Paleontología y Zoología. El museo es un centro de renombre mundial de la investigación, especialista en taxonomía, identificación y conservación. Dada la edad de la

institución, muchas de las colecciones tienen un gran valor histórico y científico, como los especímenes recogidos por Darwin. La Biblioteca Museo de Historia Natural contiene extensos libros, revistas, manuscritos y colecciones de obras de arte relacionadas con el trabajo y la investigación de los departamentos científicos.

El museo es particularmente famoso por su exhibición de esqueletos de dinosaurios, y la ornamentada arquitectura, a veces llamado una catedral de la naturaleza. El Centro Darwin, es en parte diseñado como una moderna instalación para el almacenamiento de las valiosas colecciones. (27)

Términos y condiciones de uso:

Del uso del sitio: usted no puede copiar, reproducir, publicar, descargar, emitir, transmitir, adaptar o utilizar cualquier material del sitio que no sea para su uso personal no comercial. Se prohíbe la transmisión de la exposición, carga, correo electrónico u otro de cualquier material, la publicación, difusión, uso o posesión que infrinja los derechos de cualquier persona o que sea ilegal en cualquier otro aspecto.

Base de datos: usted no puede usar el sitio: para crear una base de datos (ya sea electrónica o no) que incluya el material descargado u obtenido de otra manera desde el sitio, sin el permiso escrito del NHM; para transmitir o redistribuir cualquier material obtenido de la web a terceros.

Del acceso: NHM tendrá el derecho de negarse a permitirle el acceso al sitio o a cualquier parte del mismo en cualquier momento y por cualquier razón sin darle previo aviso.

Enlaces: Algunas de las páginas en el sitio incluye enlaces a sitios web externos. Estos son únicamente para su conveniencia. NHM no los respalda ni es responsable por el contenido de estos sitios web externos.

Ley aplicable y jurisdicción: estos términos y condiciones de uso se registrarán e interpretarán de conformidad con las leyes de Inglaterra y Gales. Cualquier disputa que surja en relación con el sitio estará sujeto a la jurisdicción exclusiva de sus respectivos tribunales. (27)

Como las instituciones anteriormente mencionadas existen otras de prestigio internacional que también comparten un grupo de políticas de privacidad con gran similitud entre ellas, algunas de estas instituciones son:

Museo Nacional de Historia de Washington DC.

Museo de Paleontología de la Universidad de California.

Museo Americano de Historia Natural de New York.

Instituciones nacionales:

Además existen otros centros nacionales como son el **Jardín Botánico Nacional** que cuenta con un grupo de colecciones orientadas principalmente a plantas, clasificadas en cuatro secciones fundamentales: Herbario, Área de plantas vivas, Registros de plantas y Banco de

germoplasmas; colecciones que se basan fundamentalmente en imágenes de los ejemplares y algunas breves descripciones física de su estado en la propia institución. (28)

El **Instituto de Ecología y Sistemática** que cuenta con 5 colecciones fundamentales: Herbario Nacional, Herbario Micológico, Hongos Micorrizógenos Arbusculares, Colecciones Zoológicas y el Centro de Información Científico – Técnica, de las cuales en tres de ellas solo aparecen descripciones generales de cada colección, mientras que las otras dos junto con la base de datos principal de ejemplares están en proceso de desarrollo desde junio de 2010. (29)

Por otra parte también tenemos el **Museo Nacional de Historia Natural** que cuenta con un grupo de colecciones de ciencias naturales que representa la biodiversidad de la naturaleza cubana en un amplio número de ejemplares catalogados que llegan aproximadamente a los veintitrés mil quinientos (23 500), entre las que se cuentan cinco colecciones cerradas e históricas de un valor incalculable desde el punto de vista científico, con un alto número de especies endémicas de Cuba, que actualmente se encuentran en peligro de extinción. (30)

Esta institución cuenta con un grupo de base de datos en las que guarda la información de sus ejemplares y a las que se pueden acceder ante la necesidad de cualquier estudio o investigación pero no cuenta con ningún sistema que permita interactuar con ellas.

A lo cual se concluye que el sistema que se desarrolla como objetivo de la presente investigación es una alternativa ante la inexistencia de un centro cuya gestión informática de la información de colecciones de esta índole sea efectiva y accesible desde un entorno libre.

En los epígrafes que se describen a continuación se hizo un estudio y análisis detallado de las herramientas utilizadas (y sus homólogos) para el desarrollo del sistema, seleccionando las más adecuadas de cada una de ellas, en aras de desarrollar un producto con calidad.

1.5 TECNOLOGÍAS Y HERRAMIENTAS

En la realización del producto, se hizo necesaria la utilización de distintas herramientas, que permitieran el óptimo desarrollo del mismo. Es por ello que se realizó un análisis previo del estado de algunas tecnologías actuales y sus características.

Para la gestión en ambiente web de las distintas colecciones marinas el cliente (Departamento de Colecciones Marinas del Acuario Nacional Cuba) sugirió al equipo de desarrollo el uso del CMS Drupal y ya que existen, además de este, varios sistemas gestores de contenidos con grandes potencialidades que permiten administrar el contenido y el diseño de los sitios web y que son compatibles con un gran número de servidores de base de datos, en el epígrafe que se describe a continuación se mencionan algunos de los CMS más utilizados en la actualidad, incluyendo Drupal, y sus principales características buscando corroborar que la sugerencia hecha por el cliente al equipo es la más adecuada para el desarrollo del sistema.

1.5.1. Selección y análisis del CMS a utilizar

En la actualidad el uso de los sistemas gestores de contenidos se ha intensificado, gracias a las potencialidades y los beneficios que brindan. En el momento de escoger un CMS para la realización de un software, se torna difícil hacer una correcta selección, ya que son muchos y con características diferentes. Algunos resultan más fáciles de trabajar que otros, pero siempre se debe tener presente en la elección un estudio profundo de sus funcionalidades respecto a la administración de contenidos, flexibilidad de organización, robustez de código, seguridad y capacidad de generar páginas dinámicas. (7)

Una investigación detallada acerca del CMS Drupal permitió conocer un poco más acerca de las ventajas que brinda esta herramienta potente y flexible. Teniendo en cuenta que existen otros CMS con funcionalidades útiles y muy eficientes, como son Joomla y Typo3, que son muy utilizados por desarrolladores, por lo que se hizo una investigación comparativa entre los mismos en cuanto a características, requisitos y ventajas.

Joomla y Drupal resultaron ganadores en el 2008 del concurso “Open Source CMS Award” hoy llamado “Premios Pack Open Source” en la categoría “Mejor CMS de código abierto escrito en lenguaje PHP”. En el 2011 se ratifica Joomla como ganador absoluto de este premio mientras Drupal clasifica en dicho evento como primer finalista resultando triunfador en las categorías “Mejor CMS a la Mejor Comunidad”, “El mejor CMS Open Source para mejorar el rendimiento” y “El mejor CMS para la Facilidad de uso”. Además de obtener en 2009 la distinción “Premio del Salón de la Fama en el 2009 Open Source CMS Awards” por sus resultados en años anteriores en este mismo concurso. (7)

Por otra parte, Typo3 también merece ser tomado en cuenta ya que es un CMS que en los últimos años ha aumentado su potencialidad y rendimiento y tiene grandes posibilidades de convertirse en unos de los mejores sistemas de gestión de contenido.

1.5.2. Joomla

Joomla es un CMS de código abierto. Está escrito en lenguaje PHP y desarrollado por un numeroso equipo de programadores. Está preparado para organizar eficientemente los contenidos de su sitio en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores.

Joomla posee una instalación muy sencilla con muchas extensiones y módulos, la documentación es exhaustiva y concisa, la interfaz de administración es muy intuitiva y poderosa, posee el editor WYSIWYG, con opciones de personalización avanzadas, con una gran comunidad de usuarios. En lo que refiere a seguridad, es uno de los pocos sistemas de gestión de contenidos que cuenta con la suficiente participación activa como para generar soluciones rápidas y precisas ante la presencia de bugs (vulnerabilidades) que vayan



surgiendo. (9)

Requisitos del sistema:

Servidor de aplicación: Apache (recomendado), servidor con soporte para PHP.

Base de datos: MySQL.

Licencia: GNU GPL (Licencia Pública General de GNU o General Public License en inglés).

Lenguaje de programación: PHP (pre-procesador de hipertexto).

Servidor web: Apache.

Sistema operativo: Multiplataforma.

1.5.3. Typo3

Es un CMS, de fuente abierta y desarrollado bajo licencia GPL. Además, de máxima flexibilidad y extensibilidad ofrece gran número de módulos disponibles lo que hacen de Typo3 una herramienta de gran importancia. Se trata de un sistema modular, muy flexible, muy potente y que posee una curva de aprendizaje muy suave para usuarios redactores, esto lo capacita como una plataforma de desarrollo de cualquier tipo de solución web. La lista de prestaciones es muy extensa y está continuamente en crecimiento debido a la gran comunidad de desarrolladores que contribuyen al producto. (7)

Requisitos del sistema:

Base de datos: MySQL (Es posible usar cualquier base datos accediendo mediante PHP).

Licencia: GNU GPL.

Lenguaje de programación: PHP.

Servidor web: Apache.

Sistema operativo: Windows o Mac.

1.5.4. Drupal

Drupal es un CMS a código abierto creado por Dries Buytaert. Este sistema permite administrar recursos web, publicar contenidos, incluye foros y blog. La información es almacenada en una base de datos. Es un sistema multiusuario, multiplataforma, con soporte para varios lenguajes, extensible y modular. Permite editar el diseño del sitio, administrar los bloques de contenidos, menús y activar/desactivar los módulos agregados en el sitio. Además, posibilita administrar los log del CMS, visualizar eventos de los usuarios que acceden al sitio, los errores por accesos denegados, páginas no encontradas, generar reportes del estado en las operaciones del sitio web y los problemas detectados. En la configuración del sitio se puede administrar los temas, configurar reportes de errores, sistemas de archivos, localización, funcionamiento, información del sitio y mantenimiento del sitio. (11)

Requisitos del sistema:

Servidor de aplicación: PHP 4.3.5 o superior.

Base de datos: MySQL, PostgreSQL.

Licencia: GNU GPL.

Lenguaje de programación: PHP.

Servidor web: Apache, IIS (Internet Information Services o Servicios de Información de Internet).

Sistema operativo: multiplataforma.

1.5.3. Situación del CMS Drupal

Después de haber hecho un análisis sobre algunas características, requerimientos y ventajas de los CMS anteriormente descritos se determinó aceptar la sugerencia del cliente y utilizar como sistema gestor de contenidos al CMS Drupal para el desarrollo de la aplicación, teniendo como base los resultados, las características y ventajas que ofrece el mismo, situándolo entre los más eficientes y usados por desarrolladores en el mundo como también en la universidad.

En los subepígrafes que se describen a continuación se muestra un estudio más profundo sobre dicho sistema.

1.5.3.1. A nivel internacional

Debido a las facilidades que brinda Drupal es utilizado en diversos lugares del mundo. No ha dejado de perfeccionarse en sus más de 7 años de existencia. Tiene más de trescientos (300) módulos hechos por terceras partes, más de cincuenta y cinco mil (55 000) sitios web que funcionan soportados en este CMS. Una comunidad numerosa y de empuje colaborativo, lo cual contribuye a que los errores sean corregidos rápidamente. (14)

1.5.3.2. A nivel nacional

En Cuba se capacitan a futuros ingenieros, profesionales y técnicos para aprender a trabajar con el CMS Drupal y actualmente existen varios sitios web desarrollados en este CMS los cuales tienen gran importancia debido a que a través de estos se promueve el intercambio mediante foros, debates, comentarios, artículos. Por ejemplo, el Instituto de Información Científica y Tecnológica (IDICT) utiliza el CMS Drupal para el servicio de publicación web, el cual tiene como objetivo que los integrantes de este grupo tengan un lugar donde publicar sus trabajos relacionados con el proyecto que ellos desarrollan; así también el portal oficial de la cultura matancera (atenas.cult.cu), el Centro de Estudios Ché Guevara, el sitio oficial de la Cámara de Comercio de la República de Cuba, así como el Centro de Estudios Martianos que tienen como portal oficial un sistema desarrollado en Drupal. (14)

1.5.3.3. En la Universidad de las Ciencias Informáticas

Por la seguridad y estabilidad que brinda se utilizó el sistema de gestión de contenidos Drupal para migrar la intranet actual y el portal Octavitos perteneciente a la facultad 4. Existe una comunidad con nombre Drupaleros con conocimiento acerca del CMS en la universidad cuyo sitio oficial también está desarrollado en dicha herramienta y que ha trabajado con buenos resultados. Además la universidad lo ha empleado para el desarrollo de proyectos importantes, tanto nacionales como para los de colaboración internacional, como es el ejemplo del portal Albet. También se han creado con el uso de este CMS las comunidades de C++ y PHP.

1.5.4. Características Generales de Drupal

Drupal contiene una serie de generalidades como CMS al fin y proporciona aspectos específicos que lo hacen interesante a la hora de considerar su uso:

- ✓ Ayuda on-line: un robusto sistema de ayuda online y páginas de ayuda para los módulos del Núcleo, tanto para usuarios como para administradores.
- ✓ Búsqueda: todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- ✓ Código abierto: el código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- ✓ Módulos: la comunidad de Drupal ha proporcionado muchos módulos que brindan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, entre otros.
- ✓ Personalización: un robusto entorno de personalización está implementado en el Núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- ✓ URLs amigables: Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda. (15)

1.5.4.1. Ventajas

Permite la creación de sub-apartados en los foros. Contiene métodos avanzados de clasificación de artículos. El acceso a la web es rápido. Permite a los usuarios configurar su propio perfil en dependencia de los permisos otorgados por el administrador.

1.5.4.2. Desventajas

Posee una alta curva de aprendizaje. Contiene gran variedad de módulos, tantos que resulta difícil seleccionar cuál es el más eficiente para determinada funcionalidad. (16)

1.5.4.3. Módulos incorporados

Módulo tema o *theme*

Para la confección de este módulo fue necesario un arduo trabajo de diseño. Uno de los requisitos no funcionales del cliente explicaba exactamente las características esenciales y las particularidades con las que debía contar la aplicación en cuestión. La realización del tema estaría guiada por la idea de colores similares a los del sitio oficial del acuario, puesto que el plan de hosting incluía el anclaje del sistema de gestión al mismo. El cliente además tenía un interés especial en que el tema no contara con un diseño complejo debido a que los conocimientos de informática de los especialistas eran bastante limitados. De esta forma se definió un montaje sencillo, en el cual se ubicó un menú con las principales opcionalidades del sitio en la parte superior, junto al banner. El cuerpo o body, donde figuran los contenidos y por último un footer con accesos directos hacia algunos sitios importantes, datos estadísticos de interés, la información de contacto y copyright. Sin embargo, la sencillez visual no impidió que se dejaran de aplicar los principios fundamentales de la Web 2.0, además de lograr un resultado fresco, novedoso y mejorado, con una interfaz en la que su buen acabado respalda cada servicio y funcionalidad brindada.

