



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

**COMPARACIÓN DEL USO DE ADMINISTRADORES DE CONTENIDO Y
PLATAFORMAS CON SISTEMAS MANEJADORES DE BASES DE DATOS
RELACIONALES. CASO DE ESTUDIO: PLONE VS. POSTGRESQL**

T E S I S

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN CIENCIAS
(COMPUTACIÓN)**

P R E S E N T A:

JORGE ALBERTO CALDERÓN MÁRQUEZ

DIRECTOR DE TESIS: DR. JAVIER GARCÍA GARCÍA

México, D.F.

2012.



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A mi Familia y amigos.
Jorge

Resumen

La comparación de las tecnologías CMS y RDBMS es un tema que ha sido poco explorado, en este trabajo el caso de estudio se remite al desarrollo, uso y análisis de un sistema piloto en ambas plataformas: CMS (producto Plone) y RDBMS (producto PostgreSQL). Esto con el objetivo de tener un escenario uniforme que fuera óptimo para la realización y análisis de distintas pruebas, las cuales permitieron establecer un punto de comparación y con ello tener bases y criterios que ayudaron a tomar las decisiones en cada rubro para establecer las ventajas y desventajas de cada plataforma (CMS o RDBMS). Las pruebas realizadas en este trabajo se pueden dividir en dos bloques que involucrán cuestiones como funcionalidad y desempeño del sistema por un lado y por otro integridad y seguridad de la información.

El plan de pruebas a seguir se diseñó a partir de la idea de evitar sesgar los resultados buscando características ajenas a las tecnologías y por otro lado dándole prioridad a la manipulación y creación de la información, para conseguir esto también fue indispensable revisar modelos de calidad de desarrollo de sistemas y de manipulación de información.

Finalmente se puede mencionar que se realizaron las pruebas necesarias en un hardware dedicado, esto con el fin de cubrir los planes y tener una visión amplia de cómo se conducían ambas plataformas para el contexto dado. El análisis requirió la obtención de tablas y gráficas que ayudaron a la visualización del comportamiento de las plataformas en cada caso particular así como de manera general. Una vez terminado esto se acompañó dicho análisis de un sustento técnico, esto es, se prestó atención a diversos recursos de cada tecnología y del contexto en común a fin de englobar los resultados obtenidos.

Contenido

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Objetivos y contribución de este trabajo | 3 |
| 1.2 | Estado del arte | 4 |
| 1.3 | Organización del trabajo | 8 |
| 2 | Definiciones y Marco teórico | 11 |
| 2.1 | Definiciones | 11 |
| 2.2 | Marco teórico | 12 |
| 3 | Descripción del Sistema Caso Estudio | 15 |
| 4 | Estructura del Sistema en las Plataformas de Estudio | 23 |
| 5 | Puesta en Operación del Sistema Utilizando RDBMS | 29 |
| 5.1 | Herramientas y bibliotecas utilizadas | 29 |
| 5.2 | Estructura Física | 30 |
| 5.2.1 | Vistas | 31 |
| 5.2.2 | Modelo | 34 |
| 5.2.3 | Control | 35 |
| 5.2.4 | Base de Datos | 35 |
| 5.3 | Seguridad en el sistema | 35 |
| 5.4 | Desarrollo del sistema | 36 |
| 6 | Puesta en Operación del Sistema Utilizando CMS | 47 |
| 6.1 | Herramientas y bibliotecas utilizadas | 47 |
| 6.2 | Estructura Física | 48 |
| 6.2.1 | Contenidos | 49 |
| 6.3 | Seguridad en el sistema | 52 |
| 6.3.1 | Estado de un contenido | 52 |
| 6.3.2 | Usuarios, Grupos y Roles | 53 |

| | | |
|----------|--|------------|
| 6.3.3 | Seguridad de un contenido | 59 |
| 6.4 | Desarrollo del sistema | 60 |
| 7 | Plan de Pruebas y Datos | 63 |
| 7.1 | Funcionalidad y desempeño del sistema | 64 |
| 7.1.1 | Configuración | 66 |
| 7.1.2 | Plataforma RDBMS | 66 |
| 7.1.3 | Plataforma CMS | 74 |
| 7.2 | Integridad y seguridad de la información | 86 |
| 7.2.1 | Plataforma RDBMS | 88 |
| 7.2.2 | Plataforma CMS | 89 |
| 8 | Análisis de Resultados | 91 |
| 8.1 | Desempeño de la respuesta del sistema | 91 |
| 8.2 | Funcionalidad del sistema | 92 |
| 8.3 | Integridad y seguridad de la información | 100 |
| 9 | Conclusiones y Trabajos Futuros | 117 |
| 9.1 | Conclusiones | 117 |
| 9.2 | Trabajos futuros | 119 |
| | Bibliografía | 121 |
| A | | 125 |

Índice de Figuras

| | | |
|-----|--|----|
| 3.1 | Casos de uso general del sistema | 18 |
| 5.1 | Modelo Vista Controlador | 32 |
| 5.2 | Tabla Categorías, Tabla Académico, Tabla Actividades, Tabla Alumno, Tabla Aportaciones, Tabla Entidad Federativa | 40 |
| 5.3 | Tabla relación campo de conocimiento y Servicio Ofrecido, Tabla De- partamento, Tabla relación campo de conocimiento y Académico, Tabla Campo de Conocimiento, Tabla relación campo de conocimiento y Labo- ratorio, Tabla relación campo de conocimiento y Servicio a Resolver, Tabla Tipo de Documento | 41 |
| 5.4 | Tabla Entregable, Tabla Equipo de Laboratorio, Tabla Especificaciones de proyecto, Tabla Especificaciones de Servicio a Resolver, Tabla Especi- ficaciones de Servicio Ofrecido, Tabla Externos | 42 |
| 5.5 | Tabla Forma de Pago, Tabla Grupo de Trabajo, Tabla Relacion Grupo de Trabajo - Academicos y Alumnos, Tabla Instituto, Tabla Laboratorio, Tabla Relacion Laboratorio - Academicos y Alumnos | 43 |
| 5.6 | Tabla Pagos por Funciones, Tabla Usuario, Tabla Persona Moral, Tabla Proveedor, Tabla Relacion Proveedor - Departamento, Tabla Proyecto . | 44 |
| 5.7 | Tabla Persona Física, Tabla Título, Tabla de relacion Soluciones, Tabla Seguimiento de Actividades, Tabla Servicio a Resolver, Tabla Servicio Ofrecido | 45 |
| 6.1 | Estados de Publicación de contenidos | 53 |
| 6.2 | Privilegios locales referentes a cada contenido | 56 |
| 6.3 | Cambios en la seguridad de cada actividad por cada rol | 60 |
| 6.4 | Cambios en el uso de cada contenido via proxy por cada rol | 60 |
| 7.1 | Cota superior en proceso de consulta de proyectos | 70 |
| 7.2 | Cota superior en proceso de inserción de servicios ofrecidos | 79 |
| 8.1 | Usuario Final: Todas las pruebas en CMS | 93 |

| | | |
|------|--|-----|
| 8.2 | Usuario Final: Todas las pruebas en RDBMS | 94 |
| 8.3 | Usuario Final: SAR RDBMS vs CMS | 94 |
| 8.4 | Usuario Final: SO RDBMS vs CMS | 95 |
| 8.5 | Usuario Final: PROY RDBMS vs CMS | 95 |
| 8.6 | Usuario Final: Todos las consultas RDBMS vs CMS | 96 |
| 8.7 | Desarrollador: Todas las pruebas en CMS | 97 |
| 8.8 | Desarrollador: Todas las pruebas en RDBMS | 98 |
| 8.9 | Desarrollador: SAR RDBMS vs CMS | 98 |
| 8.10 | Desarrollador: SO RDBMS vs CMS | 99 |
| 8.11 | Desarrollador: PROY RDBMS vs CMS | 99 |
| 8.12 | Desarrollador: Todos las consultas RDBMS vs CMS | 100 |
| 8.13 | Todos los experimentos y sus combinaciones | 100 |
| | | |
| A.1 | Diagrama general de la base de datos | 126 |
| A.2 | Estructura del Sitio en Plone | 127 |
| A.3 | Estructura del Sitio en Plone | 128 |
| A.4 | Estructura del Sitio en Plone | 129 |
| A.5 | Estructura del Sitio en Plone | 130 |
| A.6 | Estructura del Sitio en Plone | 131 |
| A.7 | Estructura del Sitio en Plone | 132 |
| A.8 | Estructura del Sitio en Plone | 133 |
| A.9 | Estructura del Sitio en Plone | 134 |
| A.10 | Estructura del Sitio en Plone | 135 |
| A.11 | Estructura del Sitio en Plone | 136 |
| A.12 | Estructura del Sitio en Plone | 137 |
| A.13 | Identificación gráfica de las pruebas de funcionalidad | 138 |
| A.14 | Estrcutura Física de Plone | 139 |

Índice de tablas

| | | |
|------|--|----|
| 2.1 | Comparativa de herramientas | 14 |
| 4.1 | Calidad de datos iso/eic 25012 necesarios en este trabajo | 27 |
| 5.1 | Descripción de los archivos Java Script | 32 |
| 5.2 | Descripción de los archivos CSS | 33 |
| 5.3 | Descripción de las vistas | 34 |
| 5.4 | Matriz de Roles y accesos | 37 |
| 6.1 | Tabla de contenidos mas usados en Plone | 49 |
| 6.2 | Descripción de grupos y sus roles | 58 |
| 7.1 | Características de las imágenes del reporte PROY - RDBMS | 67 |
| 7.2 | Características de las imágenes del reporte SAR - RDBMS | 68 |
| 7.3 | Características de las imágenes del reporte SO - RDBMS | 69 |
| 7.4 | Tiempos (ms) para la inserción masiva en la plataforma RDBMS | 69 |
| 7.5 | Tiempos (ms) para todas las pruebas en la plataforma RDBMS. | 70 |
| 7.6 | Tiempos (ms) para todas las pruebas en la plataforma RDBMS. | 71 |
| 7.7 | Características de las imágenes del reporte PROY - CMS | 76 |
| 7.8 | Características de las imágenes del reporte SAR - CMS | 77 |
| 7.9 | Características de las imágenes del reporte SO - CMS | 78 |
| 7.10 | Tiempos (ms) para la inserción masiva en la plataforma CMS | 79 |
| 7.11 | Tiempos (ms) para todas las pruebas en la plataforma CMS. | 80 |
| 7.12 | Tiempos (ms) para todas las pruebas en la plataforma CMS. | 80 |
| 7.13 | Matriz 1 de Pruebas de integridad y seguridad para proveedores | 88 |
| 7.14 | Matriz 1 de Pruebas de integridad y seguridad para clientes | 89 |
| 7.15 | Matriz 2 de Pruebas de integridad y seguridad para proveedores | 89 |
| 7.16 | Matriz 2 de Pruebas de integridad y seguridad para clientes | 90 |
| 7.17 | Usuarios utilizados en las pruebas de la plataforma RDBMS | 90 |
| 7.18 | Usuarios utilizados en las pruebas de la plataforma CMS | 90 |

| | | |
|------|---|-----|
| 8.1 | Listado de Observaciones | 101 |
| 8.2 | Resultados para el usuario usuAdmon1 - Administrador 1 | 103 |
| 8.3 | Resultados para el usuario usuAca1 - Académico 1 | 103 |
| 8.4 | Resultados para el usuario usuLab1 - Laboratorio 1 | 104 |
| 8.5 | Resultados para el usuario usuGpo1 - Grupo de trabajo 1 | 104 |
| 8.6 | Resultados para el usuario usuPF1 - Persona Física 1 | 105 |
| 8.7 | Resultados para el usuario usuPM1 - Persona Moral 1 | 106 |
| 8.8 | Resultados para el usuario usuAdmon2 - Administrador 2 | 106 |
| 8.9 | Resultados para el usuario usuAca2 - Académico 2 | 107 |
| 8.10 | Resultados para el usuario usuLab2 - Laboratorio 2 | 108 |
| 8.11 | Resultados para el usuario usuGpo2 - Grupo de trabajo 2 | 108 |
| 8.12 | Resultados para el usuario usuPF2 - Persona Física 2 | 108 |
| 8.13 | Resultados para el usuario usuPM2 - Persona Moral 2 | 109 |
| 8.14 | Resultados de las pruebas hechas en CMS y RDBMS | 116 |

Capítulo 1

Introducción

Hoy en día las soluciones que suelen brindar los profesionales del desarrollo web son demandadas en gran cantidad en la sociedad y estos cuentan con un gran abanico de posibilidades para dar respuesta a las peticiones que las personas esperan de los diversos sistemas.

Muchas veces los mismos problemas tienen inherentes características y especificaciones que pueden llegar a ser suficientes para poder brindar a los computólogos los fundamentos necesarios para definir y apoyarse en una tecnología o conjunto de ellas para las soluciones que generen.

En la web existen muchos recursos y formas de obtenerlos; poder crear información, la manera en como se utiliza y comparte es tal vez la mayor necesidad que hoy en día es soportada en internet. Para esto se puede utilizar sistemas como bitácoras personales (blogs), sistemas de encuentro, sistemas para empresas privadas o de gobierno, portales dedicados al entretenimiento, etc. Si bien, cada uno tiene sus rasgos muy específicos, se puede decir que todos ellos en los que la información es creada o compartida entre usuarios, son sistemas que de alguna forma almacenan, protegen y solucionan demandas de información.

Estos sistemas en su gran mayoría son soportados por tecnologías que cuentan con una o varias capas de subsistemas (servidores de aplicaciones, bases de datos, contenedores de información, capas de seguridad y control, capas de visualización, etc.) encargadas de la gestión de control, mantenimiento y el flujo de dicha información.

Ahora bien, un profesional de la materia, una vez que ha entendido el contexto del problema que tiene que resolver, debe de considerar más de una opción como plataforma para trabajar sobre ella, conociendo cuales serían las ventajas y desventajas que tendrá

al afrontar el problema, para después escoger una plataforma a utilizar bajo el contexto dado.

En esta tesis se presentan dos alternativas de plataformas para el desarrollo de sistemas; una de estas plataformas es la formada por los sistemas manejadores de bases de datos relacionales (o bien, sistemas administradores de bases de datos relacionales: RDBMS) y lenguajes de programación como Java, PHP, C o algún otro que pueda explotar la información contenida en los RDBMS. Los RDBMS llevan mucho tiempo siendo utilizados, y de igual forma se han realizado muchos estudios, encontrándolos como la opción más sólida, robusta, rápida y segura que hasta el momento se conoce.

Los RDBMS están regidos por una vasta teoría bien definida y aunque existen varios productos que soportan esta tecnología (ORACLE, Postgres, MySQL, MSSQL, Teradata, Sybase, Informix) y cada uno de estos emplea diversas técnicas para realizar tareas en específico, se puede hablar de ellos como un conjunto confiable, de fácil uso si se tiene el aspecto teórico y técnico, y robusto, ya que son sistemas muy complejos que realizan tareas claves para mantener en un estado consistente, confiable, y disponible la totalidad de la información con la que se disponga.

La segunda plataforma propuesta en este trabajo, es otra forma de desarrollar sistemas, misma que ha cobrado importancia en los últimos años. Esta alternativa está representada por los sistemas administradores de contenidos o sistemas gestores de contenidos (CMS). Dichos sistemas han cobrado auge ya que son tecnologías que ayudan a la creación de sistemas con el uso de herramientas que ocultan la programación de ciertos módulos y de esta forma facilitan la construcción de sistemas a las personas que no tienen muchos conocimientos sobre lenguajes de programación o bases de datos. Para las personas que sí cuentan con este conocimiento, también resultan de gran ayuda ya que se puede avanzar de manera más ágil y aun se pueden abordar puntos en específico dedicándoles más tiempo y explorando dentro de la herramienta, realizando los ajustes que así consideren necesarios.