Módulo estadísticas

Este módulo responde a la necesidad del cliente de conocer de forma factible algunas estadísticas detalladas del sitio con solo echar un rápido vistazo. A pesar de ser un módulo independiente el principio de funcionamiento del mismo se basa en la retroalimentación del módulo statistics de Drupal, el cual brinda un informe detallado de las visitas efectuadas, ya sea por páginas o por contenidos específicos, número de comentarios emitidos por los usuarios, tiempo total de conexión de un usuario a la aplicación, la dirección IP desde donde se conectó entre otras opciones. Sin embargo, el acceso y comprensión de estos resultados para un usuario promedio puede ser algo tedioso y difícil de entender a fondo. La forma en la que Drupal muestra estos detalles requieren de los conocimientos de un usuario avanzado, además, el cúmulo de información es muy extenso, hasta el punto de mostrar detalles y resultados que solo serían de interés para desarrolladores. El módulo estadísticas solo utiliza la información global de mayor importancia y la muestra a través de una sencilla interfaz, ubicada en el footer del sitio. Aparentemente el tomar los datos de uno de los módulos nativos de Drupal y mostrarlos puede parecer una tarea simple, no obstante algunos datos requieren de una programación más compleja. Ejemplo de ello son las visitas diarias, Drupal las llama hits y

las documenta por minutos, no en días, ni semanas y mucho menos en años. Por esta razón se precisa del uso de métodos auxiliares que permitan, en casos como estos, filtrar la información obtenida hasta alcanzar finalmente el resultado deseado.

Además, de las descritas se abordaron otras importantes características de Drupal que se pueden encontrar en el [Anexo 10](#).

1.6. SISTEMA DE GESTIÓN DE BASE DE DATOS

A continuación se hará énfasis en los gestores de base de datos, con los cuales Drupal tiene más relación.

1.6.1. PostgreSQL

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzados del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle (que es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Se considera como uno de los sistemas de bases de datos más completos destacando su soporte de transacciones, estabilidad, escalabilidad, además es multiplataforma). Esta herramienta aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, optimización de consultas y herencia. Tiene soporte para lenguajes procedurales internos (son aquellos en los cuales el usuario instruye al sistema para que lleve a cabo una serie de operaciones en la base de datos con el fin de calcular el resultado deseado y están fundamentados en la utilización de variables para almacenar valores y en la realización de operaciones con los datos almacenados), incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra de sus ventajas es su habilidad para usar Perl o Python. (18)

1.6.2. MySQL

MySQL es un pequeño y compacto servidor de Base de Datos, ideal para pequeñas y medianas aplicaciones. Soporta el estándar SQL (ANSI) y además está disponible para distintas plataformas. En la actualidad es el gestor de bases de datos de código fuente abierto más usado. Debido a sus características funcionales ha dado lugar a un sistema de administración de las bases de datos incomparable en velocidad, compactación, estabilidad y facilidad de despliegue. Además, provee altas características de seguridad, contiene un sistema de autenticación capaz de aceptar o denegar las conexiones, también una herramienta para administración de los privilegios y métodos de encriptación.

Es además muy utilizado en aplicaciones web como Drupal, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python). Su utilización en aplicaciones web está muy ligada a PHP, con el cual a menudo aparece combinado. Un inconveniente que pudiera señalarse de este gestor de base de datos es que es muy rápido en la lectura, lo que puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. (19)

1.6.3. Selección del sistema de gestión de base de datos

Después de haber concluido con el análisis de los sistemas de gestión de base de datos soportados por el CMS Drupal, se decidió utilizar MySQL 5.0, debido a que sus principales ventajas radian en su rapidez, robustez, es multiproceso y multiplataforma. Además, porque presenta un considerable bajo consumo de recurso frente a su homólogo. [Ver anexo 5.](#)

1.7. SERVIDORES WEB

El servidor web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de petición, el servidor web buscará una página web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML (Lenguaje de Marcas de Hipertexto o HyperText Markup Language en Inglés) al cliente o navegador que realizó la petición. El servidor web es fundamental en el desarrollo de las aplicaciones del lado del servidor, ya que estas se ejecutan en él. (20)

1.7.1. Internet Information Services

Internet Information Services (IIS) es un servidor web con un conjunto de servicios para sistemas operativos Windows. Originalmente era parte del Option Pack para Windows NT. Luego, fue integrado en otros sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS, actualmente forma parte de la distribución estándar de Windows. Los servicios que ofrece son: FTP, SMTP, NNTP y HTTP/HTTPS.

Permite que las computadoras en las que esté instalado se puedan publicar páginas web tanto local como remoto y a través de varios módulos facilita el procesamiento de distintos tipos de páginas como las Active Server Pages (ASP) y ASP.NET de Microsoft y de otros fabricantes como PHP o Perl. (42)

1.7.2. Apache

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1). Entre sus características destacan: multiplataforma, modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona y con la API de programación de módulos, para el desarrollo de módulos específicos. Soporte para manejo de archivos grandes (más grandes de 2 GB) en plataformas de 32 bits. Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor. Se desarrolla de forma abierta. (20)

1.7.3. Apache Tomcat

Tomcat funciona como contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation e implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. Aunque es un servidor web que puede funcionar de forma autónoma en entornos con alto nivel de tráfico y disponibilidad con soporte para las JSP no es un servidor de aplicación por lo cual en ocasiones el motor de servlets de Tomcat se presenta en combinación con el servidor web Apache.

Dado que Tomcat fue escrito en lenguaje de programación Java es capaz de funcionar en cualquier sistema operativo siempre y cuando este disponga de la máquina virtual Java.

1.7.4. Selección del servidor web

Después de haber concluido con el análisis de los servidores web, se decidió utilizar Apache debido a que Internet Information Services solo funciona bajo servidores Microsoft, debe ser usado bajo licencia, mientras Apache es un proyecto libre al igual que Tomcat, pero se descartó el uso de este último por las dependencias que presenta de otras aplicaciones. Por otro lado Apache se consolidó a principios de 2011 como el servidor web más usado a nivel mundial, según datos mostrados por Netcraft basados en el servicio de monitoreo Royal Pingdom el número de sitios web que utilizan Apache alcanza los 152 millones de sitios. (44)

1.8. LENGUAJES DE PROGRAMACIÓN EMPLEADOS

En esta sección se especifican los diferentes lenguajes de programación empleados en la realización del producto.

1.8.1. PHP

Se utiliza PHP pues es un lenguaje de programación usado generalmente para la creación de contenidos dinámicos para sitios web o para la creación de aplicaciones para servidores. PHP

también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos. Este lenguaje es multiplataforma, tiene la capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL, el mismo lee y manipula datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos y permite las técnicas de Programación Orientada a Objetos (POO) (permite agrupar secciones de código con funcionalidades comunes. Con la programación orientada a objetos se pretende agrupar el código encapsulándolo y haciéndolo independiente, de manera que una modificación debida al crecimiento de la aplicación solo afecte a unas pocas líneas. El objetivo de POO es catalogar y diferenciar el código, en base a estructuras jerárquicas dependientes, al estilo de un árbol genealógico). Además, el código PHP está incrustado dentro del HTML e interactúa con el mismo, lo que permite diseñar la página web en un editor común de HTML y añadir el código dinámico dentro de las etiquetas `<?php?>`. (21)

1.8.2. HTML

Lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet, HTML se ha convertido en popular para la construcción de documentos y también de los más fáciles de aprender. Existen además otros programas para la realización de sitios web o edición de código HTML, como el software Macromedia Dreamweaver, muy usado en el ámbito de diseño y programación web.

A estos programas se les conoce como editores WYSIWYG. Ahora bien, esto no significa una manera distinta de realizar sitios web, sino una forma un tanto más simple, pues estos programas, además de tener la opción de trabajar con la vista preliminar, tienen su propia sección HTML, la cual va generando todo el código a medida que se va trabajando. Este lenguaje utiliza estilos CSS. (21)

1.8.3. CSS

Las hojas de estilo en cascada o “Cascading Style Sheets” (CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. Al utilizar CSS se tiene control centralizado de la presentación de un sitio web completo con lo que se agiliza de forma considerable la actualización del mismo. Estos estilos aumentan considerablemente la accesibilidad, por ejemplo, personas con poca visibilidad pueden configurar su propia hoja de estilo para aumentar el tamaño del texto. Con estilos en las páginas el documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño. (21)

1.8.4. XML

Drupal es capaz de generar código XML, si el usuario lo desea, el XML o “eXtensible Markup Language”, que se puede traducir como lenguaje de marcas extensible, no es un lenguaje particular pues permite crear sus propias etiquetas. Con él se desea el intercambio de información estructurada entre distintas plataformas y permite la compatibilidad entre los diferentes sistemas, así como compartir información de una forma segura y fiable, se puede hacer uso de este en base de datos, editores de texto, hojas de cálculo, entre otras cosas. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos, como es el caso de HTML. Un documento XML está formado por: Prólogo, describe la versión XML, el tipo de documento y otras cosas; Cuerpo, no es opcional en un documento XML, el cuerpo debe contener al menos un elemento raíz; Elementos, los elementos XML pueden tener contenido (más elementos, caracteres o ambos), o bien ser elementos vacíos; Atributos, son una manera de incorporar características o propiedades a los elementos de un documento y Entidades predefinidas, para representar caracteres especiales para que no sean interpretados como marcado en el procesador XML. (22)

1.8.5. JavaScript

JavaScript es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Al igual que Java, JavaScript es un lenguaje orientado a objetos propiamente dicho, ya que dispone de Herencia, si bien esta se realiza siguiendo el paradigma de programación basada en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento o Document Object Model en inglés). En junio de 1997 fue adoptado como un estándar ECMA, con el nombre de ECMAScript. Poco después también lo fue como un estándar ISO. JScript es la implementación de ECMAScript de Microsoft, muy similar al JavaScript de Netscape, pero con ciertas diferencias en el modelo de objetos del navegador que hacen a ambas versiones con frecuencia incompatibles. Para evitar estas incompatibilidades, el World Wide Web Consortium diseñó el estándar DOM, que incorporan Konqueror a partir de los navegadores: versiones 6 de Internet Explorer y Netscape Navigator, Opera versión 7 y Mozilla desde su primera versión. (16)

1.9. METODOLOGÍAS DE DESARROLLO

Se entiende por metodología de desarrollo una colección de documentación formal referente a

los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia logrando así con ello el cumplimiento de los requisitos iniciales y minimizar las pérdidas de tiempo en el proceso de generación de un software. (23)

A nivel internacional las instituciones y empresas dedicadas a la industria del software emplean en su actividad de desarrollo, modelos, metodologías o procedimientos estándares para desarrollar, instalar y mantener un producto de este tipo. El objetivo de un proceso de desarrollo es aumentar la calidad del software en todas las fases por las que atraviesa con mayor transparencia y control sobre el proceso, garantizando además que esas medidas sean reproducibles en cada fase durante el desarrollo del software. Independientemente del alcance del software, hay que producirlo en el tiempo y con el costo esperado.

Para lograr un correcto desarrollo de software, existen metodologías que establecen un conjunto de actividades que definen quién debe hacer qué actividad, cómo hacerla, y cuándo llevarla a cabo.

Implantar un proceso de desarrollo es una tarea más a medio-largo plazo que una tarea de resultados inmediatos ya que consume tiempo que las personas involucradas se adapten al proceso (nuevos estilos de trabajo y nuevos puntos de vista), pero una vez superada la transformación se recupera con creces. Casi siempre, no tiene sentido ajustarse a un proceso al pie de la letra sino que hay que adaptarlo a las necesidades y características de cada empresa, equipo de trabajo o incluso casi a cada proyecto.

En los últimos tiempos la cantidad y variedad de los procesos de desarrollo ha aumentado de forma impresionante; se podría decir que en estos últimos años se han desarrollado dos corrientes en lo referente a los procesos de desarrollo, las llamadas metodologías pesadas o tradicionales y las ligeras o ágiles. La diferencia fundamental entre ambos es que las pesadas intentan lograr sus objetivos centrándose en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usará, mientras que las ágiles siguen la filosofía de darle mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, manteniendo una alta calidad.

Según la bibliografía consultada las metodologías pesadas son las más tradicionales. Estas metodologías están centradas en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar y requieren una extensa documentación, ya que pretenden prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar, respecto a tiempo y recursos que son necesarios emplear, donde se requiere de una gran organización. (38)

En el caso que ocupa a la presente investigación fue indispensable aplicar un proceso de

ingeniería inversa, debido a la urgente necesidad del cliente de obtener una primera versión funcional del producto cuanto antes (en el plazo de un mes aproximadamente). Esta fue la pieza clave que permitió, en primera instancia, escoger una metodología ágil y descartar, por consiguiente, las tradicionales o pesadas como consecuencia del ajustado calendario de entregas y el inminente trabajo contrarreloj.

Figura 1. Comparación entre metodologías ágiles y tradicionales. (43)

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
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

Las metodologías ágiles tienen marcadas diferencias como forma de desarrollar un software. Están basadas en heurísticas provenientes de prácticas de producción de código, están especialmente preparadas para cambios durante el proyecto y basan su equipo en no más de 10 programadores trabajando en el mismo sitio, utiliza menos roles y hace menos énfasis en la arquitectura del software. Además de incluir al cliente como un miembro más del equipo, abogando siempre por la comunicación interpersonal y la flexibilidad.

El sistema que se desarrolló en el presente trabajo tiene características que imponen utilizar metodologías ágiles, pues la resistencia al cambio, los procesos en extremo controlados, las reuniones protocolares y la cantidad excesiva de roles, entre otras cualidades que poseen las metodologías tradicionales, harían el proceso más lento y más complicado.

Principios del manifiesto ágil.

- La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.

- Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.
- Entregar frecuentemente versiones del software que funcionen desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.
- El personal del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.
- Construir el proyecto entorno a individuos motivados. Brindar al equipo el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo con éxito.
- El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro del equipo de desarrollo.
- El software que funciona es la medida principal de progreso.
- La atención continua a la calidad técnica y al buen diseño mejora la agilidad.
- La simplicidad es esencial.
- Las mejores arquitecturas, requisitos y diseños surgen con frecuencia de los equipos organizados por sí mismos.
- En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento. (45)

En los subepígrafes que se describen a continuación se analizan tres de las metodologías que pertenecen al antes mencionado grupo de las ágiles.

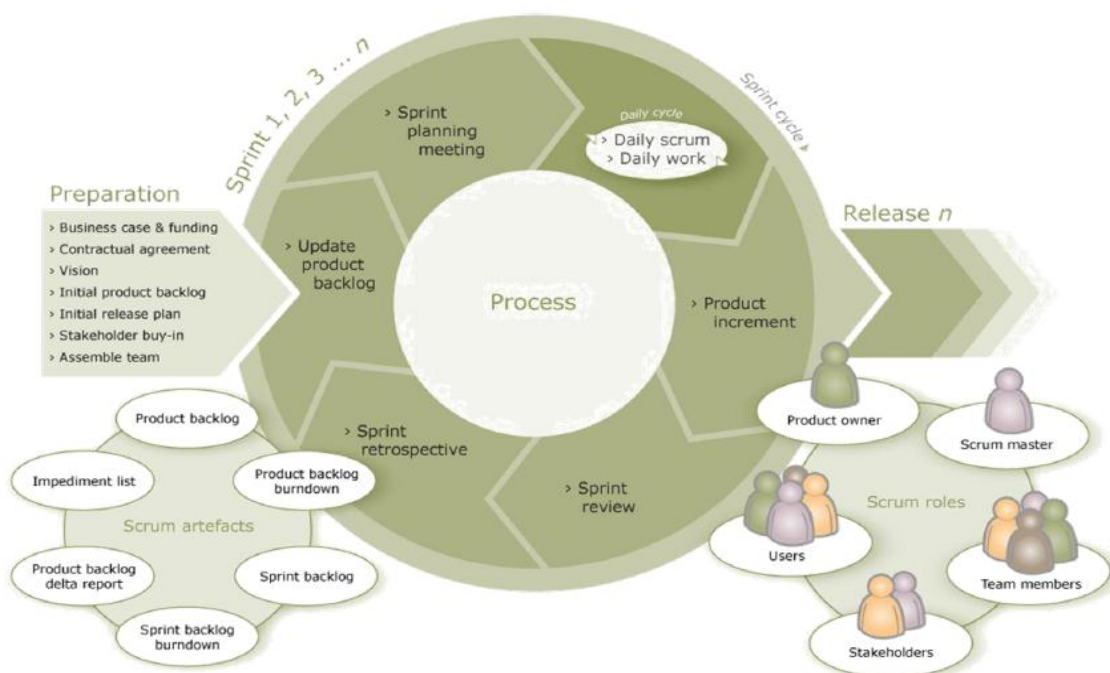
1.9.1. Scrum

Scrum es una metodología ágil que define un proceso empírico, iterativo e incremental de desarrollo que intenta obtener ventajas respecto a los procesos definidos (cascada, espiral y prototipos) mediante la aceptación de la naturaleza caótica del desarrollo de software, y la utilización de prácticas tendientes a manejar el riesgo a niveles aceptables. El mismo surge en 1986, de un artículo de la Harvard Business Review titulado “The New Product Development Game” de Hirotaka Takeuchi e Ikujiro Nonaka, que introducía las mejores prácticas más utilizadas en 10 compañías japonesas altamente innovadoras. A partir de ahí y tomando referencias al juego de rugby, Ken Schwaber y Jeff Sutherland formalizan el proceso conocido como Scrum en el año 1995. Scrum es un método iterativo e incremental que enfatiza prácticas y valores de project management por sobre las demás disciplinas del desarrollo. Al principio del proyecto se define el Product Backlog, que contiene todos los requerimientos funcionales y no funcionales que deberá satisfacer el sistema a construir. Los mismos estarán especificados de

acuerdo a las convenciones de la organización ya sea mediante: features, casos de uso, diagramas de flujo de datos, incidentes, tareas, etc. El Product Backlog será definido durante reuniones de planeamiento con los stakeholders. A partir de ahí se definirán las iteraciones, conocidas como Sprint en la jerga de Scrum, en las que se irá evolucionando la aplicación evolutivamente. Cada Sprint tendrá su propio Sprint Backlog que será un subconjunto del Product Backlog con los requerimientos a ser construidos en el Sprint correspondiente. La duración recomendada del Sprint es de un mes.

Aunque surgió como modelo para el desarrollo de productos tecnológicos, también se emplea en entornos que trabajan con requisitos inestables y que requieren rapidez y flexibilidad; situaciones frecuentes en el desarrollo de determinados sistemas de software. (46)

Figura 2. Ciclo de desarrollo de SCRUM



1.9.2. Proceso Unificado Ágil (Agile Unified Process, AUP)

AUP es una versión simplificada del Proceso Unificado Racional (RUP). Describe una simple y fácil manera de comprender un enfoque del desarrollo de software para aplicaciones empresariales utilizando las técnicas y conceptos de agilidad y manteniéndose fiel al RUP.

Al igual que el RUP, AUP define cuatro fases dentro de su marco de trabajo:

- 1- Inicio.
- 2- Elaboración.
- 3- Construcción.
- 4- Transición.

Cada una de ellas está dividida en una serie de iteraciones que ofrecen como resultado un

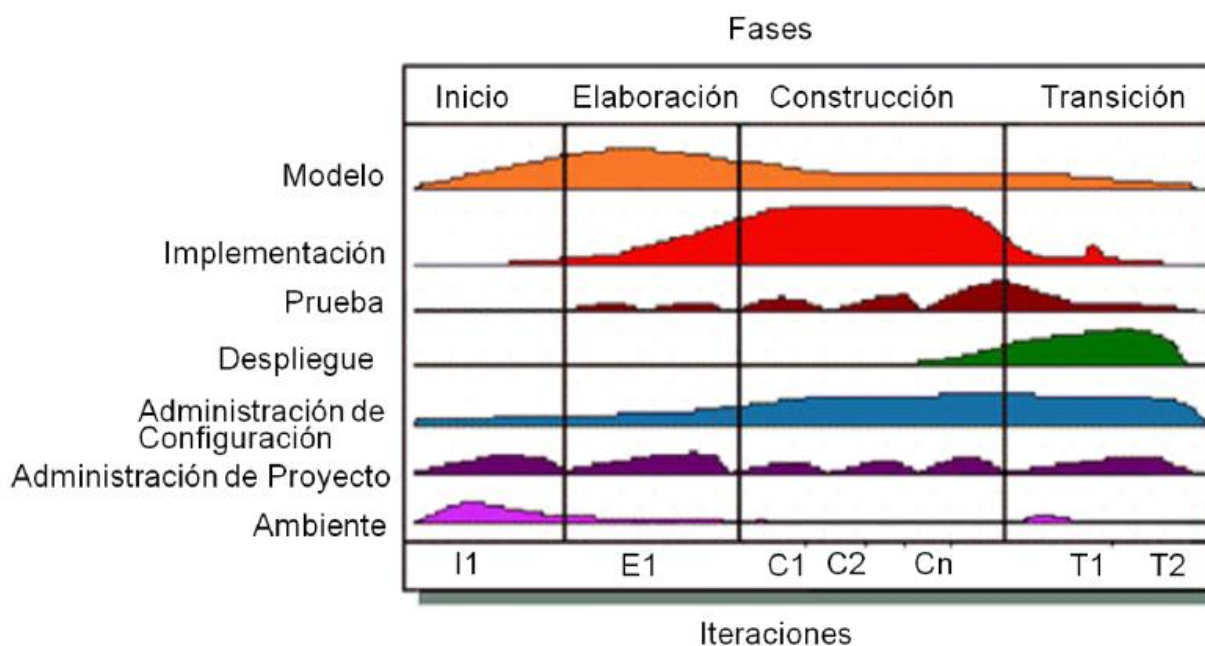
incremento del producto desarrollado, que añade o mejora las funcionalidades del sistema en desarrollo. Es decir, un “incremento” no implica necesariamente una ampliación de dicho sistema.

Durante cada una de estas iteraciones se realizarán a su vez las actividades definidas en el ciclo de vida clásico: requisitos, análisis, diseño, implementación, prueba e implantación. Aunque todas las iteraciones suelen incluir trabajo en casi todas estas actividades, el grado de esfuerzo dentro de cada una de ellas varía a lo largo del proyecto. AUP establece siete disciplinas en total, lo más notable es que aparece una disciplina de Modelado la cual abarca las disciplinas de Modelado del Negocio, Requerimientos y Análisis y Diseño de RUP. También las disciplinas de la Administración de la Configuración y Cambios ahora son la disciplina de la Administración de la Configuración.

Los equipos AUP típicamente liberan versiones desarrolladas al final de cada iteración, contrarios al enfoque "big bang" donde se entrega todo el software en una sola liberación. Esto hace que la vida de un sistema se constituya de una serie de ciclos donde al final de cada uno de ellos se obtiene una versión final del producto, que no sólo satisface ciertos casos de uso, sino que está lista para ser entregada y puesta en producción.

AUP es una metodología entre XP y lo tradicional de RUP, un proceso que es ágil pero incluye explícitamente actividades y artefactos de trabajo que se está acostumbrado a hacer en RUP (39).

Figura 3. Ciclo de desarrollo de AUP



1.9.3. Programación Extrema (XP)

La Programación Extrema o “eXtreme Programming”, es otra de las metodologías de desarrollo de software que existen en la actualidad. Intenta como metodología ágil reducir la complejidad del software por medio de un trabajo orientado directamente al objetivo, basado en las relaciones interpersonales y la velocidad de reacción. XP se basa en realimentación 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. Se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

- 1- El cliente define el valor de negocio a implementar.
- 2- El programador estima el esfuerzo necesario para su implementación.
- 3- El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
- 4- El programador construye ese valor de negocio.
- 5- Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración.

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega (Release), Iteraciones, Producción, Mantenimiento y Muerte del Proyecto.

1. Exploración: los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, a partir de lo cual se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.

2. Planificación de la Entrega: el cliente establece la prioridad de cada historia de usuario y a su vez, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

3. Iteraciones: incluye varias iteraciones sobre el sistema antes de ser entregado. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para

entrar en producción.

4. Producción: esta fase requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase.

5. Mantenimiento: mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente.

6. Muerte del Proyecto: ocurre cuando el cliente no tiene más historias para ser incluidas en el sistema, siendo satisfechas todas sus necesidades. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo.

Lo fundamental en este tipo de metodología es:

- ✓ La comunicación entre los usuarios y los desarrolladores.
- ✓ La simplicidad al desarrollar y codificar los módulos del sistema.
- ✓ La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales. (43)

Figura 4. Ciclo de desarrollo de XP



1.9.4. Selección de la metodología a utilizar

No existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada al contexto del proyecto (recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, entre otros).

Históricamente, las metodologías tradicionales han intentado abordar la mayor cantidad de situaciones de contexto del proyecto, exigiendo un esfuerzo considerable para ser adaptadas, sobre todo en proyectos pequeños, con requisitos muy cambiantes y en donde desarrollar un software que funcione es más importante que conseguir una buena documentación. Esto hace que utilizar metodologías con las características de RUP y AUP no sea lo más factible, además el desarrollo está a cargo de dos personas y se tendrían que asignar varios roles a desempeñar por cada uno, y se haría complejo cumplir con todas las actividades de cada uno de ellos. Al mismo tiempo, los requerimientos son cambiantes a lo largo de todo el ciclo de vida de la aplicación y provoca que el modelo de fases de estas metodologías se quiebre porque cada nuevo requisito necesitaría reiniciar una iteración para comenzar el ciclo de actividades que den cumplimiento a su funcionalidad.

Por otra parte, SCRUM está enfocada hacia la gestión de proyectos y a la idea de ponerse a trabajar prácticamente desde el primer momento y empezar a sacar frutos de ese trabajo para que el cliente vaya viendo los avances. Para ello, las acciones fundamentales de SCRUM son el Product Backlog, Sprint Backlog y Daily Scrum Meeting, además de una serie de reuniones y documentos en las cuales se planifican y se archivan los resultados de los Sprint, respectivamente. Debido a que no se cuenta con una planificación concreta ya las actividades se llevan a corto plazo, y que los desarrolladores están en constante comunicación y no necesitan de reuniones para ponerse de acuerdo, se consideró que utilizar a SCRUM como método desarrollo sería violar una serie de acciones que son claves en esta metodología.

La Programación Extrema centra su atención en la producción de software con medianos o pequeños equipos de desarrollo, asumiendo que la planificación nunca será perfecta, y que los requerimientos cambian a lo largo de todo el ciclo de vida de la aplicación según varíen las necesidades del negocio; por tanto, el valor real reside en obtener rápidamente un plan inicial, y contar con mecanismos de feedback que permitan conocer con precisión dónde estamos. En otras metodologías contar con la presencia del cliente es una ventaja, pero en XP es una precondition inicial; esta colaboración entre ambos siempre está disponible para resolver dudas y para decidir qué se hace y qué no se hace en cada momento, en función de los intereses del negocio. Una de las cualidades más destacables de XP es su sencillez, tanto en su aprendizaje como en su aplicación, reduciéndose así los costos de implantación en un equipo de desarrollo. XP promueve un entorno físico con un ambiente que permite la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo y

hace énfasis en que el código es la documentación, por tanto la generación de documentación para mantener el proceso en un estado comprensible no es forzado de ninguna forma. Por las características mencionadas, además de contar con abundante cantidad de información disponible y ser con diferencia la metodología ágil más popular se decidió seleccionar la metodología XP para que guíe la marcha del proyecto y asegure su éxito.

1.10. OTRAS HERRAMIENTAS PARA EL DESARROLLO DEL PRODUCTO

1.10.1. GIMP

GIMP (GNU Image Manipulation Program) es un programa de edición de imágenes digitales en forma de mapa de bits. Libre, gratuito, multiplataforma y es parte del proyecto GNU por lo que está disponible bajo la licencia pública general de GNU.

GIMP sirve para procesar imágenes digitales. Los usos típicos incluyen la creación de gráficos y logos, el cambio de tamaño, recorte y modificación de fotografías digitales, la modificación de colores, la combinación de cuadros usando un paradigma de capas, la eliminación o alteración de elementos no deseados o la conversión entre distintos formatos de imágenes. También se puede utilizar en la creación de escenas animadas sencillas, la manipulación de vectores y la edición avanzada de vídeos. (47)

Por sus múltiples aplicaciones y su ejecución multiplataforma desde entornos libres se utilizó su versión estable 2.6 en el diseño de prototipos en la investigación y en la edición de imágenes para el desarrollo de la interfaz visual del sistema.

1.10.2. Notepad++

Es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación para Microsoft Windows. Gracias a su velocidad, puede convertirse en una alternativa al bloc de notas. Con la implementación de navegación por pestañas, moverse entre los archivos de texto abiertos es más cómodo. Por defecto incluye la extensión TextFX que proporciona muchas opciones de transformación de texto. (25)

Por las características que presenta, el soporte de varios lenguajes como: JavaScript, XML, CSS y PHP, por su velocidad y el bajo consumo de recursos en su empleo, se utilizó esta herramienta como entorno de desarrollo integrado (IDE).

1.11. CONCLUSIONES DEL CAPÍTULO

En el transcurso de este capítulo, se realizó un análisis detallado de los diferentes conceptos zoológicos y biológicos empleados, del estado del arte de la Gestión de la Información, evidenciando a través de la misma el surgimiento de los Sistemas de Gestión de Contenidos, sus características y ventajas de aplicar las herramientas existentes para su gestión y



administración en el desarrollo eficiente de la aplicación. Llegando así a un aserto parcial de la investigación. Se determinó que el CMS Drupal, por las ventajas que posee como sistema gestor de contenidos sería la herramienta más idónea para la realización del software, en su versión 7.8. Además se realizó un resumen de las herramientas y tecnologías a utilizar, a través del cual se seleccionó como metodología de desarrollo a XP, como lenguajes de programación del lado del cliente JavaScript y CCS, del lado del servidor PHP, como gestor de base de datos MySQL en su versión libre 5.0 , como servidor web Apache 2.2.21, como herramienta de diseño de interfaces y prototipos GIMP 2.6 y como entorno de desarrollo integrado Notepad++ 5.8.5.

CAPÍTULO 2: EXPLORACIÓN Y PLANIFICACIÓN

2.1. INTRODUCCIÓN

Para desarrollar la solución propuesta utilizando la metodología de desarrollo XP, se tuvieron en cuenta los pasos definidos para el proceso de desarrollo que propone esta metodología. En el presente capítulo se realizará una breve descripción de las fases definidas por XP, haciendo énfasis en la fase de planificación, donde se realizarán descripciones de algunos artefactos generados por la misma, de los cuales se registran las historias de usuarios y el plan de iteración.

2.2 FLUJO ACTUAL DE ACTIVIDADES EN EL ACUARIO NACIONAL CUBA

Actualmente la gestión de la información de las colecciones marinas del Acuario Nacional Cuba es muy tediosa y engorrosa ya que se realiza de forma manual a través de un grupo de bases de datos sin conexiones entre sí, creadas con la herramienta de origen propietario Microsoft Office Access y además a través del uso de archivos en papel. Por otra parte no existen mecanismos que posibiliten de manera eficaz el acceso a la información de las colecciones, lo que dificulta e imposibilita en ocasiones la realización de consultas a las mismas por especialistas u otros usuarios que necesiten alguna información. De igual manera no existen medios para mantener al tanto de la situación de la colecciones al público, ni poder recoger la opinión de los mismos en algún espacio donde se pueda debatir sobre el trabajo que se realizan con las distintas colecciones o dejar satisfechas las dudas que puedan existir relacionadas con estas.