Los CMS están contruidos en varias plataformas para diversos lenguajes de programación; algunas soportan bases de datos relacionales, bases de datos orientadas a objetos, bases de datos estructuras o semi estructuradas; algunas versiones también varían en los servidores de aplicación que utilizan; es decir, existe una amplia gama de posibilidades para construir los CMS, por ello han crecido rápidamente y si ya se ha escogido la opción de utilizar un CMS como plataforma de desarrollo, ahora también se tendrá que dedicar algún tiempo y esfuerzo en escoger un adecuado CMS de entre

todas las posibilidades que actualmente existen.

1.1 Objetivos y contribución de este trabajo

En esta sección se podrán encontrar los objetivos y contribuciones que este trabajo tiene como prioritarios

Objetivos

El objetivo primordial al realizar este trabajo es medir y analizar el impacto global de desarrollar y usar un sistema productivo con el apoyo de un sistema manejador de bases de datos, servidores y lenguajes de programación que ayuden a procesar la información y tener un portal adecuado para usuarios finales; y por otro lado construirlo y usarlo bajo el contexto de los Sistemas Administradores de Contenidos, con sus servidores y lenguajes nativos.

Esta medición y análisis se realizó a través del desarrollo de un sistema piloto productivo, “Sistema de Vinculación” en ambos ambientes, es decir, se trató desde el punto de vista de un programador que realiza el sistema tanto en una plataforma como en la otra; y también desde el punto de vista de un usuario final al utilizar ambos sistemas. Una vez finalizado el proceso de construcción también se compararon ambos sistemas a través de la medición de ciertos aspectos que se detallan mas adelante.

Esta comparación franca entre las dos tecnologías permitió establecer puntos cualitativos y cuantitativos en los cuales una tecnología se destaca sobre la otra para así sumar resultados y establecer en términos generales qué paradigma es la mejor opción bajo un contexto dado. De la misma forma, cuando los datos arrojados no fueron lo suficientemente claros para poder definir cierta ventaja en el uso de una tecnología sobre la otra, también se dieron a conocer sus similitudes y se expusieron sus características.

La comparación se hace utilizando un conjunto de criterios como lo son la respuesta del sistema a solicitudes, la facilidad al introducir datos y la forma como se obtienen, la privacidad de la información, entre otras. Es decir, esta comparación abarca características medibles entre estas dos tecnologías, esto se llevo a cabo siguiendo modelos de calidad que se apegan a desarrollos de sistemas y desarrollo y mantenimiento

de bases de datos.

Contribuciones

La tesis tiene como principal aportación comparar los sistemas administradores de contenido contra las plataformas con sistemas manejadores de bases de datos relacionales a través de las tecnologías Plone y PostgreSQL respectivamente. Lo anterior es posible a través de un ejemplo controlado sobre la creación de un sistema en ambas tecnologías y observar a través de la experimentación y uso de los mismos cuales características o situaciones se resuelven mejor en un contexto o en otro. La medición de las pruebas arrojó como es la respuesta de ambas plataformas y en base en ello se hicieron inferencias que llevaron a fijar una postura para cada criterio.

Este trabajo tiene como base el establecer un cotejo entre las tecnologías para soportar la idea de usar una sobre la otra, esto en condiciones similares a las presentadas en cuanto a los productos utilizados, los productos actuales de desarrollo de software y el contexto del sistema mismo.

Este estudio es de relevancia con respecto al desarrollo de sistemas web dedicados a la explotación de la información. Las personas interesadas con este trabajo, encontrarán en este estudio un ejemplo real que es pensado para contemplar las fortalezas y debilidades de dos soluciones. Así mismo, se podrá tomar en consideración para posteriores análisis y que tal vez engloben otros aspectos, para así tener cada vez mayores fundamentos para saber elegir la tecnología que mejor represente los intereses que cada quien busca.

En la tesis se llevó a cabo la comparación en base a las siguientes características: ventajas y desventajas de desarrollo, facilidad de uso de herramientas, desempeño de la respuesta del sistema, funcionalidad del sistema, integridad y seguridad de la información. Cabe mencionar que dichos elementos son los únicos que se tomarán en cuenta para presentar los resultados y por ende para determinar la comparación. Como trabajo futuro se podrán considerar otros criterios como por ejemplo manejo de concurrencia, distribución y/o paralelismo y mantenibilidad.

1.2 Estado del arte

En internet todos los sistemas transaccionales, blogs, recursos multimedia y otros contenidos que se comparten en la web de manera continua, han tenido una mayor flexi-

bilidad de mantenimiento, facilidad de uso, portabilidad entre otras características en el transcurso de los años, esto tanto para su creación, como para su manejo, control y explotación.

Actualmente existen muchos trabajos relacionados con los sistemas web que son soportados con sistemas administradores de bases de datos relacionales. (RDBMS – Relational Database Management System) ya que es una teoría sólida que se ha venido desarrollando desde la década de 1970. A través de varios estudios se han demostrado su utilidad, eficiencia y confiabilidad en el tratamiento de información.

Por otro lado, el paradigma de los sistemas administradores de contenidos es más reciente y aunque ha cobrado fuerza por su rápido crecimiento; los estudios realizados con este son menores en número y profundidad comparado con el paradigma de los RDBMS.

En [23] los autores hacen un comparativo entre diversos sistemas administradores de contenidos (CMS – Content Management System) y mencionan cuales serían las mejores opciones dependiendo del contexto, convencidos de su eficiencia y poder de desarrollo de los CMS, mencionan que los lenguajes de código abierto CMS son las mejores opciones para crear y administrar sitios web sofisticados. Afirman que se puede crear sitios que resuelven de manera puntual las metas del negocio y que mantengan el sitio actualizado fácilmente al ser sistemas que proporcionan un control total de de cada aspecto del sitio. En esta tesis se plantea como opción de construcción del sistema un CMS de código abierto como lo es Plone, y se toma de inicio sin sesgo, es decir, pensando en que en realidad puede ser la mejor opción para las necesidades del sistema a desarrollar.

La elección de Plone no fue tomada al azar. Plone es un sistema de gestión de contenidos hecho en Python y soportado por Zope. En [7] se puede revisar que dentro de una gama de CMS, Plone es uno de los más flexibles, y que sus características lo colocan como uno de los preferidos por la comunidad de desarrolladores dentro de los CMS. Siendo Plone utilizado en alrededor de 57 países y existiendo más de 300 proveedores de este servicio [22].

Con Plone las funciones multimedia CMS se pueden realizar de manera sencilla utilizando productos hechos por la comunidad o bien, extendiendo las funcionalidades de los productos que vienen dentro del sistema. Imágenes, audio, video y más opciones son adheribles a carpetas de contenidos similares, [16]. En la tesis se asume la postura de trabajar con los productos nativos de Plone, y tratar de explotarlos extendiéndolos

según el contexto. Los productos de otros usuarios, serán utilizados en la menor medida posible para poder dejar el sistema con poca ayuda de frameworks y/o productos tal y como se realizó con el sistema hecho con RDBMS.

Un argumento sobre utilizar la plataforma Plone [19] es que ofrece una interfaz gráfica y por tanto el uso de la misma elimina el uso de conocimientos de programación, cubriendo las necesidades de un CMS. Durante el desarrollo de la tesis, se realizaron los cambios que son posibles con la interfaz gráfica, dejando como último recurso el uso de la programación bajo esta plataforma. Para así poder validar lo fundamentado en esta bibliografía

Para el desarrollo de los sistemas tanto con el CMS como con el RDBMS se usarán lenguajes de programación muy utilizados, uno Python y otro ya más tradicional Java, respectivamente. Al tratarse diferentes lenguajes, se pensaría que brindan escenarios completamente distintos, es decir, distintos contextos, respuestas diferentes a las mismas entradas. Sin embargo, en el artículo [26] se tienen resultados sobre un experimento de comparación entre Python y otros lenguajes entre ellos Java y resultan no ser tan distintos entre sí. Estos resultados son un trabajo previo que resulta útil al momento de realizar el análisis al dar entradas a ambos sistemas de esta tesis y comprobar sus respuestas. Otro trabajo que resalta, al hacer de dos puntos aparentemente opuestos una solución integral es el encontrado en [3], ya que en este artículo se plantea la posibilidad de mezclar o unir bases de datos orientadas a objetos con bases de datos relacionales, lo cual brinda un escenario donde se obtenga lo mejor de ambos paradigmas. En esta tesis si bien se confrontan ambas formas de ver las bases de datos, es importante tener en cuenta que existe una posibilidad que pueda ser alterna a las dos utilizadas, y sería la combinación de las mismas; si bien no es objeto de estudio de este trabajo, en ciertos contextos pudiera ser un planteamiento interesante.

Con respecto a los servidores, existen estudios [12] que comparan los desarrollos de aplicación web utilizando servidores Zope y aquellos sistemas soportados por CGI y Java. En este caso muestra ciertas ventajas de Zope sobre la otra tecnología. Aunque en esta tesis los servidores serán distintos y se deben considerar en ciertas métricas, no es objetivo de estudio comparar beneficios de cómo está construido un servidor o el otro o su respuesta ante situaciones adversas, sin embargo, es interesante observar su contribución.

Muchos sitios web ya se han realizado en este tipo de tecnologías (CMS) tanto para su estudio como para realmente soportar sus propias demandas [24],[14]. Y mues-

tran que existen tanto grandes ventajas como limitaciones en el uso de los CMS. Si bien, en este experimento se tiene la oportunidad de contemplar un ambiente controlado se necesita valorar estos estudios previos para ver el comportamiento de los resultados que arroje este análisis y ver si coinciden de manera general en algunos puntos ya estudiados.

Un análisis realizado en [27] comparando un sistema en CMS y otro hecho con PHP y Mysql resulta en una decisión difícil de tomar, es decir, utilizando CMS es considerable el tiempo y conocimiento ahorrado en bases técnicas, sin embargo, la respuesta que da el artículo no es sobre las ventajas de usar una tecnología sobre la otra. Para el estudio que se realizó en esta tesis, existen situaciones en las que los resultados no arrojan de manera contundente una ventaja de una tecnología sobre la otra, sin embargo se trata de manera general de dar una visión que permita tomar una postura.

El punto central de este estudio es ver como la información es almacenada, explotada, asegurada y administrada en dos conjuntos de tecnologías distintas. Los autores del artículo [20] hacen unos ajustes a un sitio soportado en Zope para agregar un RDBMS (en el caso del artículo Postgres), ya que argumentan que las bases de datos relacionales son flexibles, seguras y rápidas. Usando bases de datos se puede almacenar y recuperar datos sin tener capa de persistencia en la aplicación web, esto implica menos errores de codificación, mayor velocidad, etc. También mencionan sobre cuando los datos son más adecuados para un modelo relacional de base de datos que uno de objeto como en el caso de ZODB. Incluso dentro de Zope se maneja un estándar que define métodos ZSQL, para realizar la transformación antes citada. Para el trabajo de esta tesis, se compararán justo el desempeño del CMS utilizando la ZODB y todas las capas que le dan utilidad a esta herramienta y por otro lado el desempeño de llevarlo a cabo con RDBMS (Postgres). Sin embargo, este estudio previo refuerza la idea de tener en sitios webs transaccionales la ayuda de un RDBMS.

Existe un método relativamente nuevo llamado WEM, que fue propuesto en [28], el cual es un método diseñado para el desarrollo de aplicaciones web basadas en CMS, esto a través de la identificación de sus características y combinándolas con métodos probados como UML Web Engineering (UWE) y el proceso Unificado. Para la tesis el análisis y la comparación giran en torno a la programación dejando fuera la etapa de análisis y diseño, las cuales fueron parcialmente abordadas en el curso de Tecnología Orientada a Objetos, ya que se trataba de una versión muy reducida comparada con el sistema final. Sin embargo, para estudios posteriores sería interesante probar esta metodología y comparar resultados, ya que sin duda pueden aportar más conocimiento global del paradigma.

1.3 Organización del trabajo

La tesis a manera de resumen, cuenta con la siguiente estructura:

Capítulo 1 Introducción.

Aquí viene una breve *motivación* que sirve para sentar los intereses principales del desarrollo de esta tesis. También cuenta con un apartado de *objetivos y contribuciones*, el cual detalla de manera más puntual que ofrece este trabajo y cuáles son los puntos a remarcar durante el desarrollo, así como en la finalización de esta tesis.

Después como último tema de la introducción se desarrolla el *estado de arte* de los CMS y las comparaciones o estudios hechos entre RDBMS y CMS.

Capítulo 2 Definiciones y Marco Teórico.

Aquí se separa en dos partes, una dedicada a las *Definiciones* y otra de *Marco teórico*, en este último apartado se explica lo que es un sistema de bases de datos relacionales y sus características principales, también se presenta que es un sistema manejador de contenidos y sus características principales, además se justifica el uso de los productos utilizados de estas tecnologías: Postgres y Plone, respectivamente.

Capítulo 3 Descripción del Sistema Caso Estudio.

Este capítulo explica el problema a desarrollar, es decir, de que trata el sistema, cuáles serán los productos utilizados tanto para la versión que utiliza el RDBMS como la versión del CMS. Aquí se esboza de manera general que aspectos serán tomados en cuenta para realizar la comparación y el análisis de las pruebas.

Capítulo 4 Estructura del sistema en ambas plataformas.

El objetivo principal de este capítulo es mostrar las similitudes y diferencias que existen entre los paradigmas y los cuales se ven reflejados en los capítulos puesta en operación del sistema utilizando RDBMS [5] y puesta en operación del sistema utilizando CMS [6]. Es decir, este apartado está dedicado a dar una introducción y estructura de lo que se detallará mas adelante.

Capítulo 5 Puesta en operación del sistema utilizando RDBMS.

En este apartado se va a tener la estructura general del sistema hecho en la versión con RDBMS, se dan detalles de la construcción, se habla de la complejidad de su desarrollo, de las ventajas y desventajas que se tuvieron durante el desarrollo y uso del mismo.

Capítulo 6 Puesta en operación del sistema utilizando CMS.

Este capítulo es muy parecido a estructura en cuanto al cuarto, es decir, aquí también se dan detalles de construcción, se tocan puntos sobre su complejidad de desarrollo, ventajas y desventajas pero evidentemente desde la versión utilizada con CMS.

Capítulo 7 Plan de pruebas y datos.

Para este modulo se especifican ya de manera puntual cómo y qué se va a probar de ambos sistemas, así como los detalles del proceso de pruebas y herramientas a utilizar, si es el caso.

Capítulo 8 Análisis de resultados.

En este apartado cubre los datos arrojados en las pruebas y se les da el tratamiento adecuado para hacer de ellos la información adecuada para tomar posturas en cuanto a los mismos.

Capítulo 9 Conclusiones.

Para este capítulo se recogen la información del capítulo Análisis de resultados [8] y se lleva a cabo las aseveraciones y conclusiones finales del trabajo, así como de posibles trabajos futuros que se pudieran llevar a cabo, tomando como base este primer estudio.

Capítulo Bibliografía.

Aquí se enlistan los trabajos que soportan la base de conocimientos previos a este estudio.

Capítulo A Apéndice A.

Información extra y relevante al trabajo se podrá encontrar en este apartado.