El desarrollo de las actividades ocurre de la siguiente manera:

Si algún interesado o aficionado desea conocer sobre la historia de las colecciones marinas en Cuba o hacer uso de la información disponible sobre las mismas para algún tipo de estudio o investigación, debe dirigirse al departamento de colecciones marinas del Acuario Nacional Cuba y en el mismo solicitar a un especialista que le auxilie en la investigación, a lo cual corresponde en muchas ocasiones la exhaustiva búsqueda en varias bases de datos donde está contenida la información de cada ejemplar que posee la colección; otra solución sería realizar la búsqueda a través de los archivos en papel donde se encuentran catalogados cada ejemplar y su respectiva información, tarea que sería igual de engorrosa que el uso de las base de datos.

El mecanismo es similar si se decide investigar sobre las características de un grupo específico de ejemplares, ya que no se cuenta con ningún medio que permita realizar una búsqueda por categorías, la solución más próxima sería la consulta de varios ejemplares y comparar las especificaciones que presenten entre sí para arribar así a conclusiones previas, lo cual no garantiza que los resultados sean del todo precisos si el estudio no fuese a toda la

población existente en la colección a la que pertenece el grupo en cuestión. Vale la pena recordar que se trata de más de 60 000 ejemplares.

Existen algunos medios para divulgar la labor que se realiza con dichas colecciones como son a través de conferencias impartidas por los especialistas y exposiciones disponibles a quienes visitan las instalaciones del acuario, donde radica la mayor parte de las colecciones, pero esto no garantiza que la información esté disponible en momento y forma a todo el que quiera hacer uso de la misma.

El propio acuario como demás instituciones homólogas en el país carecen de un espacio dedicado a este tipo de consulta lo que muchas veces propicia el desconocimiento de la existencia de este tipo de colecciones y la útil información que propician sus ejemplares.

2.3. PROPUESTA DE SOLUCIÓN

El desarrollo del presente trabajo se proyecta a la implementación de un sistema que permita manejar toda la información referente a las colecciones marinas del Acuario Nacional Cuba. El cliente fue bastante explícito al exponer la palpable necesidad de una aplicación lo más minimalista posible en cuanto a detalles estéticos y visuales, en la que la sencillez y la facilidad de su uso fuesen la premisa fundamental que propiciarán un ambiente fluído con la finalidad de optimizar el trabajo.

La parte más compleja de esta propuesta se centró en la estandarización de las bases de datos de las colecciones, las cuales habían sido desarrolladas bajo el modelo propietario de Microsoft Access e inicialmente no se relacionaban entre sí. Primeramente se tuvo que organizar toda la extensa gama de contenidos y catalogarlos. Entiéndase un conjunto de 112 bases de datos no unificadas y agrupadas en tres clasificaciones generales según su procedencia (las provenientes del Instituto de Oceanología, el Acuario Nacional Cuba y las especies Tipo de ambas instituciones, respectivamente) y en otras 77 según su jerarquía taxonómica, para un promedio de 530 entradas por archivo, sumando un total de alrededor de 60 000 entradas con la información de cada ejemplar depositado en colección. Entre la información que se quiere manejar se encuentran imágenes y una ficha que contiene aproximadamente 42 campos distintos con la información relativa a cada ejemplar. La introducción de nomencladores y la creación de un patrón común para cada campo, además de la correcta disposición de algunos datos no estandarizados, constituyeron la base para enfrentarse a tan difícil reto.

Los requisitos fundamentales planteados inicialmente por el cliente fueron los correspondientes a gestionar un ejemplar, las restantes funciones (noticias, mensajería interna, comentarios, registro y estadísticas) de la aplicación fueron añadidas a partir de varios encuentros; en los

que se discutieron, a través de un minucioso análisis, las particularidades de estas nuevas opciones de valor agregado y sus capacidades e importancia en relación con las novedosas y más actuales tendencias tecnológicas paralelas al continuo desarrollo web, lo que puso de manifiesto las ventajas que cada una de ellas podría ofrecer de ser finalmente incorporadas.

El sistema brindará a todo usuario que acceda a este la posibilidad de hacer consultas personalizadas a través de distintos filtros de búsqueda de cada ejemplar, donde se le mostrará la ficha correspondiente al mismo y si es un usuario registrado también tendrá la posibilidad de dejar su opinión a través de la sección dedicada a los comentarios, permitiendo el intercambio y enriquecimiento entre usuarios y especialistas, con este mismo fin y además para establecer un intercambio más personal entre ambas partes contará con un sistema de mensajería interna compuesto por una bandeja de entrada para cada usuario registrado, la opción de exportar cada ficha de ejemplar a formato pdf con dos plantillas distintas también formará parte del conjunto de funcionalidades disponibles.

Contará con una sección de noticias donde los usuarios podrán estar al tanto de los cambios que se realicen en las colecciones, las nuevas inserciones de ejemplares y todo lo relevante respecto a las mismas.

Los especialistas y administradores del sistema serán los encargados de gestionar (crear, insertar, modificar y eliminar) toda la información que se publicará en el sitio y dar soporte a todas las funcionalidades que ofrecerá el sistema. En caso de que sea necesario también podrá otorgar ciertos privilegios a determinados usuarios referentes al acceso y forma de utilizar las distintas secciones y servicios que brinde la aplicación.

2.4. FASES DEFINIDAS POR XP

La metodología XP está compuesta por cuatro fases, las cuales se describen en el transcurso de este epígrafe, ellas son: Exploración, Planificación, Implementación y Pruebas.

2.5. FASE DE EXPLORACIÓN

En esta fase se define el alcance general del proyecto siendo el cliente quien precisa lo que necesita mediante la redacción de sencillas historias de usuarios. Esta fase dura aproximadamente un par de semanas y el resultado es una visión general del sistema y un plazo total estimado. (43)

2.5.1. Historias de usuario

Las historias de usuarios (HU) son escritas por el cliente en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar, aunque los desarrolladores también pueden ayudar en la descripción de las mismas. Las historias de usuario deben ser concretas y

sencillas para que los programadores puedan realizar una estimación poco riesgosa del tiempo que llevará su desarrollo. En este proceso el cliente identificó quince historias de usuarios las cuales serán detalladas a continuación:

- 1 - Registrar Usuario
- 2 - Gestionar Usuario por Roles
- 3 - Consultar Usuario por Roles
- 4 - Consultar Estadísticas
- 5 - Gestionar Ejemplares
- 6 - Consultar Ejemplares
- 7 - Gestionar Noticia
- 8 - Consultar Noticia
- 9 - Gestionar Comentario Asociado a un Ejemplar
- 10 - Consultar Comentario Asociado a un Ejemplar
- 11 - Buscar Elemento por Criterio
- 12 - Mostrar Resultados de la Búsqueda
- 13 - Exportar a PDF
- 14 - Gestionar Mensajería Interna
- 15 - Consultar Mensajería Interna

Tabla 2.1 Plantilla de Historia de Usuario

Historia de Usuario	
Nombre: identifica la historia de usuario en cuestión.	Ptos. Estimación: permite estimar duración de implementación en semanas.
No.: número sucesivo a partir de 1.	Usuario: quién ejecuta la historia de usuario.
Iteración: precisa la iteración a la que pertenece la historia de usuario.	
Descripción: explica en qué consiste la historia de usuario, teniendo en cuenta las acciones realizadas por el usuario y las respuestas brindadas por el sistema.	
Prioridad: define la relevancia e impacto de la historia de usuario para el negocio de acuerdo con las necesidades del usuario.	Nivel de complejidad: define la dificultad técnica que supone desarrollar la historia de usuario desde el punto de vista del programador.

Información adicional: información extra que se estime necesaria agregar para hacer más comprensible la historia de usuario, como conceptos y precondiciones.

Tabla 2.2 HU Registrar Usuario

Historia de Usuario		
Nombre: Registrar Usuario		Ptos. Estimación: 0.5
No.: 1	Usuario: Todos	Iteración: 3
Descripción: comienza cuando un usuario selecciona la opción de registrarse. El sistema debe mostrar un pequeño formulario con tres datos de obligatorio llenado: nombre de usuario, contraseña y correo electrónico, además un grupo de datos opcionales de contacto para enriquecer su perfil como usuario.		
Prioridad: Baja		Nivel de complejidad: Medio
Información adicional: una vez registrado tendrá el privilegio de poder acceder al módulo Comentarios en la aplicación y hacer uso del mismo.		

Tabla 2.3 HU Gestionar Usuario por Roles

Historia de Usuario		
Nombre: Gestionar usuario por roles		Ptos. Estimación: 0.5
No.: 2	Usuario: Administrador, Especialista Superior	Iteración: 2
Descripción: inicia en el instante en que el usuario selecciona la opción de gestionar un determinado usuario. El sistema debe permitir acceder al módulo Usuarios del panel de administración de Drupal donde tendrá la posibilidad de añadir, ver, editar o eliminar los datos de los usuarios registrados en la aplicación.		
Prioridad: Alto		Nivel de complejidad: Alto
Información adicional:		

Las descripciones de las restantes historias de usuarios están contenidas en el [Anexo 8](#)

2.6. FASE PLANIFICACIÓN

En esta fase el cliente y los desarrolladores acuerdan el orden en que deberán implementarse las historias de usuario y asociadas a estas las entregas. Esta fase consiste en una o varias reuniones grupales de planificación y el resultado de esta fase es un plan de entregas. Después de ser descritas e identificadas las historias de usuario y estimado el esfuerzo

propuesto para la realización de cada una de ellas, se procede a la planificación de la etapa de implementación del sistema. En este plan se especifica exactamente cuáles historias de usuario serán implementadas para cada iteración del sistema y las posibles fechas para sus liberaciones. (43)

2.6.1. Estimación de Esfuerzo por Historia de Usuario

En esta fase de la metodología XP se realiza una estimación de la duración en semanas que tendrá el desarrollo del producto. Cada uno de los puntos significa una semana de aprovechamiento total en el desarrollo del producto y se estima a partir de la complejidad de realización de cada una de las HU. A continuación se muestra una tabla con este proceso realizado por los programadores, que asignaron puntos de estimación según lo expresado por el cliente.

Tabla 2.17 Estimación de Esfuerzo por Historias de Usuario

No.	Historias de Usuario	Ptos. (semanas)	Estimación
1	Registrar Usuario	0.5	
2	Gestionar Usuario por Roles	0.5	
3	Consultar Usuario por Roles	0.5	
4	Consultar Estadísticas	0.5	
5	Gestionar Ejemplares	1	
6	Consultar Ejemplares	1	
7	Gestionar Noticia	0.5	
8	Consultar Noticia	0.25	
9	Gestionar Comentario Asociado a un Ejemplar	0.5	
10	Consultar Comentario Asociado a un Ejemplar	0.25	
11	Buscar Elemento por Criterio	0.25	
12	Mostrar Resultados de la Búsqueda	0.5	
13	Exportar a PDF	0.75	
14	Gestionar Mensajería Interna	0.75	
15	Consultar Mensajería Interna	0.25	
	Total	15	

2.6.2. Plan de iteraciones

Las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración de acuerdo con el orden preestablecido. Este plan de iteraciones especifica

qué historias de usuario serán implementadas para cada iteración y las fechas en que serán liberadas.

Tabla 2.17.1. Estimación de Esfuerzo por Historias de Usuario

Iteraciones	Descripción de la Iteración	Orden de las Historias de Usuario	Duración Total
Iteración 1	En esta iteración están las historias de usuarios que son importantes para la estructura y el diseño de la aplicación. Son las que tienen el peso del contenido, es decir, las de prioridad alta. De esta forma, se van creando las funcionalidades principales del sistema que dan soporte a la implementación de las demás funcionalidades. Se tendrá la primera versión de prueba, la cual será mostrada al cliente para obtener una retroalimentación para el posterior desarrollo de la aplicación.	Gestionar Ejemplares	1
		Consultar Ejemplares	1
Iteración 2	En esta iteración se desarrollan las restantes historias de usuarios con prioridad media alta respectivamente. Estas funcionalidades le brindan al cliente la posibilidad de consultar detalles y características de los contenidos de la aplicación. La versión de pruebas de esta iteración además de la anterior, serán mostradas al cliente con el objetivo de obtener su valoración y los cambios en caso que se encuentren.	Gestionar Usuario por Roles	0.5
		Consultar Usuario por Roles	0.5
		Gestionar Comentario Asociado a un Ejemplar	0.5
		Buscar Elemento por Criterio	0.25
		Mostrar Resultados de la Búsqueda	0.5
		Exportar a PDF	0.75
Iteración 3	En esta iteración se continuará con la implementación de las historias de usuarios de prioridad media y baja. Además, se tendrá una versión 1.0 del	Registrar Usuario	0.5
		Consultar Estadísticas	0.5
		Gestionar Noticia	0.5
		Consultar Noticia	0.25

producto final, donde a partir de ese momento, el sistema será puesto a pruebas por un período de tiempo para evaluar su desempeño.	Consultar Comentario Asociado a un Ejemplar	0.25
	Gestionar Mensajería Interna	0.75
	Consultar Mensajería Interna	0.25

2.6.3. Plan de Duración de Iteraciones

El plan de duración de iteraciones se realiza para el único equipo de desarrollo existente, estimando el tiempo que demora la implementación según la historia de usuario propuesta y estableciendo un orden de prioridad a medida que se implementan las historias de usuario de cada iteración.

Tabla 2.17.2. Plan de iteraciones

Iteraciones	Orden de las Historias de Usuario a Implementar	Duración Total
Iteración 1	Gestionar Ejemplares Consultar Ejemplares	2 semanas
Iteración 2	Gestionar Usuario por Roles Consultar Usuario por Roles Gestionar Comentario Asociado a un Ejemplar Buscar Elemento por Criterio Mostrar Resultados de la Búsqueda Exportar a PDF	3 semanas
Iteración 3	Registrar Usuario Consultar Estadísticas Gestionar Noticia Consultar Noticia Consultar Comentario Asociado a un Ejemplar Gestionar Mensajería Interna Consultar Mensajería Interna	3 semanas

2.6.4. Plan de Entregas

En el plan de entregas se indican las historias de usuario que se crearán para cada versión del programa y las fechas en las que se publicaron estas versiones. Un plan de entrega es una planificación donde los desarrolladores y clientes establecen los tiempos de implementación

ideales de las HU, la prioridad con la que serán realizadas y las que serán implementadas en cada versión del programa. Para realizar el plan de entregas, el sistema propuesto se dividió en módulos que contienen las historias de usuario relacionadas lógicamente de acuerdo a su propósito. Cada una de las entregas se correspondió con una iteración de desarrollo.

Tabla 2.18 Historias de Usuario por Módulos

Módulos	Historias de Usuario que Abarca
Usuarios	Registrar Usuario Gestionar Usuario por Roles Consultar Usuario por Roles
Grupos Taxonómicos	Gestionar Ejemplares Consultar Ejemplares
Noticias	Gestionar Noticia Consultar Noticia
Mensajes	Gestionar Mensajería Interna Consultar Mensajería Interna
Comentarios	Gestionar Comentario Asociado a un Ejemplar Consultar Comentario Asociado a un Ejemplar
Estadísticas	Consultar Estadísticas
Exportar a PDF	Exportar a PDF
Búsquedas	Buscar Elemento por Criterio Mostrar Resultados de la Búsqueda

A continuación se presenta el plan de entrega:

Tabla 2.19 Plan de Entrega

Módulos	Primera Iteración (cuarta semana de enero)	Segunda Iteración (primera semana de marzo)	Tercera Iteración (segunda semana de abril)
Usuarios		v0.1	v0.2
Grupos Taxonómicos	v0.1		
Noticias			v0.1
Mensajes			V0.1
Comentarios		v0.1	v0.2
Estadísticas			v0.1
Exportar a PDF		v0.1	
Búsquedas		v0.1	

En los epígrafes que se describen a continuación se hace referencia a los patrones arquitectónico y de diseño empleados en el desarrollo del sistema.

2.7. PATRÓN ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR

El patrón conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- **Modelo:** el modelo administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).
- **Vista:** maneja la visualización de la información.
- **Controlador:** interpreta las acciones del ratón y el teclado, informando al modelo y a la vista para que cambien según resulte apropiado.

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo independientemente de la representación visual. La separación entre vista y controlador puede ser secundaria en algunas aplicaciones. En aplicaciones web, por otra parte, la separación entre la vista (el browser) y el controlador (los componentes del lado del servidor que manejan los requerimientos de HTTP) está mucho más taxativamente definida.

A grandes rasgos este tipo de arquitectura busca distribuir la aplicación en estas tres partes diferentes que se comunican entre sí, garantizando aplicaciones de fácil mantenimiento y escalables. (33)

Drupal concibe en su esencia una fuerte separación entre los desarrollos de la interacción directa con la base de datos (modelo), la interfaz gráfica (vista) y la interacción entre administración de los datos y su representación (controlador), aunque no ha sido concebido netamente bajo este patrón de desarrollo. (34)

2.8. PATRONES DE DISEÑO

2.8.1. Programación orientada a objetos en Drupal

A pesar de la falta de clases explícitamente declaradas en Drupal, muchos paradigmas orientados a objetos se utilizan en su diseño. Varios son los rasgos necesarios para clasificar como sistema orientado a objetos, algunas de estas definiciones son exhibidas en las

características de Drupal como son objeto, abstracción, encapsulación, polimorfismo y herencia. (35)

Objetos

Muchas de las construcciones en Drupal se ajustan a la descripción de “objeto” como son los módulos, temas, nodos, y usuarios.

Los nodos son los bloques de construcción básicos de los contenidos de un sitio de Drupal. Los métodos que funcionan en este objeto se definen en `node.module`, usualmente llamado por el `node_invoke ()` función y los objetos de usuario de paquetes de datos en conjunto funcionan de manera similar, reuniendo información acerca de cada cuenta en el sitio, información de perfil y período de sesiones de seguimiento. En ambos casos, la estructura de datos se define mediante una tabla de base de datos en lugar de una clase. Drupal explota la naturaleza relacional de bases de datos para permitir que otros módulos amplíen los objetos con campos de datos adicionales. (35)

Abstracción

Los hooks definen las operaciones que se pueden realizar en un módulo. Si un módulo implementa un hook se entra en un contrato para realizar una tarea en particular cuando el hook se invoca. El código solo necesita saber acerca del módulo o la forma en que el hook se aplica con el fin de obtener un trabajo útil hecho invocando el hook. (35)

Encapsulación

Como la mayoría de otros sistemas orientados a objetos Drupal no tiene estrictamente una forma de limitar el acceso a un objeto del funcionamiento interno, sino que se basa en la convención para lograr esto. Su código se basa alrededor de las funciones que comparten un mismo nombre, este nombre está subdividido por el uso de prefijos. Siguiendo esta simple convención cada módulo puede declarar sus propias funciones y variables sin entrar en conflicto con los demás. (35)

Polimorfismo

Los nodos son polimórficos en el sentido clásico, si un módulo tiene que mostrar un nodo, por ejemplo, se puede llamar `node_view ()` en ese nodo para obtener una representación HTML. La representación real, sin embargo, dependerá de que tipo de nodo se pasa a la función, que es directamente análoga a tener la clase de un objeto y determinar su comportamiento cuando se envía un mensaje a ella. Drupal se encarga de las mismas tareas que la introspección de un lenguaje orientado a objetos de la biblioteca en tiempo de ejecución.

Además, la prestación del nodo en este ejemplo puede verse afectada por el tema activo. Los

temas son polimórficos, de la misma manera, el tema se pasa a un mensaje “render this node” y responde a ella de una manera diferente dependiendo de la aplicación del tema activo, aunque la interfaz es constante. (35)

Herencia

Los módulos y los temas pueden definir funciones. Sin embargo, puede ser pensado a heredar su comportamiento de una clase base abstracta. En el caso de los temas, el comportamiento de esta clase está determinada por las funciones de theme.inc, si un tema no se anula existe una función definida (por defecto la versión del componente de interfaz que se utiliza) pero el tema puede prever su propio renderizado. (35)

2.8.2. Patrones

Los patrones son soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos. Sus soluciones están basadas en los problemas del diseño que se repiten y que se presentan en situaciones particulares.

Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

Gran parte de la estructura interna de Drupal es más complicada que un simple mensaje de pasar la herencia, sin embargo, las características más interesantes del sistema establecido, resultantes del uso en los software de patrones de diseño y muchas de las pautas detalladas en los patrones se evidencian en el mismo. (36)

Patrón Singleton (instancia única)

Patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.

Si se piensa en los módulos y temas como objetos y luego siguiendo el patrón singleton, entonces estos objetos no encapsulan datos, lo que separa a un módulo desde otro es el conjunto de funciones que contiene, por lo que debe ser pensado como una clase con un ejemplo aislado. (35)

Patrón Decorator (envoltorio)

Patrón estructural encargado de añadir dinámicamente nuevas responsabilidades a un objeto, proporcionando una alternativa flexible a la herencia para extender la funcionalidad.

Drupal hace amplio uso de este patrón a través del polimorfismo de nodo de objetos y el uso de hook_nodeapi (), que permite a los módulos extender arbitrariamente el comportamiento de

todos los nodos.

Esta característica permite una amplia variedad de comportamientos que se añade a los nodos. Por ejemplo, un nodo comentario tiene sólo unas pocas piezas de los datos asociados: título, resumen, cuerpo, usuario, y algunos metadatos. Una necesidad común es que los archivos cargados se adjunten a un nodo, para lo que se podría diseñar un nuevo tipo de nodo con las características de uno de tipo comentario y además con la posibilidad de adjuntar archivos. El módulo upload de Drupal satisface esta necesidad de forma mucho más modular mediante la concesión de cada nodo que solicite la posibilidad de tener archivos adjuntos.

Este comportamiento podría ser imitado por el uso de los decoradores, envolviéndolos alrededor de cada nodo objeto. Más sencillamente, los idiomas que apoyan a las categorías, podrían aumentar la base común de toda clase de objetos nodo para añadir el nuevo comportamiento. La implementación de Drupal es una simple ramificación del sistema de hooks y la presencia del `node_invoke_all()`. (35)

Patrón Observer (observador)

Patrón de comportamiento encargado de definir un objeto que coordine la comunicación entre objetos de distintas clases, pero que funcionan como un conjunto.

El patrón Observer es generalizado en Drupal. Cuando una modificación es hecha a un vocabulario en el sistema de taxonomía de Drupal, el hook taxonomy es llamado en todos los módulos que lo implementan y mediante la aplicación del hook se registran como observadores del objeto vocabulario; lo que asegura poder actuar como es apropiado ante cualquier cambio. (35)

Patrón Bridge (puente)

La capa de abstracción de base de datos de Drupal se aplica de una forma similar al patrón estructural Bridge. Los módulos deben ser escritos independientemente del sistema de base de datos que se está utilizando, y la capa de abstracción esta proporcionada para ello. Se pueden definir nuevas capas de dicho sistema de acuerdo a la API definida, añadiendo soporte para otros sistemas sin necesidad de modificar el código del módulo. Desacoplando de esta manera la abstracción de su implementación. (35)

Patrón Chain of Responsibility (cadena de responsabilidad)

Patrón de comportamiento que permite establecer la línea que deben llevar los mensajes para que los objetos realicen la tarea indicada.

En cada solicitud de las páginas, el menú del sistema determina si hay un módulo para gestionar la solicitud, si el usuario tiene acceso a los recursos solicitados y que la función sea llamada para hacer el trabajo. Para ello, el mensaje se pasa a la opción del menú

correspondiente a la vía de la solicitud y si el elemento de menú no puede manejar la petición, se pasa a otro eslabón de la cadena. Esto continúa hasta que un módulo se encargue de la petición, un módulo niegue el acceso para el usuario o se haya agotado la cadena. (35)

Patrón Command (orden)

Patrón de comportamiento que encapsula una operación en un objeto, permitiendo ejecutar dicha operación sin necesidad de conocer el contenido de la misma.

Muchos de los hooks de Drupal utilizan el patrón Command para reducir el número de funciones que son necesarias para la aplicación, pasando por la operación como un parámetro, junto con los argumentos. De hecho, el propio sistema de hook utiliza este modelo, a fin de que los módulos no tiene que definir cada hook, sino solo los que vayan aplicar. (35)

2.9. DISEÑO DE LA BASE DE DATOS

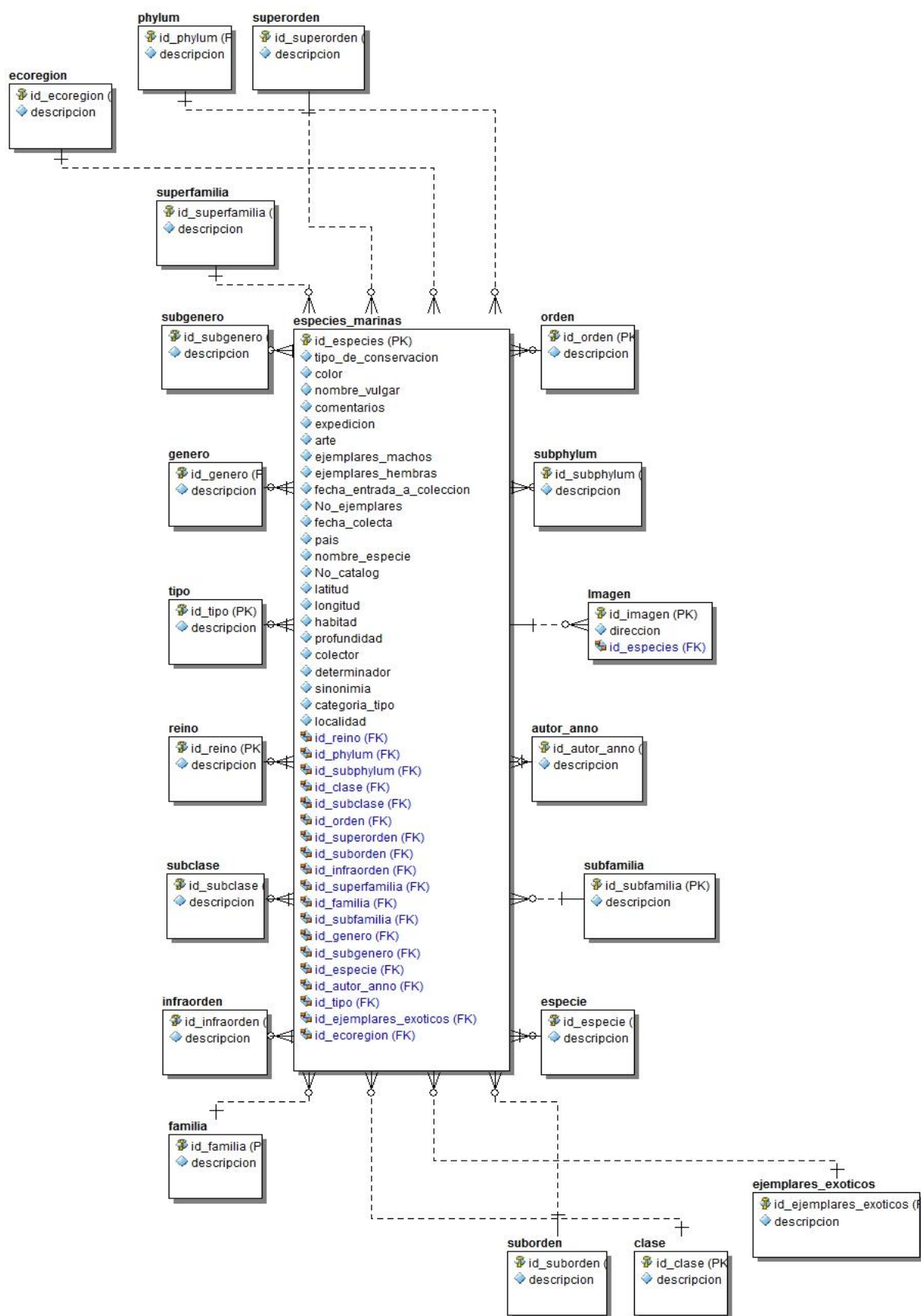
Las bases de datos necesitan de una definición de su estructura que le permita almacenar datos, reconocer el contenido, y recuperar la información. La estructura tiene que ser desarrollada para satisfacer las necesidades de las aplicaciones que la usarán.

2.9.1. Modelo Entidad-Relación

Un Diagrama o Modelo Entidad Relación (a veces denominado por su siglas, E-R "EntityRelationship", o, "DER" Diagrama Entidad Relación), es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información así como sus interrelaciones y propiedades.

La base de datos que se muestra representa los 42 campos en los que se expresan las características de cada ejemplar sin complejas relaciones entre ellos, de los cuales 19 son de tipo nomencladores.

Figura 7. Diseño de base de datos



2.10. CONCLUSIONES DEL CAPÍTULO

En el desarrollo de este capítulo, se crearon los artefactos principales para las fases Exploración y Planificación, donde el cliente redacta las historias de usuario a través de una breve descripción de lo que espera haga la aplicación. Se creó el plan de iteraciones agrupando las historias de usuario según la prioridad asignada a cada una y se realizó el plan de entregas donde se estimó el tiempo de desarrollo en semanas para cada una. Además se representa el modelo arquitectónico empleado, así como algunos patrones de diseño de la programación orientada a objetos y el modelado de datos del sistema.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS

3.1. INTRODUCCIÓN

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para su opinión y para incrementar la visión de los desarrolladores. En el presente capítulo se detallan las tres iteraciones llevadas a cabo durante la etapa de construcción del sistema, exponiéndose las tareas generadas por cada Historia de Usuario, así como las pruebas de aceptación efectuadas sobre el sistema.

3.2. DISEÑO DEL SISTEMA

Para el diseño de la aplicación, la metodología XP no requiere la presentación del sistema mediante diagramas de clases utilizando notación UML; en su lugar, se usan otras técnicas como las tarjetas CRC (Contenido, Responsabilidad y Colaboración). No obstante, el uso de estos diagramas puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante. (31)

Con el objetivo de la comprensión de este trabajo se hace una breve explicación del funcionamiento de Drupal como plataforma de publicación.

El CMS Drupal contiene un tipo de contenido genérico llamado Node que puede ser extendido por cualquier desarrollador. Este tipo de contenido tiene las propiedades básicas para distintas publicaciones como son título, autor, fecha de creación y contenido; además, proporciona los mecanismos para la creación, edición y publicación de este tipo de contenido. Los desarrolladores que deseen una publicación personalizada, sólo deben extender el Node y de esta manera, aprovechar sus propiedades. Drupal provee al desarrollador de un potente sistema de seguridad basado en roles, el mismo CMS se encarga de la creación de usuarios y roles, así como del control de accesos a los diferentes módulos según los permisos definidos por el administrador. El desarrollador se limita a exportar en su módulo los tipos de acceso que desea definir, el resto lo maneja Drupal, es decir, cuando un usuario trata de acceder a un módulo, la plataforma chequea que el usuario autenticado tenga acceso al módulo en cuestión.