Capítulo 2

Definiciones y Marco teórico

2.1 Definiciones

En este capítulo se realiza un desarrollo de los aspectos técnicos que son necesarios para llevar a cabo la construcción de los sistemas utilizando CMS y RDBMS. Primeramente, se tienen una serie de definiciones:

Datos: Los datos son pequeños fragmentos de la información contenida en una computadora (números, palabras, imágenes, sonidos). [1]

Información: La información es el texto, sonido, imagen y videos que tiene significado específico en un contexto y tiene la característica de ser compartida por grupos de personas que tengan intereses en común.[1]

Sistema: Conjunto de reglas o principios sobre una materia racionalmente enlazados entre sí. [1]

Sistemas manejadores de Bases de datos (DBMS): Un sistema manejador de bases de datos (DBMS) consiste de una colección de datos y programas para acceder a estos datos. El principal objetivo de los DBMS es proveer un ambiente conveniente y eficiente para las personas que necesitan almacenar y recuperar información. Los sistemas de bases de datos son diseñados para almacenar grandes volúmenes de información. La administración de estos datos requiere de estructuras de almacenamiento y mecanismos de manipulación de la información. En adición, los sistemas de bases de datos, proveen debido a la seguridad de la información almacenada, mecanismos de autenticación, soluciones para la concurrencia y por su puesto una vista abstracta que esconde a los usuarios finales el cómo se almacena y se le da mantenimiento a los datos.

[15]

Sistemas manejadores de Bases de datos Relacionales (RDBMS): Son sistemas manejadores de bases de datos. El modelo relacional está basado en un conjunto de tablas. Los usuarios de los sistemas de bases de datos, pueden consultar estas tablas, insertar nuevas tuplas, borrar tuplas, y modificarlas. Existen varios lenguajes para expresar estas operaciones. [15]

Contenidos: Contenido es un compromiso entre la utilidad de los datos y la riqueza de la información. El contenido es la información rica que envuelve datos simples. Los datos acerca de la información (metadatos) es una versión simplificada del contexto y el significado de la información. [13]

Sistemas manejadores de contenidos: En [21] se definen los CMS como sistemas que permiten la creación, manejo, distribución y búsqueda de contenido. Cubren todo el ciclo de vida de las páginas de un sitio Web, desde la creación y publicación del contenido hasta que es quitado o almacenado. Deben permitir que este contenido sea de gran variedad de formatos y tipos, sin limitarse sólo a HTML, XML, videos, audio, imágenes y documentos. Otros trabajos que ayudan a esta concepto son [8]

2.2 Marco teórico

Un CMS puede tener tanto páginas estáticas, estrictas en HTML, como dinámicas, utilizando alguna tecnología como JSP, ASP, Python o PHP. Pero sin importar si el contenido es una página estática o es mostrada utilizando páginas dinámicas, para el sistema sigue siendo contenido con diferencias menores, como la forma en que se va a mostrar. De hecho, la mayoría de los sitios Web grandes utilizan tanto páginas estáticas como dinámicas.

La creación del contenido debe estar estrictamente separada de la presentación que se le va a dar. Así el creador del contenido no tiene que tener conocimiento técnico para diseñar o crear una página, y el diseñador no tiene que estar creando cada página que se quiere publicar. De esta forma se reduce la complejidad de las tareas realizadas por los actores envueltos en el proceso de publicación de contenido, además de la disminución de la dificultad de comunicación que pudiera surgir entre estos.

Ahora bien de entre los sistemas manejadores de bases de datos que existen en la actualidad se consideraron estos como búsqueda inicial: Berkeley DB, Firebird, PostgreSQL, MySQL, Sybase, Apache Derby, MSQL, Qddb, Ozone, Exodus y YOODA.

Ozone, Exodus y YOODA fueron descartados al ser DBMS orientado a objetos. Ahora bien, de los restantes había que elegir, uno que fuera un sistema confiable, que contara con flexibilidad, opciones distintas de configuración, que actualmente fuera utilizada por empresas, academias, e individuos interesados en el tema. Con estos requisitos el conjunto de posibilidades se redujo a PostgreSQL, MySQL y Sybase.

Además de estos requisitos, dos puntos importantes orillaron a la elección final, uno es que fuera posible la conexión con algún o varios CMS para almacenar datos, esto en caso de necesitarse o bien, explorar más su funcionamiento y finalmente que al tratarse de un sistema que es productivo, tendría que adaptarse mejor al ambiente de producción de la facultad de Ciencias. Por todo esto la elección fue utilizar el RDBMS PostgreSQL.

Para acompañar a este motor de base de datos, se eligió un lenguaje de programación que fuera flexible, y de igual forma que el RDBMS fuera muy utilizado en la actualidad y tuviera grandes posibilidades de adaptación con el ambiente productivo, de entre lenguajes como PHP, C y Java, la elección fue Java. En cuanto al servidor de aplicaciones se tomó como opción Apache Tomcat al ser perfectamente compatible con Java. Otros lenguajes de apoyo como CSS, HTML, y Ajax fueron utilizados para completar la primera plataforma.

Ahora bien detallando la plataforma de los CMS, se tuvo que tomar la decisión bajo la misma perspectiva que la otra plataforma, es decir, que fuera un administrador de contenidos muy utilizado por la comunidad, con tecnología reciente, que aportara gran flexibilidad, seguridad y portabilidad, y de ser posible que tuviera presencia en México. En [7] se pudo comparar varias características de muchos CMS, resultando un conjunto de varios CMS a elegir, pero debido al estudio previo que se ha realizado sobre uno en particular aquí en México, se tomó la decisión de trabajar sobre Plone. Parte de este estudio previo se puede encontrar en trabajos como lo son [30], [31] y [21]. Este administrador de contenido tiene como soporte un servidor de aplicaciones Zope, esta codificado en Python como lenguaje de programación y tiene como base de datos ZODB la cual es una base de datos orientada a objetos. Además cuenta con lenguajes de apoyo como CSS, Ajax y XHTML. Presencia internacional de Plone dentro de la academia existen muchos ejemplos, uno de ellos se puede consultar en [29].

Para finalizar este capítulo se presenta una tabla comparativa de ambas plataformas con las elecciones realizadas en cuanto a los productos utilizados que conforman ambas tecnologías.

Tabla 2.1: Comparativa de herramientas

| Recurso/Plataforma | RDBMS | CMS |
|---------------------------|---------------|------------|
| Base de datos | PostgreSQL | ZODB |
| Lenguaje de programación | Java | Python |
| Servidor de Aplicaciones | Apache tomcat | ZOPE |
| Lenguaje de apoyo | CSS | CSS |
| Lenguaje de apoyo | HTML | XHTML |
| Lenguaje de apoyo | Ajax | Ajax |

Capítulo 3

Descripción del Sistema Caso Estudio

Descripción

El sistema piloto que sirvió para este caso de estudio surge con una funcionalidad doble, es decir, por un lado pretende brindar un contexto serio y real para tener una base en la cual realizar la experimentación de esta tesis, y por otro lado tiene la virtud de automatizar un proceso real en la Secretaría de Vinculación de la Facultad de Ciencias.

El sistema pensado como un sitio donde almacenar la información de la Secretaría de Vinculación, cumple con cuestiones como el registro de usuarios, definición de roles o grupos de usuarios, manejo de información y sustento de proyectos, más adelante se describe a detalle su funcionalidad.

Este proyecto ayuda a las personas externas o clientes a registrarse para después presentar a través del sitio los problemas y/o proyectos de su interés que sean viables a resolver dentro de la facultad, o bien, también pueden solicitar servicios ya existentes dentro del sitio. Parte de la funcionalidad que tienen los clientes es encontrar y elegir con facilidad, de entre una lista de propuestas, algún proveedor (Laboratorio, grupo de trabajo o académico) que les presente la que ellos consideren la mejor solución a su problema, a través del apoyo de la secretaría. Por otro lado, los proveedores también al inscribirse en el portal, podrán tener la oportunidad de ofrecer sus servicios y buscar a través de la secretaría, proyectos en los cuales participar y de esta manera crear vínculos que permitan colocar a los integrantes de la Facultad como prestadores activos de servicios a la industria y/o el gobierno.

Objetivos del sistema

Un cliente tiene un problema o proyecto y está buscando quien tenga la mejor solución y lo propone a través de un sistema web.

Proveedores de la Facultad de Ciencias, esto es académicos, alumnos pertenecientes a un Laboratorio o grupo de trabajo revisan estos problemas y pueden proponerse para solucionarlo. El cliente recibe de la secretaría las soluciones propuestas, escoge una o varias y se inician una negociación para trabajar conjuntamente.

Por otra parte los proveedores de la Facultad de Ciencias, pueden publicar servicios que pueden proporcionar indicando sus capacidades y habilidades a fin de que algún cliente quiera ponerse en contacto para establecer un convenio.

Características principales

Se espera que el sistema cumpla con las siguientes características:

- La persona o empresa solicitante, es decir, aquel cliente que tiene un problema o proyecto, podrá registrarse en el sitio. El registro incluye sus datos como: nombre, correo electrónico, teléfono, entre otros.

- Los clientes inscritos podrán registrar su problema. Los datos que se requieren para el registro del problema son: título del problema, categoría, descripción, entre otros.

- Se podrá modificar el problema o proyecto para aclararlo más, o inclusive borrarlo si no recibe propuestas de solución.

- Por otro lado, los proveedores puede entrar a este sitio y registrarse. El registro incluye sus datos como: nombre, correo electrónico, teléfono, si es persona física o de algún laboratorio, etc.

- Los proveedores pueden ver los problemas planteados y si alguno le interesa, puede proponer sus aportes. Los datos que se requieren para el registro de una solución

son: solución propuesta, cómo contactarse, entre otros. Cada proveedor podrá eliminar o modificar sus propios aportes.

- Cualquier persona podrá consultar los problemas, servicios o proyectos que estén registrados en el sistema y registrarse.

Sobre requerimientos no funcionales se van a cuidar los siguientes aspectos.

Desempeño

Garantizar la confiabilidad, la seguridad y el desempeño del sistema a los diferentes usuarios. Para que la información almacenada, consultada y modificada por otros usuarios no afecte el tiempo de respuesta y la certeza de la información consultada.

El sistema debe tener la capacidad de dar respuesta al acceso de todos los usuarios, y en un tiempo de respuesta aceptable

Disponibilidad

Debe estar disponible las 24 horas del día, de preferencia, ya que a veces los usuarios no tienen un tiempo específico para trabajar.

Escalabilidad

El sistema debe ser construido de manera que permita en un futuro el desarrollo de nuevas funcionalidades, y que estas al ser incorporadas afecten al sistema de la menor manera posible

Facilidad de Uso

El sistema debe ser de fácil uso, contando con interfaces entendibles que permitan un sencillo manejo por parte de los usuarios.

Seguridad

El acceso a algunos módulos del Sistema deben estar restringidos por el uso de nombres de usuario y contraseñas asignadas a cada uno de los usuarios, los cuales estarán clasificados en varios tipos de usuarios (o roles) con acceso a las opciones de trabajo definidas para cada rol.

Validación de Información

El sistema debe validar automáticamente la información contenida en los formularios y/o paginas. En el proceso de validación de la información, se deben tener en

cuenta aspectos tales como validación de campos, longitud de caracteres permitida por campo, manejo de tipos de datos, etc.

El diagrama de casos de uso general es (donde ABCC es altas, bajas, cambios y consultas, respectivamente):

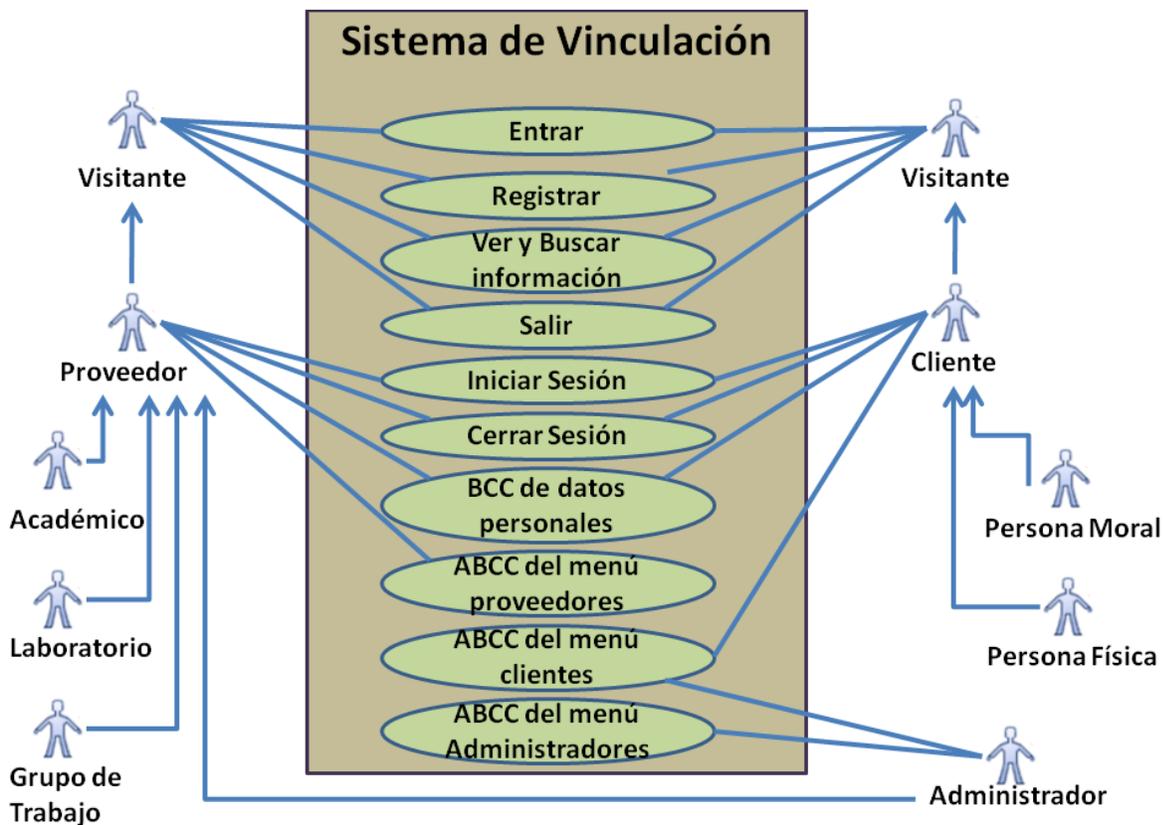


Figura 3.1: Casos de uso general del sistema

Características de las pruebas y métricas.

Teniendo en consideración los requerimientos básicos funcionales y no funcionales, así como la definición esencial del sistema; lo que debe seguir en esta descripción del problema es preguntarse acerca de la información concerniente a la necesidad de identificar los atributos, métodos y/o procesos fundamentales de la comparación a realizar de ambos sistemas, que son pilares en los resultados de la comparación entre ambos

sistemas, como de igual forma su posterior análisis y caracterización.

Es importante tener en cuenta que el trabajo de comparación se realizó de la manera más objetiva posible, esto es, que se tomaron en cuenta qué características medir, cómo medirlas y sobre todo qué comparar, todo esto tomándolo sobre un contexto lo más uniforme posible, es decir, sin beneficiar de antemano ninguna plataforma, y tratando de minimizar las características especiales o únicas de las plataformas, para así ofrecer un ambiente lejos de particularidades que pudieran inclinar los resultados hacia alguna plataforma.