3.3. MÓDULOS DE DRUPAL

Los módulos son el principal mecanismo para extender las funcionalidades de Drupal. Implementan una interfaz bien definida que permite la interacción con el Núcleo. Un módulo es la unión de varias funciones que se juntan en Drupal y ayudan a ofrecerle mayor funcionalidad a la web. Drupal llama hooks a las funciones que forman esta interfaz. Los hooks los podemos

agrupar en 3 categorías: -Autenticación: proveen mecanismos adicionales de autenticación de usuarios. –Núcleo (Core): necesario para responder e interactuar con el corazón de Drupal. - Node: provee un nuevo tipo de nodo al sistema. Un módulo para Drupal consta de uno o más ficheros, el fichero principal con extensión .module, .info, .install. Básicamente existen dos tipos de módulos: los módulos de contenido, que son los que definen un nuevo tipo de contenido personalizado y la funcionalidad para su creación, edición y publicación y los módulos funcionales, estos tienen disímiles propósitos dependiendo del objetivo con el que se desarrollan. La tarea de estas funciones es actuar como enganche, al ser llamadas por Drupal a la hora de construir una página web y gestionar el contenido. El sistema de bloques de Drupal consiste en una serie de bloques (ya sean definidos por un módulo o en la misma interfaz de Drupal) que se activan o desactivan para ser mostrados en las áreas de menú de la plataforma. Se utilizará el término “paquete” para referirse a módulos del sistema de gestión de contenidos en cuestión.

Las descripciones de los paquetes y subpaquetes de Drupal pueden encontrarse en [Anexo 10](#).

3.4. TARJETAS CRC

Las tarjetas CRC permiten trabajar con la metodología basada en objetos. Posibilitan que el equipo completo contribuya en la tarea del diseño. Una tarjeta CRC representa un objeto. El nombre de la clase se coloca a modo de título en la tarjeta, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente. (43)

Tabla 3.20 Plantilla de Tarjeta CRC

Clase: Nombre de la clase que se está modelando.	
Responsabilidades	Clases relacionadas (módulos)
Es una descripción de alto nivel del propósito de la clase.	Indica con qué clases requiere relación, para cumplir la responsabilidad.

Tabla 3.21 Tarjeta CRC Usuarios

Módulo Usuarios	
Responsabilidades	Clases relacionadas (módulos)
Adicionar usuario	Mensajes
Editar usuario	Comentarios
Eliminar usuario	Estadísticas
Consultar usuario	

Tabla 3.22 Tarjeta CRC Grupos Taxonómicos

Módulo Grupos Taxonómicos	
Responsabilidades	Clases relacionadas (módulos)
Adicionar ejemplar	Usuarios
Editar ejemplar	Comentarios
Eliminar ejemplar	Exportar a PDF
Consultar ejemplar	Búsquedas
	Estadísticas

Tabla 3.23 Tarjeta CRC Noticias

Módulo Noticias	
Responsabilidades	Clases relacionadas (módulos)
Adicionar noticias	Comentarios
Editar noticias	Exportar a PDF
Eliminar noticias	
Consultar noticias	

Tabla 3.24 Tarjeta CRC Mensajes

Módulo Mensajes	
Responsabilidades	Clases relacionadas (módulos)
Adicionar mensajes	Usuarios
Editar mensajes	
Eliminar mensajes	
Consultar mensajes	

Tabla 3.25 Tarjeta CRC Comentarios

Módulo Comentarios	
Responsabilidades	Clases relacionadas (módulos)
Adicionar comentarios	Usuarios
Editar comentarios	Grupos Taxonómicos
Eliminar comentarios	
Consultar comentarios	

Tabla 3.26 Tarjeta CRC Estadísticas

Módulo Estadísticas

Responsabilidades	Clases relacionadas (módulos)
Consultar estadísticas	Usuarios Grupos Taxonómicos

Tabla 3.27 Tarjeta CRC Exportar a PDF

Módulo Exportar a PDF	
Responsabilidades	Clases relacionadas (módulos)
Crear PDF	Grupos Taxonómicos

Tabla 3.28 Tarjeta CRC Búsquedas

Módulo Búsquedas	
Responsabilidades	Clases relacionadas (módulos)
Consultar resultados	Grupos Taxonómicos Noticias Comentarios

3.5. ESTÁNDARES DE CODIFICACIÓN

Los estándares de codificación son pautas que se utilizan en la escritura del código fuente, las mismas aseguran de que todos los programadores del proyecto mantengan un vocabulario común. La utilización de los mismos es muy ventajosa, ya que permite asegurar la legibilidad del código entre distintos programadores, le provee una guía para el encargado del mantenimiento o actualización del sistema, con código claro y bien documentado y además facilita la portabilidad entre plataformas y aplicaciones. (49)

3.5.1. Convenciones de nombres (Funciones, Variables Globales)

Los nombres que se utilizan son cortos, y de forma abreviada, siempre que su contexto sea específicamente local y su lectura sea intuitiva. Para los nombres se establecen las siguientes reglas:

- ✓ **Funciones y métodos:** se utiliza la notación Camel que consiste en escribir los identificadores con la primera letra de cada palabra en mayúsculas y el resto en minúscula: EndOfFile. Las funciones y métodos se nombran usando el formato CamelCase (notación camello) que es una variante de la notación Camel donde la primera letra es minúscula.
- ✓ **Clases:** Los nombres de las clases comienzan con el sufijo del módulo en este caso del



módulo Resultados seguidos del formato CamelCase ejemplo reMenuPrincipal.

Las clases que se generan en el Modelo-Entidad-Relación comienzan con:

- Tb_: las clases de entidad.
- Nom_: las clases de nomencladores.
- R_: las clases que representan relaciones muchos-a-muchos.

3.5.2. Indentación y espacios en blanco

Alineación y longitud de líneas

Indentar con dos espacios, sin tabulador, para que cualquier editor de texto reconozca correctamente la indentación. Utilizar líneas entre 75-80 caracteres de longitud, de esta manera se maneja mejor algunas herramientas y terminales.

Condicionales if, for, while, switch

Debe existir un espacio entre la palabra clave y el paréntesis de apertura. Esto mejora la legibilidad y disminuye la posibilidad de errores lógicos al agregar nuevas líneas de código.

Líneas plegadas

Cuando una expresión no cabe en una línea simple debido a su extensión se divide en más de una línea, siguiendo las siguientes precisiones:

- Dividir después de una coma.
- Dividir después de un operador.
- Alinear la nueva línea al inicio de la expresión en el mismo nivel que la línea anterior.

3.5.3. Comentarios

El uso de comentarios en línea se utiliza para facilitar la comprensión del código, sobre todo en procedimientos complejos. Los comentarios pueden ser con fin documental o para ayudar a la memoria del programador. Se recomienda utilizar el estilo (*/* */*).

En el caso de los bloques de comentarios tienen solo un asterisco al inicio de cada línea exceptuando la primera. Ejemplo:

```
/**  
    *Comentario  
*/
```

Los comentarios cortos pueden aparecer en una sola línea alineados al mismo nivel del código que lo sigue, además debe estar precedido de una línea en blanco. Ejemplo:

```
/* Comentario*/
```

3.6. MODELO DE IMPLEMENTACIÓN

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros. El modelo de implementación define una jerarquía. (37)

3.7. FASE DE IMPLEMENTACIÓN

Durante el transcurso de las iteraciones, se realiza la implementación de las historias de usuario seleccionadas para ser realizadas en cada una de ellas. Al principio de estas se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de desarrollo, asignando a un grupo de desarrollo (o una persona), la responsabilidad de su implementación. Estas tareas son para el uso estricto de los programadores, pueden ser escritas en lenguaje técnico y no necesariamente entendible por el cliente. (43)

Teniendo en cuenta la planificación realizada anteriormente, se llevaron a cabo tres iteraciones de desarrollo sobre el sistema, obteniéndose como finalidad un producto con todas las restricciones y características deseadas para ser utilizado. A continuación se detallan las iteraciones.

3.7.1. Iteración 1

En esta iteración se implementaron las historias de usuario de mayor prioridad, con el fin de obtener una versión del producto con algunas de las funcionalidades críticas para ser mostrado al cliente y tomar nuevas iniciativas de forma rápida.

Tabla 3.29 Módulos abordados en la primera iteración

Módulo	Historia de Usuario	Tiempo de Implementación	
		Estimación	Real
Grupos Taxonómicos	Gestionar Ejemplares	1	1
	Consultar Ejemplares	1	0.5

A continuación se muestran las tareas efectuadas para cada uno de los módulos implementados en esta iteración:

Tabla 3.30 Tarea 1 Módulo Grupos Taxonómicos

Tarea	
Número de Tarea: 1	Número de Historia de Usuario: 5
Nombre de la Tarea: Gestionar ejemplares asociados a sus respectivos grupos taxonómicos.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 28 de Febrero del 2012	Fecha Fin: 6 de Marzo del 2012
Programador Responsable: Javier Núñez	
Descripción: se implementarán las funcionalidades y las tablas necesarias para ejercer la correcta gestión del módulo Grupos Taxonómicos, además de la asignación de los permisos requeridos por roles para acceder al mismo.	

Las tablas de los módulos abordados de las iteraciones 2 y 3 junto a sus tareas correspondientes se encuentran descritas en el [Anexo 11](#).

3.8. FASE DE PRUEBAS

Unos de los pasos más importantes de la metodología XP es el proceso de pruebas, el cual anima a los desarrolladores a probar constantemente la aplicación tanto como sea posible. Mediante esta filosofía se reduce el número de errores no detectados, así como el tiempo entre la introducción de éste en el sistema y su detección. Todo esto contribuye a elevar la calidad de los productos desarrollados y a la seguridad de los programadores, a la hora de introducir cambios o modificaciones. La metodología XP divide las pruebas en dos grupos: pruebas unitarias; desarrolladas por los programadores, encargadas de verificar el código de forma automática y las pruebas de aceptación; destinadas a evaluar si al final de una iteración se obtuvo la funcionalidad requerida, además de comprobar que dicha funcionalidad sea la esperada por el cliente. (33)

3.8.1. Pruebas unitarias

Las pruebas unitarias aseguran que un único componente de la aplicación produce una salida correcta para una determinada entrada. Este tipo de pruebas validan la forma en la que las funciones y métodos trabajan en cada caso particular. Las pruebas unitarias se encargan de un único caso cada vez, lo que significa que un único método puede necesitar varias pruebas unitarias si su funcionamiento varía en función del contexto. Estas pruebas aisladas proporcionan cinco ventajas básicas, fomentan el cambio, simplifica la integración, documenta el código, separa la interfaz del código y hace que los errores estén más acotados y sean fáciles de localizar.

XP aconseja que la realización de las pruebas unitarias al sistema sea en etapas tempranas del desarrollo permitiendo disminuir la ocurrencia de defectos y aprovechar las ventajas de la retroalimentación que se produce en el proceso. Aunque la tarea de crear casos de prueba manuales, que luego serán ejecutadas y analizados sus resultados, es extenuante pueden utilizarse complementariamente a las anteriores. (41)

En el desarrollo de la aplicación se utilizaron este tipo de pruebas principalmente en el desarrollo de los módulos que fueron desarrollados por el equipo de trabajo. Se realizaron en el proceso de escritura del código y era revisado de forma alternada, garantizando así que quien escribiera el código no fuese quien lo revisara.

3.8.2. Pruebas de aceptación

Las pruebas de aceptación son pruebas de caja negra que se crean a partir de las historias de usuario. Durante las iteraciones las HU seleccionadas serán traducidas a pruebas de aceptación. En ellas se especifican, desde la perspectiva del cliente, los escenarios para probar que una HU ha sido implementada correctamente. Una HU puede tener todas las pruebas de aceptación que necesite para asegurar su correcto funcionamiento. El objetivo final de estas es garantizar que los requerimientos han sido cumplidos y que el sistema es aceptable. Una HU no se considera completa, hasta que no ha pasado por las pruebas de aceptación. (33)

Tabla 3.46 Prueba 1 al Módulo Usuarios

Caso de Prueba de Aceptación	
Código: HU1_P1	Historia de Usuario: 1
Nombre: Registrar usuario	
Descripción: prueba para la funcionalidad de registrar un nuevo usuario en la aplicación.	
Condiciones de Ejecución: cualquier usuario anónimo puede acceder al registro desde cualquier parte del sitio a través de un botón ubicado en la parte superior derecha del banner.	
Entrada / Pasos de Ejecución: se intenta añadir los datos requeridos por el formulario de registro de usuario.	
Resultado Esperado: se debe registrar el usuario de manera correcta en la base de datos para su posterior autenticación mostrando un mensaje de aviso e indicando que el proceso fue satisfactorio. En caso que el usuario ignore algunos campos de carácter obligatorio o introduzca datos incorrectos se debe mostrar un mensaje informándole sobre los errores en los que pudo incurrir y además comunicándole que no se pudo completar el proceso de registro.	
Evaluación de la Prueba: prueba satisfactoria.	

Tabla 3.47 Prueba 2 al Módulo Usuarios

Caso de Prueba de Aceptación	
Código: HU2_P2	Historia de Usuario: 2
Nombre: Añadir nuevo usuario por roles	
Descripción: prueba para la funcionalidad de añadir un nuevo usuario por roles a la aplicación.	
Condiciones de Ejecución: únicamente el Administrador del sistema y los Especialistas Superiores están capacitados y cuentan con los permisos necesarios para ejecutar esta función.	
Entrada / Pasos de Ejecución: se intenta añadir un nuevo usuario por roles en la aplicación.	
Resultado Esperado: el proceso se ejecutó de forma correcta.	
Evaluación de la Prueba: prueba satisfactoria.	

Los restantes casos de pruebas hechos a los demás módulos se encuentra en el [Anexo12](#).

Resumen de pruebas

A lo largo del desarrollo de todo el sistema se realizaron los casos de prueba, con el objetivo de comprobar la eficiencia del código escrito y el cumplimiento de los requisitos funcionales planteados por el cliente. Por cada historia de usuario se realizó un caso de prueba, salvo algunas excepciones a las que por su complejidad y tamaño se les realizó más de uno. Para un total de 15 historias de usuario se realizaron 30 casos de prueba, de manera que fueron comprobados todos los posibles escenarios, confirmando que los resultados arrojados de las estimaciones muestran un adecuado nivel de similitud ante los resultados reales.

En la primera iteración de este proceso se detectaron 2 pruebas fallidas correspondientes al módulo Grupos Taxonómicos (HU5_P1 y HU5_P2), las cuales no arrojaron los resultados esperados puesto que presentaron incoherencias visuales y desacoplamiento con la interfaz de la aplicación. Luego de realizar un análisis minucioso se detectó donde radicaban los problemas y los mismos fueron solucionados. Tras lo cual se volvieron a ejecutar los casos insatisfactorios hasta alcanzar finalmente el 100 por ciento de satisfacción.

La diferencia máxima existente entre las curvas de los resultados de estimación osciló entre 0.25-0.5 meses de diferencia debido fundamentalmente a los cambios de requisitos por parte del cliente y a varias afecciones de acceso tecnológico por parte del equipo de desarrollo.

3.8.3. Pruebas de rendimiento, de carga y estrés del software

Pruebas de rendimiento del software

En la ingeniería del software, las pruebas de rendimiento son las pruebas que se realizan,

desde una perspectiva, para determinar cuán rápido se realiza una tarea de un sistema en condiciones particulares de trabajo. También puede servir para validar y verificar otros atributos de la calidad del sistema, tales como la escalabilidad, fiabilidad y uso de los recursos. Las pruebas de rendimiento son un subconjunto de la ingeniería de pruebas, una práctica informática que se esfuerza por mejorar el rendimiento, englobándose en el diseño y la arquitectura de un sistema, incluso antes del esfuerzo inicial de la codificación.

Pruebas de carga

Este es el tipo más sencillo de pruebas de rendimiento. Una prueba de carga se realiza generalmente para observar el comportamiento de una aplicación bajo una cantidad de peticiones esperadas. Esta carga puede ser el número esperado de usuarios concurrentes utilizando la aplicación y que realizan un número específico de transacciones durante el tiempo que dura la carga. Esta prueba puede mostrar los tiempos de respuesta de todas las transacciones importantes de la aplicación. Si la base de datos y el servidor de aplicaciones se monitorizan, entonces esta prueba puede mostrar el cuello de botella en la aplicación.

Pruebas de estrés

Esta prueba se utiliza normalmente para romper la aplicación. Se va doblando el número de usuarios que se agregan a la aplicación y se ejecuta una prueba de carga hasta que se rompe. Este tipo de prueba se realiza para determinar la solidez de la aplicación en los momentos de carga extrema y ayuda a los administradores para determinar si la aplicación rendirá lo suficiente en caso de que la carga real supere a la carga esperada.

Pasos previos a la ejecución

El Sistema de Gestión de las Colecciones Marinas del Acuario Nacional Cuba se desarrolló con planes futuros de publicación en internet. Para velar que la aplicación no sufra errores y garantizar su correcto funcionamiento bajo escenarios críticos se realizaron un conjunto de pruebas de rendimiento, en las que se priorizaron las de carga y estrés. Las mismas se llevaron a cabo en una estación con las siguientes prestaciones:

Asus P5LD2, procesador Intel Celeron R con velocidad de 2.40GHz, memoria RAM 1GB y tarjeta de red Realtek PCIe FE Family Controller a 100 Mbps, sobre un servidor wamp Apache en su versión 2.2 y sistema operativo Windows XP Service Pack 2.

Para establecer la línea base de tiempos de respuesta, se ejecutó una primera ejecución con varias peticiones en serie. Se evitó lanzar ráfagas de peticiones con un sólo ciclo, por ejemplo, 100 usuarios virtuales y un sólo ciclo. Con estas ejecuciones los tiempos no siempre son reales porque la misma herramienta toma su tiempo antes de arrancar el número de hilos solicitados.

Se procuraron más bien varios ciclos en casos de muchos usuarios concurrentes, por ejemplo 100 usuarios virtuales y 3 ciclos.

Se preparó adecuadamente una hoja de cálculo para tomar nota de cada ejecución realizada.

Figura 16. Ejemplo de hoja de cálculo para tomar datos durante la ejecución

	A	B	C	D	E	F
1	Petición	Tiempo de Respuesta (ms)	Usuarios Virtuales	Número de peticiones	Peticiones Fallidas	Observaciones
2						
3						
4						
5						
6						
7						

Se cerraron todas las aplicaciones que pudieran interferir con el proceso. Solo permanecieron abiertas únicamente las estrictamente necesarias.

Ejecución de las pruebas

La ejecución de pruebas de carga debe iniciar con peticiones en serie para conocer los tiempos de la aplicación cuando no está sometida a carga. Después en cada siguiente ejecución se incrementa la exigencia aumentando el número de usuarios virtuales y durante cada incremento se toma nota de los tiempos y consumo de recursos, especialmente de memoria.

A continuación se mostrarán las imágenes asociadas a cada ejecución efectuada. La primera imagen será de la interfaz de consumo de recursos de la herramienta JMeter (JConsole), seguidamente se mostrará otra imagen con el informe agregado de los de resultados y por último la hoja de cálculo con las notas finales.

Prueba 1 Ejecución 1

1 usuario virtual y un ciclo de 20 iteraciones.

Figura 17. JConsole

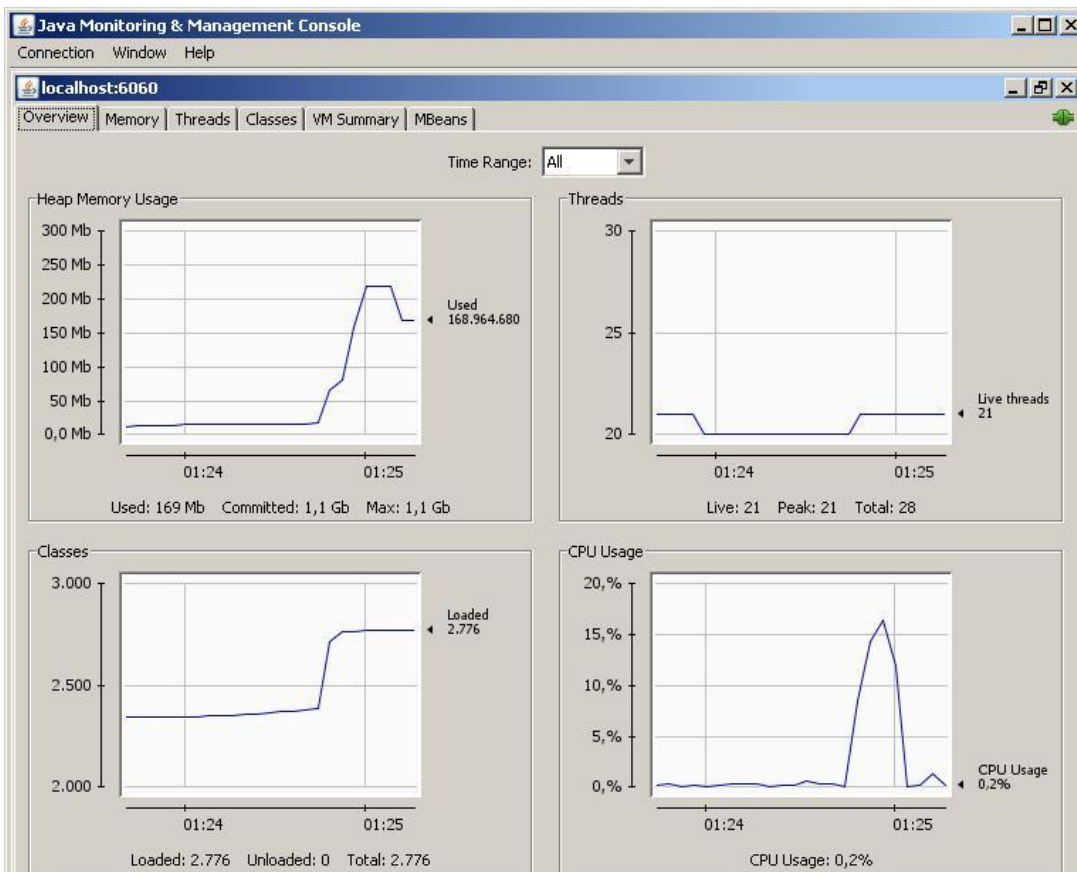


Figura 18. Informe agregado

Label	# Muestras	Media	Mediana	Linea de 90%	Mín	Máx	% Error	Rendimiento	Kb/sec
01. /EjemploPruebaCarga/	20	43	22	89	14	370	0.00%	1.7/sec	227.7
02. /EjemploPruebaCarga/pages/login.jsf	20	4	3	7	2	12	0.00%	1.7/sec	.6
03. /EjemploPruebaCarga/	20	30	21	97	13	116	0.00%	1.7/sec	234.7
04. /EjemploPruebaCarga/pages/login.jsf	20	349	331	454	294	536	0.00%	1.7/sec	229.6
05. /EjemploPruebaCarga/pages/stuff.jsf	20	189	177	236	170	347	0.00%	1.7/sec	233.4
TOTAL	100	123	32	347	2	536	0.00%	8.0/sec	874.7

Figura 19. Datos registrados en la hoja de cálculo

	A	B	C	D	E	F
1	Petición	Tiempo de Respuesta (ms)	Usuarios Virtuales	Número de peticiones	Peticiones Fallidas	Observaciones
2	01. /EjemploPruebaCarga/	43	1	20	0	
3	02. /EjemploPruebaCarga/pages/login.jsf	4	1	20	0	
4	03. /EjemploPruebaCarga/	30	1	20	0	
5	04. /EjemploPruebaCarga/pages/login.jsf	349	1	20	0	
6	05. /EjemploPruebaCarga/pages/stuff.jsf	189	1	20	0	

Conclusiones:

- La prueba fue exitosa y se establecieron tiempos de respuesta base.
- El consumo de memoria para 20 ejecuciones del script fue de aproximadamente 220 MB. Con la memoria asignada podríamos esperar un máximo de 100 ejecuciones.
- El tiempo promedio de respuesta fue de 123 milisegundos.

Los demás casos de pruebas se encuentran en el [Anexo 13](#).

3.9. IMPACTO SOCIAL

El sistema está pensado con el principal objetivo de publicarlo a disposición de los internautas; ofreciendo a todo interesado en la historia e información referente a las colecciones marinas en Cuba, dentro y fuera del país, un total acceso mediante consultas a la ficha completa de cada ejemplar catalogado. Además, constituye un medio para promover la comunicación educativa entre usuarios y especialistas en el tema con fines culturales, investigativos y científicos.

3.10. CONCLUSIONES DEL CAPÍTULO

En el transcurso de este capítulo se construyeron los artefactos principales para las fases Implementación y Pueba, donde se realizó el plan de tareas que permitió desglosar por actividad cada Historia de Usuario para su cumplimiento. Se realizaron las pruebas de aceptación, que permitieron comprobar el correcto funcionamiento de las mismas y que propician al cliente conformidad y seguridad ante el sistema.

CONCLUSIONES GENERALES

Con la realización de la presente investigación se ha dado cumplimiento a los objetivos propuestos, arribando a las siguientes conclusiones:

- ✓ La confección de los principales artefactos correspondientes a los flujos de trabajo de la metodología de desarrollo de software XP sirvieron de apoyo en la implementación de las funcionalidades que permitieron dar una solución rápida y eficiente al problema de la investigación.
- ✓ Se hizo un estudio detallado que permitió conocer el estado de las colecciones marinas pertenecientes al Acuario Nacional Cuba, demostrando así la necesidad de realizar una aplicación con tecnología Drupal que fuese capaz de gestionar las mismas, de forma tal que dotara a los especialistas del acuario de una herramienta robusta y eficaz que garantizara, agilizara, y optimizara con mayor facilidad la documentación de las bases de datos de los más de 60 000 ejemplares en colección.
- ✓ Con el desarrollo del sistema se favorece la gestión de información de ejemplares de las colecciones marinas del Acuario Nacional Cuba por los especialistas de la institución y garantiza su consulta entre usuarios interesados en la sistemática, la taxonomía y la ecología de las distintas especies marinas.
- ✓ El producto resultante tiene un gran valor científico y cultural ya que permite el acercamiento de interesados en el tema desde un entorno de libre acceso a la información recopilada.
- ✓ La aplicación realizada, propiciará que sea altamente configurable y adaptable a cualquier sitio desarrollado sobre el gestor de contenidos Drupal, debido a que las funcionalidades fueron desarrolladas de forma modular, permitiendo además una mejor portabilidad del sistema.
- ✓ A través de las pruebas unitarias llevadas a cabo por el equipo de desarrollo y la confección y aplicación de los casos de pruebas de aceptación se validaron que los requisitos finales levantados con el cliente hayan sido cumplidos.

RECOMENDACIONES

Como resultado de la investigación y del proceso de desarrollo de la aplicación, han surgido ideas que serían recomendables para futuras versiones del sistema y el perfeccionamiento del mismo:

- ✓ La internacionalización de la aplicación, principalmente a los lenguajes inglés y francés.
- ✓ La utilización del sistema en otros departamentos del ANC adaptándolo a las características propias correspondientes de cada departamento.
- ✓ El uso de la documentación generada en el proceso de desarrollo como base para sistemas similares.

REFERENCIAS BIBLIOGRÁFICAS

1. **Mayr, E.** *What is a Species, and What is Not?* s.l. : Philosophy of Science, 1996, vol.63: 262-277.
2. **Brusca, R. C. & Brusca, G. J.** *Invertebrados, 2da edición.* Madrid, España : McGraw-Hill-Interamericana, 2005. ISBN 0-87893-097-3.
3. **Sulekic, Antenor A. y Rugolo de Agrasar, Zulma E.** Una nueva especie de *Gymnopogon*(Poaceae, Cynodonteae). Argentina : s.n., ago/dic 2006. Vol. Vol 44, 2, p 504-507. ISSN 001-6793.
4. **Zuloaga Fernando O., Monroe Osvaldo, Belgrano Manuel J.** Taxonomical and nomenclatural novelties for the Vascular Flora of the Southern Cone of South America. s.l. : Darwiniana, ago/dec 2007. Vol. vol. 45, No. 2 p. 236-241. ISSN 0011-6793.
5. **Toledano, Díaz.** The Architecture of Enterprise Information System. A view based on patterns. [En línea] 2006. [Citado el: 10 de enero de 2012.] <http://www.moisesdaniel.com/wri/eisa.doc>.
6. **Sarduy Domínguez, MsC. Yanetsys y Urra, Lic. Pedro.** Sistemas de gestión de contenidos: En busca de una plataforma ideal. [En línea] 4 de octubre de 2008. [Citado el: 12 de enero de 2012.] http://www.bvs.sld.cu/revistas/aci/vol14_4_08/aci11406.htm.
7. **Huerta García, Cristina.** Creación de sitios web con sistema de gestión de contenidos (CMS)con aspecto personalizado. [En línea] 2011. [Citado el: 18 de enero de 2012.] <http://hdl.handle.net/10115/4175>.
8. *La educacion social y la web 2.0 Nuevos espacios de innovación e interacción social en la Educación Superio.* **Guillermo Domínguez Fernández, Maria del Carmen Llorente.** No. 35 p 105-114, s.l. : Pixel-Bit: Revista de medios y educación, 2009. ISSN 1133-8482.
9. **Museo de altamira.** MCU. [En línea] 2009. [Citado el: 18 de enero de 2012.] <http://museodealtamira.mcu.es/brs/formulario.jsp>.
10. **Universia.** Universia. [En línea] Universia. España, 2009. [Citado el: 18 de enero de 2012.] http://www.universia.es/portada/actualidad/noticia_actualidad.jsp?noticia=101507.
11. **Soporte remoto Inc.** Support. [En línea] Soporte remoto Inc, 2009. [Citado el: 18 de enero de 2012.] http://www.soporteremoto.com.mx/help_desk/articulo03.html.
12. **Encartes online.** Catalogos. [En línea] Catalogos de Peru, 2010. [Citado el: 19 de enero de 2012.] <http://www.deperu.com/catalogos>.
13. *Drupal, el mejor CMS del momento.* **López, Jorge.** No. 116 p 29-93, España : Todo Linux: Revista mensual para entusiastas de GNU/Linux, 2010. ISSN 1188-236.
14. **Peraza Sierra, Yusleidy y Pérez Formantín, Indira.** *GUÍA PARA EL TRABAJO CON EL CMS DRUPAL.* Ciudad de la Habana : s.n., junio de 2007. S.N.
15. **Martín Cruañez, Roberkys, Rey, Enriquez, Derick y Viltres Sala, Hubert.** *Trabajo Investigativo sobre Sistemas de Gestión de Contenido. CMS Drupal.* La Habana : Universidad de las Ciencias Informáticas, 2005. S.N.