Para poder medir las dos plataformas tomando en cuenta la característica de que son construcciones de aplicaciones con bases de datos, hay que considerar varios aspectos, sin embargo, la primera particularidad que sale a la vista es la de diferenciar entre qué puntos del mismo desarrollo y/o proceso de construcción se fueron presentando a favor y en contra de cada plataforma, esto es, encontrar que módulos, funciones, implementación de datos, procesos en el código, puesta en operación de funcionalidades se encontraron con una mayor fluidez y facilidad de uso en cada unas de las plataformas o bien cuando se tuvieron huecos, dificultades o tomas de decisiones alternas al análisis inicial por enfrentarse a una posición desfavorable en el momento de su construcción. Este aspecto queda englobado en el punto de “Desarrollo del Sistema”, puesto en capítulos puesta en operación del sistema utilizando RDBMS [5] y puesta en operación del sistema utilizando CMS [6], para cada plataforma respectivamente. Esto si bien pudiera pensarse como una medición aleatoria o cualitativa, también viene acompañada de una realidad muy puntual sobre la complejidad que muestran ambas plataformas, o bien las virtudes de cada una de ellas en diferentes escenarios. De esta manera se diferencian las opciones que como todo un paquete de desarrollo brinda mejores resultados.

Si bien este aspecto va casado con la formación, experiencia y facilidad de desarrollo de la persona en turno en el uso de estas herramientas, no deja de ser un parámetro de comportamiento que pudiera asemejarse a una conducta un tanto normal, ya que al tratar de utilizar las herramientas en sus formas más básicas, no se involucra mucho el conocimiento tan profundo de ellas y eso da un poco la perspectiva de un programador un tanto nuevo en ambas plataformas.

Ahora bien, las otras medidas son más fácilmente ubicadas en el ramo de las mediciones cuantitativas como son: Desempeño de la respuesta del sistema, Integridad, Seguridad y Funcionalidad. A continuación se explora cada una de ellas, para desmenuzar las particularidades de cada una de ellas. Pero antes de entrar en mate-

ria en cada una de ellas, hay que mencionar que todos estos puntos van enfocados al tratamiento y uso de la información que se maneja dentro del sistema. Para ello, este trabajo se apoyó sobre uno previamente realizado en [17], ya que ahí se trabaja un estándar que intenta brindar apoyo a desarrollos de medios pequeños o medianos y que su intención primordial es cuidar el desarrollo adecuado de las bases de datos en todo el contexto de desarrollo de una aplicación.

Si bien se pudiera pensar que el trabajo en [17] sólo está enfocado a las bases de datos, es precisamente esos puntos tan específicos los que dan la medida necesaria para hacer la comparación franca sin importar las capas intermedias de las dos plataformas trabajadas en la tesis, y mucho más importante sin sesgar por dar importancia a las diferencias entre las plataformas. El utilizar este trabajo previo es un intento de rescatar el punto primordial a medir aquí en esta tesis, que es el beneficio otorgado de una plataforma sobre otra en la construcción de aplicaciones con bases de datos, y al decir con bases de datos, efectivamente, se refiere a la intención de centrar la importancia en el manejo, explotación, seguridad, y credibilidad de la información guardada y manipulada en dicho desarrollo.

Hablando sobre el “Desempeño de la respuesta del sistema”, tenemos que en [17] está tomado como lo siguiente: *“Esta prueba verifica que las transacciones a la base de datos se realizan de manera correcta. Como resultado de la prueba se obtiene el rendimiento de la base de datos en casos extremos (transacciones largas)”*, para esto se tiene que cambiar sólo un poco el punto de vista de esta prueba y trasladarlo al margen de las plataformas, es decir, es medir el tiempo de respuesta en consultas, en cuestiones como inserciones y/o registros de usuarios, proyectos o documentos. El tiempo de respuesta del sistema en [15] se define como *“la cantidad de tiempo que toma el sistema en completar una sola tarea”*. Para esta tarea, se tomará como tiempo de respuesta de la aplicación la cantidad de tiempo que toma en completar una sola tarea. Y para tratar de ser mas uniformes en las funcionalidades de cada plataforma se toman varias para descartar o disminuir la posibilidad de sesgo.

Un punto más a revisar es la Integridad en [17] abarca lo siguiente: *“Esta prueba verifica la integridad de los datos. Las pruebas deben verificar las reglas de integridad referencial, de entidad, de dominio, y la integridad definida por el usuario”*. Para el trabajo que se llevó a cabo en esta tesis, se enfocó en los 3 primeros aspectos, por ejemplo, en la integridad referencial, es ver cómo se maneja en ambas plataformas, y corroborar que esa integridad al momento de modificar, crear o eliminar información es respetada. De igual forma pasa con la integridad de entidad y la integridad de do-

minio, es decir, es corroborar como las plataformas se comportan al tratar de violar estas reglas a través de la manipulación de los datos inmersos en el sitio; de la misma forma es reconocer cuando detengan una manipulación incorrecta de la información o bien actúen de manera correcta cuando todo sea un flujo dentro de las reglas.

Sobre el tema de seguridad, la mejor forma de abordarlo es acotarlo ya que se trata de un tema extenso hablando de sistemas de información, la propuesta es tratar de medir a través de los roles, y/o grupos de usuarios definidos, la respuesta de las plataformas al intentar realizar una acción en cada rol que quede fuera de sus privilegios.

Para el punto relacionado con la Funcionalidad, se va a detallar en el plan de pruebas, pero la idea principal es tomar un algunas funcionalidades que posea el sistema en ambas plataformas, para así ver como se realizan y se llevan a buenos términos, esto implica que la funcionalidad debe ser un proceso evidente en el sistema para un usuario final y que debe de entregar resultados y/o debe ser un proceso que tenga valor al momento de finalizar para ese usuario final. Ejemplos de ellos, pueden ser una correcta inscripción de los usuarios, un seguimiento adecuado de un proyecto, el alta correcta de problemas, entre otros.

Una vez definidos como estos puntos son comparados en ambas plataformas, lo siguiente es tener una matriz de pruebas que permitan abarcar una gama de posibilidades tal que del análisis de sus resultados exista la posibilidad de ver varios escenarios y formarse con ello la oportunidad de fijar una postura. Dichas especificaciones se darán a conocer en los capítulos posteriores.

Capítulo 4

Estructura del Sistema en las Plataformas de Estudio

Este capítulo tiene como finalidad ser parte introductoria de los dos siguientes capítulos es decir, Puesta en operación del sistema utilizando RDBMS [5] y Puesta en operación del sistema utilizando CMS [6].

Esto está estructurado así ya que al tratarse de un mismo sistema planteado para dos plataformas distintas, se pensaría que el mapeo de su funcionalidad también podrá observarse en un plano físico, y aunque cierta verdad existe en ello, cabe aclarar que habrá ciertas diferencias que conviene puntualizar para que al adentrarse con más detalle en el desarrollo del sistema en cada tecnología se pueda encontrar con mayor facilidad aquello que representa cierta característica en cada plataforma y viceversa, además de distinguir entre aquello que será totalmente ajeno a una plataforma por tratarse de algo íntimamente relacionado con la naturaleza de la otra.

La estructura general del sistema en ambas plataformas está constituida de las siguientes partes: herramientas y librerías utilizadas, estructura física, seguridad en el sistema y desarrollo del sistema. Estas 4 categorías fueron propuestas debido a que son los puntos más abstractos dentro de las plataformas que muestran un punto en común y dan sentido a su desarrollo, pero además de ello también clarifican aquellas características que hacen de cada plataforma única en el sentido del manejo y uso de la información. En herramientas y librerías utilizadas, se encontrará una breve descripción de aquellas utilidades y productos necesarios para la construcción y correcto uso de esta plataforma en el contexto del sistema antes citado, es importante encapsular esta información para poder hacer una breve distinción entre las plataformas CMS y RDBMS. Para estructura física, lo importante es que el lector establezca de manera sencilla y concreta una vista general del sistema, del trabajo realizado en ambas plataformas y

de manera granular de como está constituido el sistema a fin de crear conciencia de la cantidad de trabajo, pero también de como está organizada la información en ambas plataformas. En el apartado de seguridad en el sistema se encontrarán diferencias enormes entre las plataformas, pero esto sienta las bases del entendimiento y conceptualización de las pruebas sobre funcionalidad, integridad y por su puesto la misma seguridad del sistema. Por último el punto desarrollo del sistema, destaca las ventajas y desventajas de la construcción del sistema en ambas plataformas CMS y RDBMS, esto es parte de las medidas cualitativas que son parte del análisis y medición del sistema en ambas plataformas. El por qué incluir este último en cada capítulo de desarrollo de los sistemas y no en pruebas y/o análisis de resultados, es porque al tratarse de las únicas observaciones cualitativas, el tratamiento inmediato en el apartado de puesta en operación de ambas plataformas da mayor sentido a la información detallada dentro del mismo.

Herramientas y bibliotecas utilizadas

Al hablar de puesta en operación de un sistema, no importando la tecnología que se escoja siempre habrá que ubicarlo dentro de un contexto lógico y físico, para ello este subtema titulado “Herramientas y bibliotecas utilizadas”, ya que es aquí donde se describen para ambas plataformas el servidor donde están alojadas los sistemas que ayudan a la aplicación a estar activa, además este apartado sirve para enlistar de manera puntual todo aquello que se utilizó para el desarrollo de cada sistema, esto va en relación al servidor de aplicaciones, lenguajes de programación, ambiente de desarrollo, base de datos y cualquier otra estructura que de soporte al desarrollo y uso de la nueva aplicación. También, aquí se mencionan las bibliotecas, productos o clases que ayudan a tareas más puntuales y que se integraron al sistema para una adaptación y uso mas eficiente de alguna funcionalidad en particular.

Ya desde este apartado, se pueden empezar a notar las diferencias entre una plataforma y otra, que si bien, ambas tienen herramientas en común como servidor de aplicaciones, ambientes de desarrollo, etcétera, no son exactamente las mismas, sus alcances y/o limitaciones proveerán de ciertas cualidades a una o a otra plataforma tal que, estas puedan tener un tiempo y potencial de desarrollo distintas.

Estructura Física

Estructura Física es un subtema en el que ya empiezan a ver diferencias más drásticas, ya que al tratar la información con paradigmas totalmente diferentes, se

tienen estructuras, métodos, y flujos de información distintos que hacen de la forma física un compuesto desigual entre las plataformas.

La forma en como el lenguaje de programación trabaja, el servidor de aplicaciones conserva activo el portal, la manera en como es tratada la información y como interactuarán los diversos módulos de cada plataforma hacen un cúmulo de necesidades que la estructura física absorbe y con esto se crea un molde único, según el contexto y es por eso que mientras en el capítulo puesta en operación del sistema utilizando RDBMS [5] se tocan temas como arquitectura, patrón de desarrollo, base de datos, en el capítulo puesta en operación del sistema utilizando CMS [6] se hace una referencia de contenidos, contenedores, tipo de ellos y forma de interacción. Si bien, parecieran en primera instancia tratarse de puntos sin relación alguna, todos ellos conllevan al resguardo, presentación y flujo en como la información es llevada bajo cada paradigma, al final de cuentas, lo que une estos puntos va mas allá de sus características inherentes, para converger en un sentido más profundo que es el trato de la información misma.

Seguridad en el Sistema

Un punto básico en todo sistema de información es tener control sobre la seguridad, y esto conlleva a pensar no sólo en cuestiones de resguardo de información, sino de uso y explotación de la misma, distintos privilegios para tratar la información y de manera general, la idea de tener una estructura bien definida del como se prevé que la información este protegida. Llevando esto a las dos plataformas, CMY y RDBMS, es necesario tocar puntos como tipos de usuario, roles, grupos de usuarios, y en general los métodos que utilizan ambas plataformas, o por lo menos, los implementados en estos desarrollos. Hablando sobre seguridad, aunque cada plataforma tiene sus métodos y características que hacen lo posible para proteger la información en varios niveles, ya sea de forma horizontal, es decir, en profundidad sobre la complejidad del sistema, o vertical, haciendo referencia a los tipos de usuarios; ambas implementaciones de seguridad tienen mas similitudes que diferencias, al tratarse como el foco principal el resguardo y buen uso de la información. Sin embargo, y a pesar de lo dicho anteriormente, es importante, resaltar como es que se maneja y se proveen métodos de seguridad, en cual es mas granular, cual es la eficiencia de uno y otro. Todo eso se puede observar al adentrarse en ambos capítulos y ver a través de sus particulares puntos de vista, las diferencias, virtudes y complicaciones que cada plataforma presenta.

Desarrollo del sistema

De manera evidente se puede deducir que este apartado aunque lo comparten ambas plataformas es un punto de vista totalmente desapegado a las características que comparten las plataformas, ya que aquí en “desarrollo del sistema” lo que importa es hacer notar las ventajas y desventajas que cada desarrollo presentó antes, durante y después de su ejecución, cuales fueron aquellos puntos que dan fortaleza a esta plataforma y cuales son oportunidades de mejora o flaquezas dentro de todo este aparato.

Encontrar detalles importantes para dar realce a una característica o a otra para bien o para mal, es todo un reto y en estos subtemas se tratarán de la forma más objetiva sin perder de pista cuestiones generales, ya que parte de esta conciencia de trabajar tendrá lugar en las conclusiones finales de todo este proyecto.

Si bien la intención es mencionar los puntos más interesantes encontrados en el proceso de desarrollo de cada una de las plataformas, la forma de hacerlo se dará tratando de seguir 3 líneas bajo las cuales descansan las características que dan soporte a un buen desarrollo y mantenimiento del sistema, y la calidad del mismo.

La primer línea es siguiendo el modelo de calidad de procesos de desarrollo y mantenimiento de la base de datos, propuesta en [17], la cual abarca los siguientes puntos:

Captura de Requerimientos

Análisis y diseño

Normalización

Modelado físico

Implementación

Cohesión de los datos

Desempeño

Complejidad

Mantenimiento

Modificabilidad

Posibilidad de corregir errores

La segunda forma de apoyar el proceso de construcción y pruebas de este sistema es siguiendo los apartados de Construcción del Software, Pruebas del Software y Calidad del Software propuestos en [25]. Por último, utilizar como base el estandar ISO/IEC 25012 [11] para la calidad de los datos.

Tomando en cuenta esto se detallan 6 áreas a vigilar durante el proceso de desarrollo de software y creación, ejecución y análisis del plan de pruebas:

Funcionalidad: Adecuación, exactitud, interoperabilidad, seguridad y conformidad.

Fiabilidad: Madurez, tolerancia a fallos, recuperabilidad y conformidad.

Usabilidad: Comprensibilidad, facilidad de aprendizaje, operabilidad y conformidad.

Eficiencia: Comportamiento temporal, utilización de recursos y conformidad.

Mantenibilidad: Analizabilidad, cambiabilidad, facilidad de prueba y conformidad.

Portabilidad: Adaptabilidad, facilidad de adaptación, coexistencia, reemplazabilidad y conformidad.

Revisando la calidad de los datos, sobre el plan de pruebas, es importante tener en cuenta las características descritas en la siguiente tabla 4.1:

Tabla 4.1: Calidad de datos iso/eic 25012 necesarios en este trabajo

| Característica | Inherente al sistema | Dependiente del sistema |
|-----------------|----------------------|-------------------------|
| Exactitud | ✓ | |
| Consistencia | ✓ | |
| Credibilidad | ✓ | |
| Accesibilidad | ✓ | |
| Eficiencia | ✓ | ✓ |
| Disponibilidad | | ✓ |
| Recuperabilidad | | ✓ |

Cabe mencionar que estas características no serán revisadas de manera exhaustiva, la intención en los capítulos donde se aborde el tema de “desarrollo del sistema”, será hacer una puntualización de lo encontrado en el desarrollo, pero tratando de mapear las características aquí mencionadas.

Si bien se puede mencionar que existen más temas de los cuales se puede tratar en la puesta en operación de un sistema, como bien son protocolos y/o métodos de recuperación de información o de servicio, formas y flujos de trabajo, políticas de privacidad, métodos de concurrencia, existencia de paralelismo, estas no son consideradas dentro de esta tesis, y más puntualmente de estos capítulos, al tratarse de temas sino ajenos si más alejados de los puntos a contemplar en los capítulos posteriores: plan de pruebas y datos [7], análisis de resultados [8] y conclusiones [9].