16. **BENDOO e-work solutions** . Bendo. [En línea] 2002. [Citado el: 2012 de febrero de 23.] <http://www.bendoo.com/es/typo3-cms/>.
17. **Mozilla**. MOZILLA DEVELOPER NETWORK. [En línea] 2012. [Citado el: 2012 de febrero de 25.] <http://developer.mozilla.org/es/docs/JavaScript>.
18. **Schumacher, R. Lentz, A.** MySQL. [En línea] julio de 2010. [Citado el: 14 de febrero de 2012.] <http://dev.mysql.com/tech-resources/articles/dispelling-the-myths.html>. GPL License.
19. **Bakken, Sæther, Schmid, Stig y Egon**. Manual de php. [En línea] 2003. [Citado el: 20 de enero de 2012.] <http://www.php.net/docs.php>.
20. **Ciencia y Técnica Administrativa: CyTA**. HERRAMIENTAS CASE. [En línea] 2008. [Citado el: 21 de enero de 2012.] <http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/proyectoinformatico/libro/c5/c5.htm>. ISSN 1666-1680.
21. **Matos Padilla, Reynier**. Maestros del web. [En línea] 2002. [Citado el: 25 de enero de 2012.] <http://www.maestrosdelweb.com/editorial/rendimiento-entre-sistemas-gestores-de-contenido-cms>. CC: Creative Commons (CC BY-NC-SA 3.0).
22. **GONZALEZ, YUDISLANDRY AGUILA**. *SISTEMA DE GESTIÓN DE JAVA ARCHIVES*. La Habana, Cuba : s.n., junio, 2009. S.N.
23. **Freddie**. Cristalab. [En línea] 2011. [Citado el: 3 de febrero de 2012.] <http://www.cristalab.com/blog/prototipos-bocetos-y-wireframes-con-balsamiq-mockups-c61456l>.
25. **Universidad de Harvard**. Museo de Zoología Comparada. [En línea] Universidad de Harvard. [Citado el: 21 de enero de 2012.] www.mcz.harvard.edu.
26. **Museo de Historia Natural de Londres**. Museo de Historia Natural de Londres. [En línea] octubre de 2005. [Citado el: 15 de enero de 2012.] <http://www.nhm.ac.uk>.
27. **Jardín Botánico Nacional de Cuba**. Jardín Botánico Nacional de Cuba. [En línea] Universidad de la habana, Cuba. [Citado el: 20 de enero de 2012.] <http://www.uh.cu/centros/jbn/>.
28. **CEMBIOS**. Instituto de Ecología y Sistemática. [En línea] Ministerio de Ciencia tecnología y Medio Ambiente. Cuba, junio de 2010. [Citado el: 12 de enero de 2012.] <http://www.ecosis.cu/>.
29. **Museología y Comunicación**. Museo Nacional de Historia Natural. [En línea] Museo Nacional de Historia Natural. [Citado el: 18 de enero de 2012.] <http://www.mnhnc.inf.cu/>.
30. **Miranda, Hilda Libia Martínez**. Herramienta Web para la autonomía en el aprendizaje. [En línea] 2007. [Citado el: 13 de febrero de 2012.]
31. **Escribano, Gerardo Fernández**. Introducción a Extreme Programming. [En línea] 2002. [Citado el: 18 de febrero de 2012.] <http://www.clubdevelopers.com/prog/articulos/xp/downloads/xp.pdf>.
32. **Drupal.org**. Sistemas de Gestión de Contenidos: "About". [En línea] [Citado el: 10 de febrero de 2012.] <http://drupal.org/features>.
33. **Tery Software**. (2008, noviembre 20). *Blog Tery*. Retrieved febrero 10, 2012, from



<http://blog.terysoftware.com/2008/11/construyendo-el-portal-de-tery-software-maqueteado>

34. **Koala Soft**. Desarrollamos. [En línea] mayo 12, 2012. [Citado: mayo 19, 2012.]
<http://www.koala-soft.com/Drupal>.

35. **Varios**. Drupal Org. [En línea] febrero 5, 2012. [Citado: mayo 3, 2012.]
<http://drupal.org/node/547518>. WCAG 2.0.

36. **Mühlrad, Daniel**. *Patrones de diseño*. 2008.

37. **Ivar Jacobson, Grady Booch, James Rumbaugh**. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.

38. **Pérez, Isaías Carrillo, González, Rodrigo Pérez y Martín, Aureliano David Rodríguez**. Metodología de Desarrollo del Software. [En línea] 15 de octubre de 2008. [Citado el: 12 de diciembre de 2011.] <http://www.google.com>.

39. **W. Ambler, Scott**. The Agile Unified Process (AUP) Home Page. *Ambyssoft*. [En línea] 2005. [Citado el: 18 de Febrero de 2010.] Disponible en:
<http://www.ambyssoft.com/unifiedprocess/agileUP.html>.

40. **Foros del Web**. MySQL 5 vs PostgreSQL 9 el verdadero test. [En Línea] 4 de enero de 2011. [Citado el: 10 de marzo de 2012] <http://www.forosdelweb.com/f21/mysql-5-vs-postgresql-9-verdadero-test-858912> CC: Licencia Creative Commons BY-NC-SA.

41. **Nazcasoft**. Pruebas Unitarias. *CalidadSoftware*. [En línea] [Citado el: 15 de Marzo de 2010.] Disponible en: http://www.calidadsoftware.com/testing/pruebas_unitarias1.php.

42. **Microsoft Corporation**. The Official Microsoft IIS Site. [En línea] Microsoft Corporation. [Citado: abril 10, 2012.] <http://www.iis.net/>.

43. **José H. Canós, Patricio Letelier, M^a Carmen Penadés**. *Todo Agil*. Valencia : DSIC - Universidad Politécnica de Valencia.

44. **varios**. Somos Libres. [En línea] SurNoticias. Perú, enero 6, 2011. [Citado: mayo 6, 2012.] <http://www.somoslibres.org/>.

45. **Suárez, Danais Rojas**. *Desarrollo del Sitio Web para los profesores guías de la Facultad 8*. Ciudad de La Habana : s.n., 2010. S.N.

46. **Serrano, Jorge**. Explicando Scrum a mi abuela. *MVP. Visual Developer*. [En línea] 2007. [Citado: febrero 5, 2012.] <http://geeks.ms/blogs/jorge/archive/2007/05/09/explicando-scrum-a-mi-abuela.aspx>. S.N.

47. **Varios**. GIMP. *GNU Image Manipulation Program*. [En línea] GIMP, 2001. [Citado: mayo 14, 2012.] <http://www.gimp.org/>. S.N.

48. **Instituto de ecología y Sitemática**. IES Zoología. [En línea] Ministerio de Ciencia Tecnología y Medio Ambiente, marzo 14, 2010. [Citado: febrero 16, 2012.] <http://www.ecosis.cu/zoologia.htm>. S.N.

49. **Bermúdez, Yasmany Antonio Pérez**. *Implementacion del componente entrada del modulo activo fijo tangible del Sistema Integral de Gestion Cedrux*. Ciudad de La Habana : s.n., 2010.



BIBLIOGRAFÍA

1. **Abraham.** MySQL vs PostgreSQL. ¿Cuándo emplear cada una de ellas? [Online] javaHispano, 2007. [Cited: octubre 25, 2011.] http://www.javahispano.org/contenidos/es/mysql_vs_postgresql_cuando_emplear_cada_una_de_ellas_11.
2. **Alfonso Lorenzo, Ana Belén Martínez Piñeiro, Esther Martínez Piñeiro.** Revista Electrónica de Investigación y Evaluación Educativa. *ALTIA Consultores*. [Online] Universidade de Santiago de Compostela (España), diciembre 21, 2004. [Cited: diciembre 8, 2011.] http://www.uv.es/RELIEVE/v10n2/RELIEVEv10n2_6.htm.
3. **Amble, Scott W.** The Agile Unified Process (AUP) Home Page. [Online] 2005. [Cited: enero 15, 2012.] Disponible en: <http://www.ambysoft.com/unifiedprocess/agileUP.html>.
4. **Beck, Kent.** Extreme Programming Explained: Embrace Change. Addison Wesley. [Online] septiembre 28, 2009. <http://www.extremeprogramming.org>.
5. **Bermúdez, Yasmany Antonio Pérez.** *Implementación del componente entrada del módulo activo fijo tangible del Sistema Integral de Gestión Cedrux*. Ciudad de La Habana : s.n., 2010. S.N.
6. **C. Castilla Plaza, Dr. Santodomingo Garachana.** Implicaciones de las tecnologías de la información en la gestión del sistema empresa. *Departamento de Economía Financiera y Contabilidad I*. Universidad Complutense de Madrid : s.n., 1999.
7. **Craig, Larman.** UML y patrones. s.l. : Prentice Hall, 2004. Vol. s.l. ISBN 978-8420534381.
8. **Dolores Lankenau, Cleopatra Garza.** Sistema Tecnológico de Monterrey. *Dirección de Desarrollo Académico*. [Online] 2012. [Cited: marzo 21, 2012.] <http://www.mty.itesm.mx/rectoria/dda/usols/creacion.htm>.
9. **Fersko-Weis, H.** The case for improved software design. Lotus. 1989. Vol. 5, 8. ISSN 8756-7334.
10. **Guevara, Humberto Rivero.** Trabajo de Diploma: Análisis, diseño e implementación del módulo Aprehensión del SIIPOL. Ciudad de la Habana : s.n., 2008. S.N.
11. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2000.
12. **J. Rumbaugh, I. Jacobson.** El Lenguaje Unificado de Modelado. Manual de Referencia. s.l. : WESLEY, 2000. ISBN 84-7829-037-08.
13. **Jiménez, Arianna Rodríguez.** *Proceso de evaluación de la documentación para el servicio de Pruebas Funcionales a nivel de Sistema que ofrece el Grupo de Calidad del centro FORTES*. Habana : s.n., 2011. S.N.
14. **Jorge Amado Soria Ramírez, Enrique José Altuna Castillo.** Trabajo de Diploma: Análisis, Diseño e Implementación de un Jurado Online. Ciudad de la Habana : s.n., 2008. S.N.
15. **José H. Canós, Patricio Letelier, María Carmen Penadés.** Metodologías Ágiles en el



Desarrollo de Software. Valencia : s.n. S.N.

16. **José H. Canós, Patricio Letelier, M^a Carmen Penadés.** Todo Agil. Valencia : DSIC - Universidad Politécnica de Valencia.

17. **Keating, Brian.** DatabaseSpecialists. [Online] [Cited: diciembre 10, 2011.] http://www.dbspecialists.com/files/presentations/mm_replication.html.

18. **Kniberg, Henrik.** Scrum y XP desde las trincheras. s.l. : C4Media, 2007. Vol. s.l. ISBN: 978-1-4303-2264-1.

19. **Martínez, Albania Padilla.** Sistema de datos homologados y estandarización de catálogos. 2003. Vol. Boletín de Política Informática, 6.

20. **Mestras, J. Pavón.** Patrones de Diseño Orientado a Objetos. [Online] Universidad Complutense de Madrid, 2004. <http://www.fdi.ucm.es/profesor/jpavon/poo/2.14PDOO.pdf>.

21. **Monge, Raúl.** Base de datos Distribuidas. Valparaiso : s.n., 2005. S.N.

22. **Morales, Freddy.** Slideshare. [Online] Instituto Nacional de Estadística e Informática. [Cited: diciembre 10, 2011.] <http://www.slideshare.net/FreddyMorales1/herramientas-case-11216676>.

23. **Mühlrad, Daniel.** Patrones de diseño. 2008.

24. **Nazcasoft.** Pruebas Unitarias. [Online] CalidadSoftware. [Cited: diciembre 9, 2011.] Disponible en: http://www.calidadsoftware.com/testing/pruebas_unitarias1.php.

25. **Patricio Letelier, M^a Carmen Penadés.** Metodologías ágiles para el desarrollo de software: Extreme Programming (XP). Valencia : Universidad Politécnica de Valencia. S:N.

26. **Pérez, Javier Eguiluz.** CCS 2.1. 2008.

27. **Pino, José Luis López.** lopezpino. *Servidores web más usados.* [Online] julio 30, 2010. [Cited: noviembre 22, 2011.] <http://lopezpino.es/2010/07/30/servidores-web-mas-usados>.

28. **Plummer, Paul.** Estimate Master. [Online] Coon Creek Software. [Cited: noviembre 11, 2011.] <http://www.cooncreeksoftware.com/index.html>.

29. **Pressman, Roger.** *Ingeniería del Software, un enfoque práctico 5ta Edición.* s.l. : Mcgraw-hill, 2001. ISBN: 8448132149.

30. **R. Figueroa, C. Solís, A. Cabrera.** Metodologías Tradicionales Vs. Metodologías Ágiles. [Online] 2008. [Cited: enero 24, 2012.] <http://adonisnet.files.wordpress.com/2008/06/articulo-metodologia-de-sw-formato>.

31. **Reynoso, Billy Carlos.** Metodos Agiles en Desarrollo de Software, Introducción a la Arquitectura de Software. [Online] [Cited: noviembre 11, 2011.] <http://www.microsoft.com/spanish/msdn/arquitectura>.

32. **Roberth Figueroa, Camilo Solis, Armando Cabrera.** Metodologías tradicionales vs metodologías ágiles. *Universidad Técnica Particular de Loja.* s.l. : Escuela de Ciencias en



Computación, 2007. S.N.

33. **Roelvis Coello Abijana, Kárel Adrián Silveira García.** Multimedia para Graduados de la Universidad de Ciencias Informáticas. Ciudad de la Habana : s.n., 2007. S.N.

34. **Serrano, Jorge.** Explicando Scrum a mi abuela. *MVP. Visual Developer.* [Online] 2007. [Cited: febrero 5, 2012.] <http://geeks.ms/blogs/jorge/archive/2007/05/09/explicando-scrum-a-mi-abuela.aspx>. S.N.

35. **Sommerville, Ian.** Ingeniería del Software. s.l. : Pearson Education, 2005. Vol. S.I.

36. **Suárez, Danais Rojas.** *Desarrollo del Sitio Web para los profesores guías de la Facultad 8.* Ciudad de La Habana : s.n., 2010. S.N.

37. **Torrens, R.** Arquitectura de la información. [Online] 1999. [Cited: diciembre 4, 2011.] <http://oai.saber.ula.ve/db/ssaber/Edocs/postgrados/computacion/ArquitecturadelaInformacion.pdf>.

38. **U.P.M, Departamento de O.E.I.-.** Diseño y optimización de Bases de Datos: Bases de Datos Distribuidas. 2010.

39. **Valdés, Damián Pérez.** Los diferentes lenguajes de programación para la web. [Online] [Cited: diciembre 12, 2011.] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web>.

40. **varios.** Somos Libres. [Online] SurNoticias. Perú, enero 6, 2011. [Cited: mayo 6, 2012.] <http://www.somoslibres.org/>.

41. **Varios.** GIMP. *GNU Image Manipulation Program.* [Online] GIMP, 2001. [Cited: mayo 14, 2012.] <http://www.gimp.org/>. S.N.

42. **Vázquez, J. A. Gallego.** Desarrollo Web con PHP y MySQL. Madrid : Grupo Anaya S.A., 2003. ISBN 84-415-1525-5.

43. **Vegas, Jesús.** Departamento de informática. [Online] Universidad de Valladolid. [Cited: noviembre 20, 2011.] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.