Capítulo 5

Puesta en Operación del Sistema Utilizando RDBMS

Este capítulo está dedicado al entendimiento de la estructura del sistema hecho en la versión con RDBMS, se dan detalles de la construcción, se habla de la complejidad de su desarrollo, de las ventajas y desventajas que se tuvieron durante el desarrollo y uso del mismo. Cuestiones como el diseño de la base de datos, la estructura de la aplicación web serán tratadas. También se expondrán el número de páginas realizadas, hojas de estilos hechas, funciones de control y cuestiones sobre la seguridad implementada. En general se va a detallar todas las características que sean relevantes para las pruebas posteriores y para tener un contexto en general del sistema. Para el desarrollo del sistema utilizando RDBMS, se requirió del apoyo del libro [15], así como también del sitio de internet [5]

5.1 Herramientas y bibliotecas utilizadas

El sistema está alojado en un servidor con las siguientes características:

Hardware:

Memoria Ram 2.0 GiB

Procesador 0 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 1 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 2 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 3 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Disco Duro: 129.1 GiB

Software:

S.O. Ubuntu 10.04 LTS

Núcleo Linux 2.6.32-33-generic

GNOME 2.30.2

El sistema será soportado utilizando las siguientes herramientas:

Base de datos relacional: Postgres 8.4

Lenguaje de programación: Java

Lenguajes de apoyo: AJAX, CSS, HTML.

Ambiente de desarrollo (IDE): NetBeans 6.5

Servidor de Aplicaciones: Apache Tomcat

Bibliotecas adicionales.

postgresql-8.3-603.jdbc3.jar JDBC para conexión con la base de datos.

commons-fileupload-1.2.2.jar. Biblioteca de apoyo para subir documentos.

commons-io-2.0.1-javadoc.jar Biblioteca de apoyo para subir documentos.

commons-io-2.0.1-sources.jar Biblioteca de apoyo para subir documentos.

commons-io-2.0.1.jar Biblioteca de apoyo para subir documentos.

JDK 1.5 Para la máquina virtual de Java.

Apache Tomcat 6.0.18 Para el servidor de aplicaciones.

5.2 Estructura Física

La estructura Física del sistema con la plataforma RDBMS se puede implementar de varias formas, sin embargo, un método que da buenos resultados y está ampliamente probado es basarse en uno o varios patrones de diseño. La arquitectura a utilizar para el sistema será emulando el patrón modelo vista controlador (MVC. Model View Controller), Figura 5.1. El MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

La elección de este patrón es debido a que es ampliamente utilizado en aplicaciones

web, ya que ofrece una estructura ordenada y fácil de implementar. Describiéndolo brevemente, se puede decir que la parte del Modelo, es una capa que también es conocida como lógica de negocio, o lógica de aplicación, es la parte encargada de tener comunicación con la base de datos y persistir la información. La Vista especifica la visualización de los datos, es la parte que los usuarios finales conocen del sistema, y es lo que permite la interacción y explotación de los datos en su capa más abstracta. El controlador es el coordinador entre las otras dos capas, es decir, define como la interfaz de usuario reacciona ante la entrada de usuario, para dar paso a nuevas vistas o bien a la comunicación con el modelo.

Con la estructura de un desarrollo en MVC, se hace mucho más sencillo la independencia entre su funcionamiento, debido a que las capas conservan cierta independencia, permite el escalamiento de manera más natural, actualizaciones en la forma, no afectan a la manipulación de los datos y viceversa. El desarrollo en el manejo de errores es facilitado por la clara identificación de los componentes involucrados. Ahora bien, las posibles desventajas de utilizar este patrón podrían ser, cierta complejidad al tener muchos archivos involucrados en el flujo de un proceso, y su mantenimiento y desarrollo requieren de una curva de aprendizaje, que pudiera verse disminuida utilizando modelos más sencillos.

El controlador recibe peticiones a través de los distintos eventos, que se dan sobre la interfaz de usuario. Una vez validada la petición este podrá redirigirla hacia el modelo, o bien hacia la interfaz de usuario nuevamente. El modelo por su parte accede a la base de datos y recupera, suprime, modifica o inserta información, una vez actuando este transfiere el resultado al control para que a su vez este lo muestre hacia la interfaz en caso de ser necesario.

Ahora bien, llevando este patrón al desarrollo realizado, cada una de las partes se componen de varios archivos, clases y componentes, los cuales se detallan en las siguientes secciones.

5.2.1 Vistas

Como ya se mencionó las vistas son parte de la arquitectura del patrón Modelo-Vista-Controlador, para este sitio, la sección de las vistas se componen de varios archivos de apoyo, imágenes y algunos otros elementos que en conjunto dan esta parte de la arquitectura que sirve de puente para que los usuarios puedan hacer uso de la información según sus alcances y beneficios dentro del sistema.

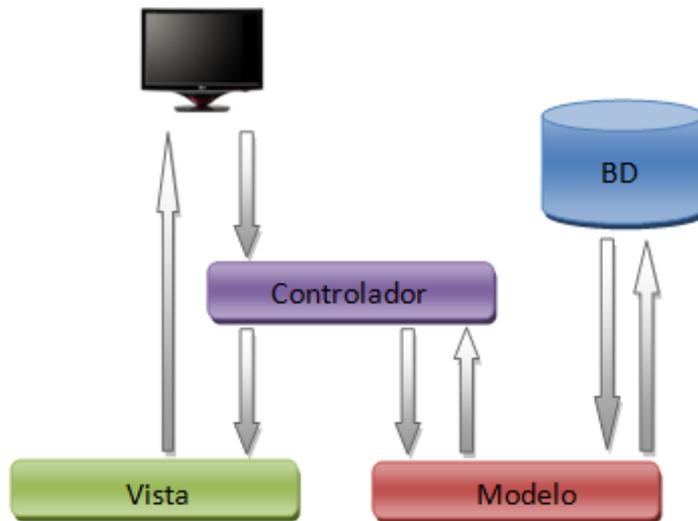


Figura 5.1: Modelo Vista Controlador

Para las vistas que se ocuparon en este sistema se necesitaron los siguientes archivos:

Archivos Javascript

Estos archivos sirvieron de apoyo en momentos de validación de campos, para separar funcionalidad, o bien por orden y hacer más claro cada archivo y no saturarlo con muchas líneas de código.

Tabla 5.1: Descripción de los archivos Java Script

| Archivo | Descripción |
|----------------|---|
| calendario | Archivo de apoyo para validar fechas y mostrar el calendario gráfico. |
| login | Valida los campos del login, ayuda a redireccionar para crear o dar de baja una sesión. |
| validarAlumno | Valida los campos y formatos al momento de dar de alta o modificar un alumno. |
| validarExterno | Valida los campos y formatos al momento de dar de alta o modificar un externo. |
| validarUsuario | Valida los campos y formatos al momento de dar de alta o modificar un usuario. |

Archivos CSS (Cascading Style Sheets)

Estos dos archivos ayudan a darle estilo, definir formas, colores, tamaños y todo aquello que se necesitó para darle la vista adecuada al portal.

Tabla 5.2: Descripción de los archivos CSS

| Archivo | Descripción |
|------------|--|
| calendario | Archivo dedicado al estilo del calendario gráfico. |
| style | Archivo de estilos de todo el portal. |

Imágenes

Listar cada una de las imágenes no es tan necesario, lo que sí se debe de mencionar es que estas son íconos, parte del fondo de la vista, botones y que junto con los estilos css complementan la visualización del sitio. Aunque haciendo un recuento de lo realizado en cuanto a imágenes se tiene lo siguiente:

27 imágenes PNG *

1 imagen JPG *

22 imágenes GIF *

1 imagen Favicon (ico) *

* Todas involucradas en la mayoría de las páginas del portal.

Archivos PSD

Se trabajó con 4 archivos de diseño, estos dieron cabida al surgimiento de los css y de las imágenes antes mencionadas, mucho del cómo se visualizó el portal tiene su origen en estos archivos psd.

Los archivos cuadro.psd y unamlogo.psd están dedicados al fondo del portal, por otra parte home.psd contiene la idea y diseño general de la pagina y por último menu.psd es utilizada para los botones de cada rol creado.

Archivos JSP

Esta lista de archivos constituye todo la vista del sistema, a partir de archivos jsp, los cuales sirven de interfaz para que el usuario pueda interactuar con los datos, darles forma, agregar, modificar o eliminar información que será resguardada por el portal mismo. Dicha lista se puede consultar en la tabla [5.3](#)

Tabla 5.3: Descripción de las vistas

| Página | Descripción |
|--------------------|---|
| administradores | Reporte de administradores y académicos con opción a cambiar el rol entre ellos. |
| alta_alumnos | Reporte, alta, modificación y eliminación de usuarios con rol de alumno. |
| alta_externos | Reporte, alta, modificación y eliminación de usuarios con rol de externo. |
| asignar_depto | Dar de alta o quitar relación entre académico, grupo de trabajo o laboratorio a uno o varios departamentos e institutos. |
| bottom | Archivo incrustado en todos los restantes, indica el pie de página de cada jsp, donde viene ligas a inicio, mapa de sitio, y información referente a la última actualización del sistema. |
| catalogo_equiplab | Descripción de cada laboratorio y su equipo, así como una liga de interés |
| ccategoria | Reporte, alta, modificación y eliminación de categoría de académicos. |
| cdepartamento | Reporte, alta, modificación y eliminación de departamentos. |
| centidadfederativa | Reporte, alta, modificación y eliminación de entidades federativas. |
| ccampoconocimiento | Reporte, alta, modificación y eliminación de campos de conocimiento. |
| cinstitucion | Reporte, alta, modificación y eliminación de institutos. |
| ctipodocumento | Reporte, alta, modificación y eliminación de tipo de documentos. |
| ctitulo | Reporte, alta, modificación y eliminación de titulo de académicos. |
| equipo.lab | Reporte, alta, modificación y eliminación de laboratorio. |
| extra | Página estática con información relevante a aspectos circundantes a la Secretaría de Vinculación. |
| index | Página estática con información de bienvenida a la Secretaría de Vinculación. |
| integrantes_gpo | Reporte, alta y eliminación de integrantes de grupos de trabajo. |
| integrantes.lab | Reporte, alta y eliminación de integrantes de laboratorios. |
| links | Página estática con información sobre links de interés relacionados con la Secretaría de Vinculación. |
| mapa | Página estática con información sobre las páginas y estructura del sitio. |
| menu | Archivo incrustado en todos los restantes, indica el menú de botones y dependiendo del rol del usuario muestra las opciones pertinentes. |
| mis_datos | Página que funciona para modificar los datos del registro del usuario. |
| mis_problemas | Reporte, alta y eliminación de problemas de los clientes. |
| mis_servicios | Reporte, alta y eliminación de servicios de académicos, grupos de trabajo o laboratorios. |
| mision | Página estática con información sobre la misión de la Secretaría de Vinculación. |
| objetivo | Página estática con información sobre los objetivos de la Secretaría de Vinculación. |
| politicas | Página estática con información sobre las políticas relevantes a la Secretaría de Vinculación. |
| proyecto_act | Reporte, alta, modificación y eliminación de actividades realizadas en un proyecto. |
| proyecto_aporta | Reporte, alta, modificación y eliminación de aportaciones realizadas en un proyecto. |
| proyecto_detalle | Reporte sobre el detalle general de un proyecto. |
| proyecto_docs | Reporte y alta sobre los documentos de apoyo de un proyecto. |
| proyecto_ent | Reporte, alta, modificación y eliminación de entregables dentro de un proyecto. |
| proyecto_mod | Modificación de datos generales de un proyecto. |
| proyecto_pagos | Reporte, alta, modificación y eliminación de pagos realizados en un proyecto. |
| proyectos | Reporte, alta y eliminación de proyectos. |
| qsomos | Página estática con información sobre quienes integran la Secretaría de Vinculación. |
| recuperar | Página dedicada a la recuperación de la contraseña. |
| registrarse | Contiene los elementos de cada rol necesarios para el registro al portal. |
| subirdocproy | Archivo de apoyo, para almacenar documentos del proyecto en el servidor. |
| subirproblema | Archivo de apoyo, para almacenar documentos de los problemas en el servidor. |
| subirservicio | Archivo de apoyo, para almacenar documentos de los servicios en el servidor. |
| top | Archivo incrustado en todos los restantes, indica titulo de bienvenida, quien está en sesión o bien las opciones para el login del sitio. |
| ver_clientes | Página para proveedores que muestra los clientes registrados en el portal. |
| ver_problemas | Página de abierta que muestra los problemas registrados en el portal. |
| ver_proveedores | Página para clientes que muestra los proveedores registrados en el portal. |
| ver_proyectos | Página de abierta que muestra los proyectos registrados en el portal. |
| ver_servicios | Página de abierta que muestra los servicios registrados en el portal. |
| ver_soluciones | Página para clientes que muestra las soluciones a problemas registrados en el portal. |
| vision | Página estática con información sobre visión de la Secretaría de Vinculación. |

5.2.2 Modelo

Solo tiene una clase GestionDatos.java, la cual contiene alrededor de 155 funciones que ayudan a conectarse a la base de datos, obtención de fechas y el resto es repartida

para persistir información proveniente de las vistas y el control, y recuperación de información para reportes y/o combos.

Como archivos de apoyo, se tiene 3 beans (Alumnos.java, Externos.java y Usuario.java), que sirven para mapear lo escrito en un formulario con los campos de distintas entidades en la base de datos.

5.2.3 Control

En la parte del control, se tiene igual que en el modelo un solo archivo que sirve para dar la funcionalidad a toda esta estructura, el archivo control.java contiene aproximadamente 153 flujos para capturar todos los eventos solicitados por las vistas.

5.2.4 Base de Datos

La base de datos fue lo primero que se abordó al tratarse la plataforma RDBMS. La base de datos brinda sustento y contexto a los datos, aporta seguridad y la certeza de que se tiene unos cimientos adecuados para soportar todo el desarrollo del sistema. Se cuenta con un total de 37 tablas. El diagrama de la base de datos completo se muestra en la parte de Apéndice A en la figura [A.1](#). Cabe mencionar que la intención de este diagrama no es que el lector vea a detalle la estructura de la base de datos, es solamente un apoyo que da una idea de todo el conjunto que se creó para darle sentido a la información que se iba a alojar en el manejador de base de datos. Mas adelante se muestran las 37 tablas de manera independiente para un mejor detalle, ahora sí, sin puntualizar sus relaciones, sino mas bien revisando entidad por entidad sus características inherentes.

5.3 Seguridad en el sistema

La seguridad que se implementó para este sistema está basado en si las personas están conectadas en sesión al sistema y dependiendo del tipo de usuario se le da acceso a ciertas vistas y por ende a ciertas funcionalidades dentro del sistema mismo. Otra forma de protección con la que cuenta la información alojada en el sistema es el usuario mismo de la base de datos, encargado de persistir las consultas realizadas en el sistema a través de las diversas peticiones que los usuarios hacen dentro el mismo. En la

tabla 5.4. se muestra una matriz de tipos de usuarios y las vistas que son permitidas en cada caso. Las columnas (roles) están descritas a continuación: 1.- Administrador, 2.- Académico, 3.- Grupo de trabajo, 4.- Laboratorio, 5.- Persona Física, 6.- Persona Moral, 7.- Alumno. (Es parte de un catálogo, en realidad es un usuario sin registro), 8.- Externo. (Es parte de un catálogo, en realidad es un usuario sin registro), 9.- Visitante del sistema.

5.4 Desarrollo del sistema

VENTAJAS

Lo primero que se llevó a cabo de este sistema fue el diseño de la base de datos; ir definiendo las entidades, sus características y los formatos de cada atributo, las relacionales entre estas entidades y hacer un trabajo de rediseño hasta hacer que todas las relaciones cuadraran y formaran un todo armónico, lo cual, da seguridad y certeza a la idea de formar una estructura muy sólida que permita un inicio en la programación del sistema respetando los lineamientos inherentes del sistema. De esta forma se asegura, que la información va a gozar de los privilegios de estar almacenada y administrada en un sistema gestor de bases de datos relacionales, es decir, la información desde ese momento cuenta con procesos y módulos encargados de la seguridad, concurrencia, integridad y fiabilidad. Al entrar en este proceso se cuidó los detalles de la normalización, la cohesión de los datos, y el modelado físico de los mismos.

Al implementar el patrón MVC para el desarrollo del sitio, se encontró que una de las ventajas más notables es la escalabilidad y la separación de componentes, haciendo que la detección de errores y mantenimiento sea más rápida al contar con esta separación de código y funcionalidad; aquí la modificabilidad se vio favorecida con la implementación de este patrón.

Al ir desarrollando el sitio otra ventaja que se encontró es que varios módulos de reportes, altas, bajas o modificaciones de catálogos, se parecen mucho debido a la estructura mostrada en el diseño de la base de datos, por tanto al tener una interfaz, un manejo de eventos en el control y/o una función que permita mapear los resultados del formulario hasta hacerlos persistentes en la base de datos, o viceversa, lo restante era replicar dichos procesos y hacer cambios menores para adaptarlo a nuevos contextos. Esto es, reciclar el código fue algo que permitió que el avance en tiempo y en complejidad disminuyera al tomar en cuenta los catálogos de este sitio (ccatego-

Tabla 5.4: Matriz de Roles y accesos

| Páginas/Roles | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---------------------|---|---|---|---|---|---|---|---|---|
| administradores | ✓ | | | | | | | | |
| alta_alumnos | ✓ | ✓ | ✓ | ✓ | | | | | |
| alta_externos | ✓ | ✓ | ✓ | ✓ | | | | | |
| asignar_depto | ✓ | ✓ | ✓ | ✓ | | | | | |
| bottom | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| catalogo_equiplab | ✓ | | | ✓ | | | | | |
| ccategoria | ✓ | | | | | | | | |
| cdepartamento | ✓ | | | | | | | | |
| centidadfederativa | ✓ | | | | | | | | |
| cerrar_sesion | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| ccampoconocimiento | ✓ | | | | | | | | |
| cinstitucion | ✓ | | | | | | | | |
| correoejemplo | | | | | | | | | |
| ctipodocumento | ✓ | | | | | | | | |
| ctitulo | ✓ | | | | | | | | |
| equipo_lab | ✓ | | | ✓ | | | | | |
| extra | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| index | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| integrantes_gpo | ✓ | | ✓ | | | | | | |
| integrantes_lab | ✓ | | | ✓ | | | | | |
| links | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| mapa | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| menu | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| mis_datos | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| mis_problemas | ✓ | | | | ✓ | ✓ | | | |
| mis_servicios | ✓ | ✓ | ✓ | ✓ | | | | | |
| mision | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| objetivo | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| politicas | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| proyecto_act | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_aporta | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_detalle | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_detallelog | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_docs | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_ent | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_mod | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyecto_pagos | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| proyectos | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| qsomos | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| recuperar | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| registrarse | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| subirdocproy | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | |
| subirproblema | ✓ | | | | ✓ | ✓ | | | |
| subirservicio | ✓ | ✓ | ✓ | ✓ | | | | | |
| top | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_clientes | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_problemas | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_proveedores | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_proyectos | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_servicios | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| ver_soluciones | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

ria, cdepartamento, centidadfederativa, catalogo_equiplab, ccampoconocimiento, cinsti-
tucion, ctipodocumento, ctitulo). Del mismo modo, la reutilización de código, se
presentó en los reportes aumentando la facilidad al codificar y sobre todo evitó rehacer
ciertas partes, dicho ventaja se logró al sobre escribirlas con las particularidades de

cada reporte (`ver_clientes`, `ver_problemas`, `ver_proveedores`, `ver_proyectos`, `ver_servicios`, `ver_soluciones`). Aquí se refleja mucho las características de calidad como mantenabilidad, modificabilidad y adecuación.

En todo el sitio se encuentra un claro uso de la eficiencia, ya que, se encuentran partes de código muy similares o prácticamente iguales, las cuales son reutilizadas y esto representa una ventaja debido a que si en una primera implementación, no se tienen errores de código, se puede replicar el funcionamiento y esto ofrece cierta ventaja en cuanto a la rapidez y producción.

La utilización del Control y del Modelo, las primeras veces pudieron ocasionar ciertos retrasos, pero una vez entendidos y obtenidos son fácilmente adaptables y también presentaron gran reutilización de código.

Algo que también representó una significativa ventaja fue el control del flujo de información a través de los botones y sus bloqueos por roles, así como en cada página la validación encargada de ver si era requerido verificar el inicio de sesión de los usuarios y de que tipo de usuario se trata. Aunque implementarlo no fue complicado, su creación robustece mucho la programación del sitio y brinda la seguridad sobre la funcionalidad que los distintos tipos de usuarios o roles y niveles que estos tiene al usar la página se haga valer y se lleve de acuerdo a lo establecido, es decir, brinda alto grado de confiabilidad.

Probar funcionalidades, validaciones o flujos en general suele ser algo sencillo debido a que se puede seguir el código y paso a paso el control de variables y el camino que van siguiendo en el programa; esto aporta un punto importante en la posibilidad de corregir errores.

DESVENTAJAS

Parte de las desventajas al ir desarrollando sobre esta plataforma, es que encontrar errores en la programación a veces suele ser complicado, debido que hay que seguir el flujo rastreando el proceso en el código y tratando de ver si es un problema de sintaxis, de compatibilidad entre el modelo y la base de datos, o de formato de variables y/o atributos de negocio, es decir, este punto refleja un alto grado de complejidad en la corrección de errores cuando el código crece.

Cambios surgidos en la estructura de la base de datos a veces se convierte en un proceso que no es muy transparente y se deben contemplar algunos cambios en cascada.

Luego implica invertir tiempo en aplicar todos los cambios, es decir la mantenabilidad se puede ver comprometida en ciertos aspectos.

La interacción al subir los documentos al servidor, o bien, reflejar el envío del correo fueron dos puntos muy difíciles de resolver, ya que programarlos sin ayuda de bibliotecas (commons-io y commons-fileupload) hubiera sido muy costoso en cuanto a la curva de aprendizaje y tiempo invertido; ahora bien al aprovechar dichas bibliotecas se tienen ciertas limitaciones y de todas formas hubo que ocupar tiempo en ajustarlo, pero sin duda fue una mejor solución, aunque se considera desventaja ya que fueron dos puntos difíciles de pasar con esta plataforma.

Cierta desventaja que se presenta a medida que el sistema crece es que el control (control.java) y el Modelo (gestionDatos.java) aumentan en tamaño y complejidad de forma considerable, esto debido a que para controlar los eventos de las interfaces o la disponibilidad de la información, se necesita considerar más de un camino o posibilidad por interfaz. Al tener entonces un Control y/o Modelo grande se debe tener una estricta organización para no tener confusiones. Para contrarrestar esto, es recomendable llevar un control en los nombres para que sean representativos de lo que las funciones o su interior harán. Aunque si bien para este sistema se tiene un archivo de control y otro de modelo de un tamaño considerable, esto no parece haber afectado mucho, sin embargo, tenerlo segmentado o llevar a cabo una repartición en cuanto a grandes divisiones o módulos hubiera resultado más funcional y eficiente.

A continuación se detallan las tablas utilizadas para la construcción del sistema bajo la plataforma con el RDBMS.

Notación:

Llave - indica que el campo es parte de la llave primaria

Diamante con relleno blanco - indica que el campo no es obligatorio

Diamante con relleno rojo - indica que ese campo contiene una llave foránea

Diamante con relleno azul - indica que ese campo es obligatorio

| Tabla | Campos | Índices |
|--------------------------|--|--|
| categorias | <ul style="list-style-type: none"> idcategorias INT(11) nombre_f VARCHAR(70) nombre_m VARCHAR(70) abreviatura_f VARCHAR(45) abreviatura_m VARCHAR(45) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY |
| academico | <ul style="list-style-type: none"> idusuario INT(11) rfc_aca VARCHAR(13) numerotrabajador DECIMAL(10,0) nombre VARCHAR(45) apaterno VARCHAR(45) amaterno VARCHAR(45) numeroplaza DECIMAL(11,0) idcategorias INT(11) idtitulo INT(11) | <ul style="list-style-type: none"> PRIMARY no_trabajador_unico rfc_unico fk_academico_categorias1 fk_academico_proveedor1 fk_academico_titulo1 |
| actividades | <ul style="list-style-type: none"> idactividades INT(11) idservicioresolver INT(11) idservicioofrecido INT(11) nombre VARCHAR(90) descripcion TEXT fechainicio DATE fechafin DATE idproyecto INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) idusuario INT(11) tipousuario INT(11) | <ul style="list-style-type: none"> PRIMARY fk_actividades_proyecto1 fk_actividades_servicioresolver1 fk_actividades_servicioofrecido1 |
| alumno | <ul style="list-style-type: none"> idalumno INT(11) numerocuenta VARCHAR(10) fechaingreso DATE nombre VARCHAR(45) apaterno VARCHAR(45) amaterno VARCHAR(45) fechanacimiento DATE semestreactual DECIMAL(2,0) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY no_cuenta_unico |
| aportaciones | <ul style="list-style-type: none"> idpagosproyecto INT(11) idproyecto INT(11) concepto VARCHAR(45) monto VARCHAR(45) fechapago DATE observaciones TEXT numerorecibooficial VARCHAR(45) idformapago INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY fk_aportaciones_formapago1 fk_pagos_proyecto_proyecto1 |
| entidadfederativa | <ul style="list-style-type: none"> identidadfederativa INT(11) nombre VARCHAR(90) descripcion TEXT idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY |

Figura 5.2: Tabla Categorías, Tabla Académico, Tabla Actividades, Tabla Alumno, Tabla Aportaciones, Tabla Entidad Federativa

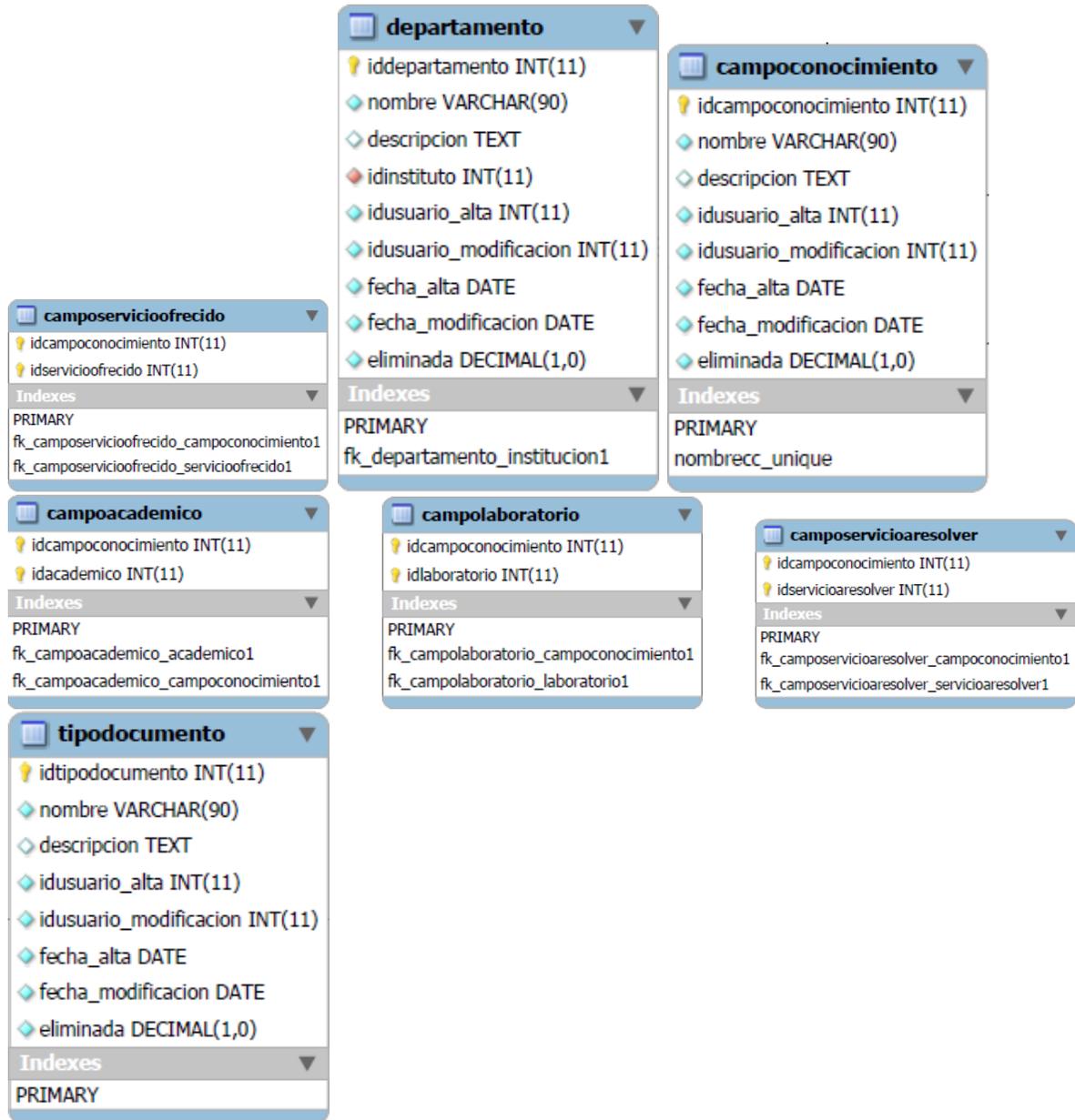


Figura 5.3: Tabla relación campo de conocimiento y Servicio Ofrecido, Tabla Departamento, Tabla relación campo de conocimiento y Académico, Tabla Campo de Conocimiento, Tabla relación campo de conocimiento y Laboratorio, Tabla relación campo de conocimiento y Servicio a Resolver, Tabla Tipo de Documento

| Tabla | Campos | Índices |
|---------------------------------------|---|--|
| entregable | <ul style="list-style-type: none"> identregable INT(11) versionentregable INT(11) idactividades INT(11) descripcion TEXT fechaentrega DATE ubicacion VARCHAR(255) idresponsable INT(11) numeroregistro VARCHAR(45) idproyecto INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) nombre VARCHAR(90) tipousuario INT(11) | PRIMARY fk_entregable_proyecto1 fk_productos_actividades1 fk_productos_proveedor1 |
| equipolaboratorio | <ul style="list-style-type: none"> idequipolaboratorio INT(11) idlaboratorio INT(11) nombre VARCHAR(45) descripcion TEXT cantidad DECIMAL(9,0) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY fk_equipolaboratorio_laboratorio1 |
| especificacionesproyecto | <ul style="list-style-type: none"> idespecificacion INT(11) nombre VARCHAR(70) idproyecto INT(11) numeroregistro VARCHAR(45) idtipodocumento INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY fk_especificacionesproyecto_proyecto1 fk_especificacionesproyecto_tipodocumento1 |
| especificacionservicioresolver | <ul style="list-style-type: none"> idespecificacion INT(11) idservicioresolver INT(11) nombre VARCHAR(70) idtipodocumento INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY fk_especificaciones_servicioresolver1 fk_especificaciones_tipodocumento1 |
| especificacionservicioofrecido | <ul style="list-style-type: none"> idespecificacion INT(11) nombre VARCHAR(70) idtipodocumento INT(11) idservicioofrecido INT(11) precio DECIMAL(9,2) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY fk_especificacionservicioofrecido_servicioofrecido1 fk_especificacionservicioofrecido_tipodocumento1 |
| externo | <ul style="list-style-type: none"> idexterno INT(11) nombre VARCHAR(45) apaterno VARCHAR(45) amaterno VARCHAR(45) rfcexterno VARCHAR(13) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY rfcad_unique |

Figura 5.4: Tabla Entregable, Tabla Equipo de Laboratorio, Tabla Especificaciones de proyecto, Tabla Especificaciones de Servicio a Resolver, Tabla Especificaciones de Servicio Ofrecido, Tabla Externos

| Table Name | Fields | Indexes |
|-----------------------------|---|--|
| formapago | <ul style="list-style-type: none"> idformapago INT(11) nombre VARCHAR(45) descripcion TEXT idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY |
| grupodetrabajo | <ul style="list-style-type: none"> idusuario INT(11) nombre VARCHAR(45) descripcion TEXT idacademicoresponsable INT(11) | PRIMARY nombregt_unique fk_grupodetrabajo_academico1 fk_grupodetrabajo_proveedor1 |
| grupotrabajo_aca_alu | <ul style="list-style-type: none"> idusuario_grupodetrabajo INT(11) idusuario INT(11) tipousuario INT(11) | PRIMARY fk_grupotrabajo_academico_grupodetrabajo1 |
| instituto | <ul style="list-style-type: none"> idinstituto INT(11) nombre VARCHAR(90) descripcion TEXT idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | PRIMARY |
| laboratorio | <ul style="list-style-type: none"> idusuario INT(11) nombre VARCHAR(45) descripcion TEXT ubicacion VARCHAR(200) costohora DECIMAL(9,2) horario VARCHAR(100) idacademicoresponsable INT(11) | PRIMARY nombre_unique fk_laboratorio_academico1 fk_laboratorio_proveedor1 |
| laboratorio_aca_alu | <ul style="list-style-type: none"> idlaboratorio INT(11) idusuario INT(11) tipousuario INT(11) | PRIMARY fk_laboratorio_aca_alu_laborator |

Figura 5.5: Tabla Forma de Pago, Tabla Grupo de Trabajo, Tabla Relacion Grupo de Trabajo - Academicos y Alumnos, Tabla Instituto, Tabla Laboratorio, Tabla Relacion Laboratorio - Academicos y Alumnos

| Table Name | Fields | Indexes |
|------------------------------|--|--|
| pagosfunciones | <ul style="list-style-type: none"> idpagofunciones INT(11) monto VARCHAR(45) idusuario INT(11) fechapago DATE periodotrabajado VARCHAR(45) numeromeses DECIMAL(2,0) idformapago INT(11) idproyecto INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) tipousuario INT(11) | <ul style="list-style-type: none"> PRIMARY fk_pagosfunciones_formapago1 fk_pagosfunciones_proyecto1 |
| usuario | <ul style="list-style-type: none"> idusuario INT(11) tipousuario VARCHAR(1) usuario VARCHAR(45) contrasena1 VARCHAR(45) genero VARCHAR(1) estatus VARCHAR(1) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY usuario_unico |
| personamoral | <ul style="list-style-type: none"> idpersonamoral INT(11) nombrepersonamoral VARCHAR(45) razonsocial VARCHAR(150) rfc_pm VARCHAR(12) giro VARCHAR(45) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY razonsocial_unico rfc_pm_unico |
| proveedor | <ul style="list-style-type: none"> idusuario INT(11) email VARCHAR(100) telefono VARCHAR(20) celular VARCHAR(20) identidadfederativa INT(11) delegacionmunicipio VARCHAR(70) colonia VARCHAR(70) codigopostal VARCHAR(5) calle VARCHAR(70) | <ul style="list-style-type: none"> PRIMARY fk_proveedor_entidadfederativa1 fk_proveedor_usuario1 |
| proveedordepartamento | <ul style="list-style-type: none"> iddepartamento INT(11) idproveedor INT(11) tipousuario INT(11) | <ul style="list-style-type: none"> PRIMARY fk_proveedordepartamento_departamento1 fk_proveedordepartamento_proveedor1 |
| proyecto | <ul style="list-style-type: none"> idproyecto INT(11) nombre TEXT descripcion TEXT fechainicio DATE fechafin DATE costo DECIMAL(9,2) estatusproyecto DECIMAL(1,0) numeroregistro VARCHAR(45) objetivo TEXT idresponsable INT(11) idservicioresolver INT(11) idusuario_alta INT(11) idusuario_modificacion INT(11) fecha_alta DATE fecha_modificacion DATE eliminada DECIMAL(1,0) | <ul style="list-style-type: none"> PRIMARY numeroregistro_unico fk_proyecto_academico1 fk_proyecto_servicioresolver1 |

Figura 5.6: Tabla Pagos por Funciones, Tabla Usuario, Tabla Persona Moral, Tabla Proveedor, Tabla Relacion Proveedor - Departamento, Tabla Proyecto

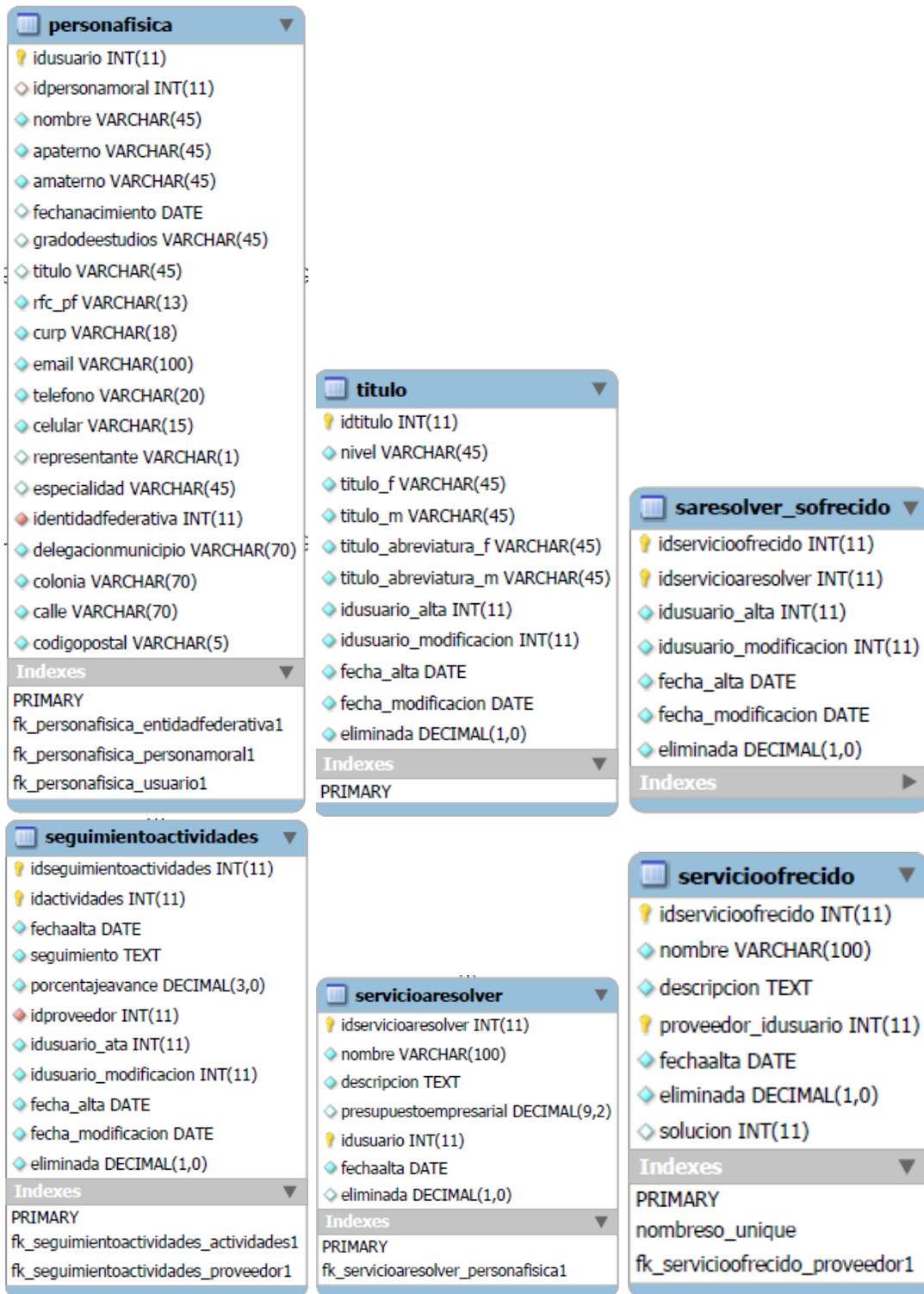


Figura 5.7: Tabla Persona Física, Tabla Título, Tabla de relacion Soluciones, Tabla Seguimiento de Actividades, Tabla Servicio a Resolver, Tabla Servicio Ofrecido

Capítulo 6

Puesta en Operación del Sistema Utilizando CMS

En este capítulo se van a detallar las características del desarrollo del sistema hecho con la plataforma CMS. El diseño de los contenedores, la estructura de la aplicación web, los productos utilizados, los procesos, flujos de trabajo y cuestiones sobre seguridad implementada serán parte de lo que se detalla en esta sección. También se expondrán el número y tipo de contenidos realizados, hojas de estilos modificadas, funciones de control. En general se va a detallar todas las características que sean relevantes para las pruebas posteriores y para tener un contexto en general del sistema bajo este paradigma. Para el desarrollo del sistema utilizando CMS, se requirió del apoyo de los libros [22], [13] y [18]; así como también de los sitios de internet [4], [6], [10], [9] y [2]

6.1 Herramientas y bibliotecas utilizadas

El sistema está alojado en un servidor con las siguientes características:

Hardware:

Memoria Ram 2.0 GiB

Procesador 0 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 1 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 2 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Procesador 3 : Intel(R) Xeon(R) CPU 5110 @ 1.60GHz

Disco Duro: 129.1 GiB

Software:

S.O. Ubuntu 10.04 LTS

Núcleo Linux 2.6.32-33-generic

GNOME 2.30.2

El sistema será soportado utilizando el CMS Plone, dentro de este existen las siguientes herramientas:

Repositorio de Objetos: ZODB

Lenguaje de programación: Python 2.6.7

Lenguajes de apoyo: AJAX, CSS, HTML.

Ambiente de desarrollo (IDE): Plone 4.1.0.9

Servidor de Aplicaciones: Zope 2.13.8

Productos extras relevantes en el uso y desarrollo del sistema:

PloneFormGen 1.7.0

PloneFormGen Save Data to Content 1.3.1

jQuery UI 1.8.16.4

MassLoader 3.1.0

6.2 Estructura Física

El paradigma que rodea este sistema es el de los sistemas gestores de contenidos o CMS. Estos hacen referencia a que todo dentro del sistema es un contenido, incluso el sistema mismo es considerado un contenido por el servidor de aplicaciones, que a su vez como está programado en Python, es considerado un objeto. El lenguaje de programación todo lo maneja como objetos, por tanto es posible englobar en estos dos conceptos generales todo el desarrollo del sistema. Es decir, dentro del sistema existen objetos contenidos dentro de objetos más generales, estos objetos pueden ser o no contenidos y/o contenedores. Un diagrama de como está compuesto este CMS se puede consultar en la parte de anexos en la tabla [A.14](#).

6.2.1 Contenidos

Detallando un poco más la definición previa, se ampliará en las características y uso de de los contenidos utilizados en este sistema.

Contenidos en Plone

Los contenidos de mayor uso dentro del IDE Plone y los productos instalados son los que figuran en la tabla 6.1

Tabla 6.1: Tabla de contenidos mas usados en Plone

| Contenido | Contenido |
|-------------------------------|-----------------------------------|
| Adaptador de envío de correo | Adaptador de script personalizado |
| Adaptor de guardado de datos | Archivo |
| Campo Captcha | Campo de Texto |
| Campo de contraseña | Campo de etiqueta |
| Campo de etiqueta con formato | Campo de fichero |
| Campo de líneas | Campo de número decimal |
| Campo de número entero | Campo de puntuación con escala |
| Campo de selección | Campo de selección múltiple |
| Campo de texto con formato | Campo string |
| Carpeta | Carpeta de conjunto de campos |
| Checkbox Field | Colección |
| Comentarios | Date/Time Field |
| Enlace | Evento |
| Fieldset Begin | Fin de conjunto de campos |
| Formulario | Imagen |
| Noticia | Página |
| Página de agradecimiento | Page Template |
| Sript Python | Save Data to Content Adapter |
| Save Data to Content Entry | |

Sin embargo, no todos fueron indispensables, por eso sólo se detallarán aquellos que tomaron parte dentro del sistema.

Archivo

Un archivo en Plone es cualquier archivo binario que es utilizado de manera general en el sistema, o bien, un archivo cargado al sistema, para ser compartido por usuarios. Ejemplos de estos archivos son aquellos que se suben en las secciones de documentos para servicios a resolver, proyectos y/o servicios ofrecidos. Estos pueden ser archivos PDF, documentos de Word, hojas de cálculo, imágenes, archivos rar, zip, hojas de presentación y muchos más, todos ellos configurados en la sección “mimetypes_registry”, el cual es un portal dentro del sistema otorgado en los productos por defecto en Plone que organiza los detalles de los tipos que pueden coexistir en el sistema. Ahí mismo se

pueden configurar detalles como su nombre, la extensión del tipo de archivo, y si es de tipo binario o no.

Campo de Texto

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo de Texto común utilizado para escribir una o múltiples líneas de texto para descripciones, direcciones, nombres, identificadores, teléfonos, números entre otros.

Campo de Fichero

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo para subir un archivo. Tiene la capacidad de cargar cualquier tipo de archivo binario.

Campo de selección

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo de selección de tipo lista para escoger una opción de entre un conjunto dado.

Campo de selección múltiple

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo de selección múltiple de tipo lista para escoger una o varias opciones de entre un conjunto dado.

Campo string

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo de Texto común utilizado para escribir una línea de texto para direcciones, nombres, identificadores, teléfonos, números entre otros.

Carpeta

Este es una propiedad nativa de Plone. Una carpeta en Plone es similar a una carpeta en un sistema de ficheros, es decir, solo sirve para localizar el contenido dentro de él de manera mas eficiente.

Checkbox Field

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo que arroja un resultado booleano, comúnmente utilizado para determinar un cierto estado.

Comentarios

Este es una propiedad del producto nativo de Plone, es un campo de texto utilizado y creado en cada contenido para ser utilizado en búsquedas y para aclarar la función del contenido específico que alberga dicho comentario.

Date/Time Field

Este es una propiedad del producto PloneFormGen, es utilizado para la estructura de un formulario. Es un campo guardar fecha y hora de algo en específico, en el sistema generalmente su uso es reservado solo para los datos del día, no de la hora.

Enlace

Este es una propiedad del producto nativo de Plone, es utilizado para la estructura de un formulario. Es un hipervínculo para saltar de un contenido a otro dentro del sistema.

Formulario

Este es una propiedad del producto PloneFormGen, el Formulario es la propiedad más relevante dentro de PloneFormGen, debido a que es un contenedor de todas las otras propiedades.

Imagen

Este es una propiedad del producto nativo de Plone, es un archivo específico que puede ser compartido por los usuarios.

Página

Este es una propiedad del producto nativo de Plone, es un archivo de estructura sencilla de tipo html para poder darle funcionalidad de tipo web al sistema, pueden ser normalmente usadas para información estática.

Page Template

Este es una propiedad del producto nativo de Plone, es un archivo de estructura sencilla de tipo html enriquecida, es decir aquí pueden ser montados lenguajes como python, javascript, dtml y otros para poder hacer páginas más robustas y dinámicas.

Script Python

Son archivos de apoyo de lenguaje Python. En este sistema son utilizados principalmente para mostrar, manipular y darle flujo a la información necesaria. General-

mente pueden ser montados en Page Templates, Paginas, o otros archivos de soporte.

Save Data to Content Adapter

Este es una propiedad del producto PloneFormGen, está propiedad permite que una vez validados los datos del formulario, se transformen y se puedan guardar y manipular no como datos de formulario sino como contenidos y así formar parte de una forma natural del paradigma de los CMS.

En la parte de Apéndice A, se puede dar una vista completa de la disposición física y jerárquica del sistema, esto con el fin de comprender su complejidad y organización, la forma en como se van relacionando los contenidos y tener ubicados los procesos por los que la información es contenida desde su raíz hasta el punto más profundo en el sistema. En las imágenes que se anexan se estructura en forma de árbol, esta vista es la más conveniente debido al encapsulamiento que muestra el paradigma de los gestores de contenidos. La serie de figuras de la [A.2](#) a la [A.12](#) agregadas en la parte de Apéndice A, representan el árbol del sistema y sus componentes.

6.3 Seguridad en el sistema

La seguridad en los CMS es un tanto distinta que en los sistemas realizados con apoyo de los RDBMS, en el caso de Plone el desarrollo de la seguridad del sistema se basa en una serie de mecanismos y componentes que se ayudan entre sí para lograr los fines deseados.

Sumergirse en el entendimiento de la seguridad de un sistema hecho con Plone es revisar, entender y aplicar tres grandes áreas en el tema de seguridad. Si bien estos temas pueden sonar conocidos, cada uno aborda particularidades del CMS y la forma como interpreta estos elementos.

6.3.1 Estado de un contenido

El paradigma de CMS da la posibilidad de manejar y tener un flujo sobre los contenidos ubicados en el portal. Estos estados o fases hacen que el contenido adquiera ciertos bloques para diversos roles y por ende grupos y usuarios.

Los estados que puede tener un contenido son: **Privado**, el estado de privado es un estado inicial, es decir, al crear cualquier contenido, su estado de publicación será privado, este estado solo permite que esté disponible para el propietario y los administradores del sitio. Otro estado en el que se puede encontrar los contenidos es

Publicado y cuando tienen este estado los contenidos puede ser vistos por cualquier persona autenticada o no dentro del sitio, es decir, en cuestión de su estado de publicación es libre de ser consultado por todos los visitantes y usuarios del portal. **Revisión pendiente** es una estado intermedio entre los dos anteriores ya que es cuando un usuario no tiene permisos para publicar y envía el contenido a revisión para que un usuario con permisos de revisor pueda publicarlo o echarlo para atrás y/o retirarlo, a hacer esto, automáticamente se transforma en privado nuevamente. Además al estar en Revisión pendiente el contenido se vuelve visible para usuarios registrados en el sistema.

Para una visualización más clara del proceso de los estados de publicación de los contenidos véase la figura 6.1.

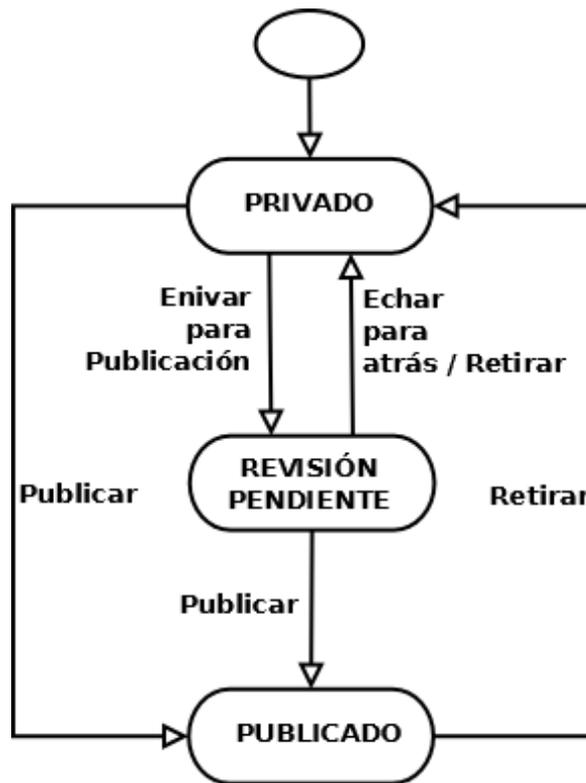


Figura 6.1: Estados de Publicación de contenidos

6.3.2 Usuarios, Grupos y Roles

En esta sección se podrán encontrar los diferentes tipos de usuarios, grupos y roles. Estas categorías son de gran utilidad para definir permisos, establecer la seguridad del

sitio y dar distintos privilegios a diferentes usuarios con características bien definidas.

Usuarios

Los usuarios dentro del sistema y bajo el portal de Plone se comportan de una manera muy sencilla, para identificarlos es posible hacerlo bajo el contexto de dos situaciones, es decir, existen dos grandes categorías, y esto es determinado por su conexión al sistema, es decir, los usuarios cuando están conectados se les puede llamar miembros del portal, de otra forma son usuarios visitantes o Anónimos, es decir usuarios sin identificación.

Para el buen manejo del sistema es muy importante saber si un usuario está conectado o no y el tipo de privilegios con los que cuenta para poder cambiar su nivel de restricción o por otro lado extender sus privilegios a fin de darle la posibilidad de explorar con mayor holgura el sistema.

Roles

Los roles son utilizados en el sistema para dar cabida a tareas mas complejas que solo ver el contenido del mismo, estos permisos son otorgados a través de privilegios, dichos privilegios son encapsulados a través de los roles. En el portal de Plone existen roles predefinidos con el motivo de ayudar a los programadores en un inicio a englobar ciertas actividades dentro del sistema. Los Roles nativos son los siguientes:

Miembro: A todo usuario que se registra o es registrado en el sistema adquiere este rol, el cual le sirve para diferenciarlo de un usuario anónimo y le brinda ciertos privilegios para el uso del contenido dentro del portal.

Contribuyente: Este rol otorga privilegios para agregar contenido al portal y por ende es propietario del mismo, pero no se le permite su publicación. Al ser dueño de este nuevo contenido tiene permisos para modificarlo o eliminarlo si así lo considera pertinente.

Editor: Como su nombre lo indica este rol puede modificar contenido propio y ajeno, es decir, sus privilegios le permiten hacer cambios a contenidos de los cuales no posee la propiedad. Como el contribuyente este rol tampoco permite el cambio de publicación del contenido.

Revisor: Los usuarios que tienen este rol puede publicar contenido, retirarlo o hacer cambio en el estado de cualquier contenido.

Lector: El rol de Lector permite a los usuarios leer la información publicada en el sitio de manera interna.

Administrador de Sitio: El rol de Administrador de sitio contiene a la vez privilegios de contribuidor, editor y revisor. Es el que tiene más privilegios dentro del portal y puede hacer cualquier tarea con un contenido, también tiene privilegios sobre las cuentas de usuario y sobre la configuración del sitio.

Cuando se registra un usuario en el sitio de manera automática se le asigna el rol de Miembro, sin embargo, los otros roles tendrá que darse a través de un administrador; este tiene el privilegio de asignarle nuevos roles a los usuarios. A un usuario dentro del sistema, se le pueden asignar uno o varios roles a fin de que pueda desempeñarse correctamente dentro del sistema. Una característica extra de los roles es que pueden ser globales o locales, es decir, su validez se extiende a todo el sitio en caso de ser globales, o bien, se puede especificar sobre un contexto o contenido dado. De manera local se puede asignar privilegios en cada contenido en la sección de “compartir”. Estos privilegios son:

Puede agregar: Esto permite añadir, editar y apropiarse de contenidos nuevos.

Puede editar: Esto permite la modificación de contenidos existentes.

Puede ver: Esto como su nombre lo indica permite observar el contenido de algo, sin permisos para añadir, modificar o eliminar.

Puede revisar: Esto da la posibilidad del cambio de estado de publicación de los contenidos.

Un ejemplo de lo anterior se puede revisar en la figura [6.2](#).

Cabe mencionar que para estos permisos locales puede existir o no, se hace uso de jerarquización, es decir, si un elemento padre tiene ciertos permisos otorgados a un conjunto de grupos y/o usuarios, estos a su vez tendrán los mismos privilegios con los contenidos hijos si es que esta característica está habilitada, en caso contrario, se detalla contenido a contenido.

Aunado a este conjunto de roles nativos, para el Sistema de Vinculación se amplió con estos 8 roles nuevos, detallados a continuación:

| Nombre | Puede agregar | Puede editar | Puede revisar | Puede ver |
|---------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| Academicos (Academico) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Administrators | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Grupo de Trabajo (GrupoTrabajo) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Laboratorios (Laboratorio) | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| Usuarios conectados | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| Site Administrators | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Figura 6.2: Privilegios locales referentes a cada contenido

Cliente: Este rol es un privilegio general que se da a los usuarios con perfil de Personas Físicas o Personas Morales y que comparten tareas o secciones dentro sus actividades.

Proveedor: Este rol es un privilegio general que se da a los usuarios con perfil de Académicos, Laboratorios o Grupos de Trabajo y que comparten tareas o secciones dentro sus actividades.

PersonaFísica: Este rol es específico para aquellos usuarios que se registren como clientes y tengan el perfil de personas físicas.

PersonaMoral: Este rol es específico para aquellos usuarios que se registren como clientes y tengan el perfil de personas morales.

Académico: Este rol es específico para aquellos usuarios que se registren como proveedores y tengan el perfil de Académico.

Laboratorio: Este rol es específico para aquellos usuarios que se registren como proveedores y tengan el perfil de Laboratorio.

GrupoDeTrabajo: Este rol es específico para aquellos usuarios que se registren como proveedores y tengan el perfil de Grupo de Trabajo.

Administrador: Este rol es específico para aquellos usuarios que se registren como proveedores y tengan el perfil de Académicos y posteriormente se les asigne un perfil adicional como Administrador.

Generalmente en el sistema implementado se dieron permisos sobre cada contenido, ya que los roles y sus actividades son muy específicas, sobre todos los contenidos a los roles de Administradores de Sitio y Administradores se dieron permisos de puede agregar, editar, revisar y ver. Para la parte libre (Inicio, ¿Quiénes Somos?, Visión, Objetivo, Ver servicios a Resolver, Ver servicios Ofrecidos, Extra, Políticas, Links de interés y Ver Proyectos) es decir, aquellas secciones que pueden ser consultadas inclusive por usuarios sin autenticación (anónimos). Se les dio permisos de ver a los grupos Académicos, Laboratorios, Grupo de Trabajo, Personas Físicas, Personas Morales y usuarios autenticados. En la sección de Menú de administradores solo Administradores y Administradores de Sitio tenían permisos y estos son ilimitados. Para la sección de Menú de Proveedores solo tiene permisos además de administradores y administradores de sitio, Laboratorios, grupos de trabajo y académicos, en distintos niveles según la subsección específica. Y por último, la sección de clientes se hace lo pertinente con personas física y morales además de los administradores y administradores de sitio.

Grupos

Los grupos son utilizados para concentrar un perfil de usuarios y trabajar sobre estas cuestiones de seguridad, permisos, accesos, privilegios y niveles de restricción, esto se puede realizar perfectamente sobre cada uno de los usuarios, pero para tener un control más eficiente y facilidad de administración se utilizan los grupos. Generalmente es a los grupos a los que se les asigna un rol. De inicio el portal tiene los siguientes grupos creados:

Administrators: Este grupo es usado para asignar el rol de Administrador.

Grupo Virtual- Usuarios Autenticados : Este grupo virtual es usado cuando se quiere discriminar entre usuarios anónimos y aquellos registrados en un momento en específico dentro del sistema. De entrada no tiene roles predefinidos.

6.3.3 Seguridad de un contenido

Por último la seguridad de un contenido; esta es tal vez la más extensa y la más interna dentro de todos los procesos que ocurren alrededor de la seguridad del CMS Plone. Dentro de cada contenido, es decir en su configuración interna, se puede encontrar en la herramienta de administración de Zope una pestaña en las opciones de configuración llamada “security”. Un ejemplo es el contenedor principal del sistema que se ubica en la raíz principal del portal, es decir, se trata del propio nombre del sistema “/Sistema de Vinculación”, en este contenido se tiene un matriz de 228 elementos que se puede controlar por Rol, se maneja como un booleano a través de un checkBox, es decir si está palomeado un rol en específico tendrá la actividad correspondiente liberada.

Así existe una matriz para cada contenido; la lista o el conjunto de acciones varía dependiente del tipo de contenido. De manera automática todos los contenidos manejan una configuración inicial de seguridad a este nivel, pero puede ser alterada para producir efectos distintos en cada rol y en cada actividad.

Este nivel es el más granular de todos, aquí se controlan actividades como crear contenidos de todo tipo, ver contenidos, hacer scripts, ejecutar archivos, cambiar configuraciones y propiedades de cualquier tipo de contenido, tener acceso a ciertos contenidos, manejar propiedades de sesión, agregar o modificar productos, manejo de proxy, ftp, cuentas de usuario, manejo de grupos, roles, es decir, todas las cosas que se pueden hacer dentro del portal puede ser manejadas en cada contenido bajo esta modalidad.

Para el Sistema de Vinculación hubo que realizar modificaciones a estas matrices de seguridad con el fin de dar permisos de ejecución, control y acceso a básicamente dos archivos muy puntuales, los scripts Python para el manejo de información y los Page Templates que soportan las acciones de dichos scripts. Los cambios mas recurrentes se hicieron bajos dos roles Anonymous y Authenticated, ya que estos brindaban la posibilidad de manejar toda la gama de los roles existentes. En las figuras 6.3 y 6.4 se muestras los cambios descritos con anterioridad.

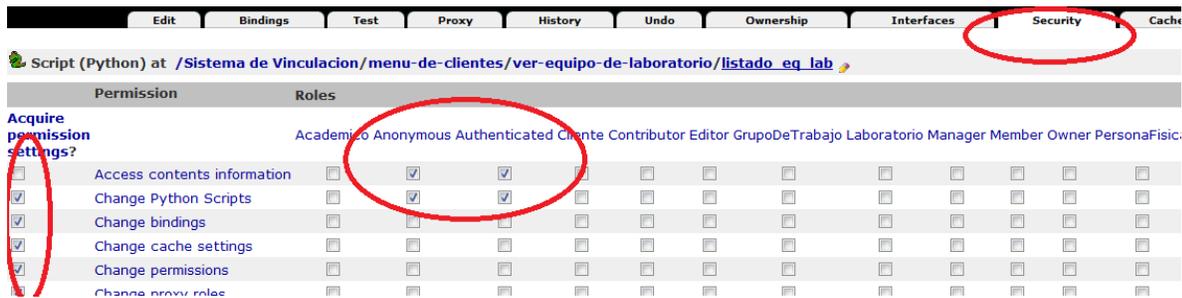


Figura 6.3: Cambios en la seguridad de cada actividad por cada rol

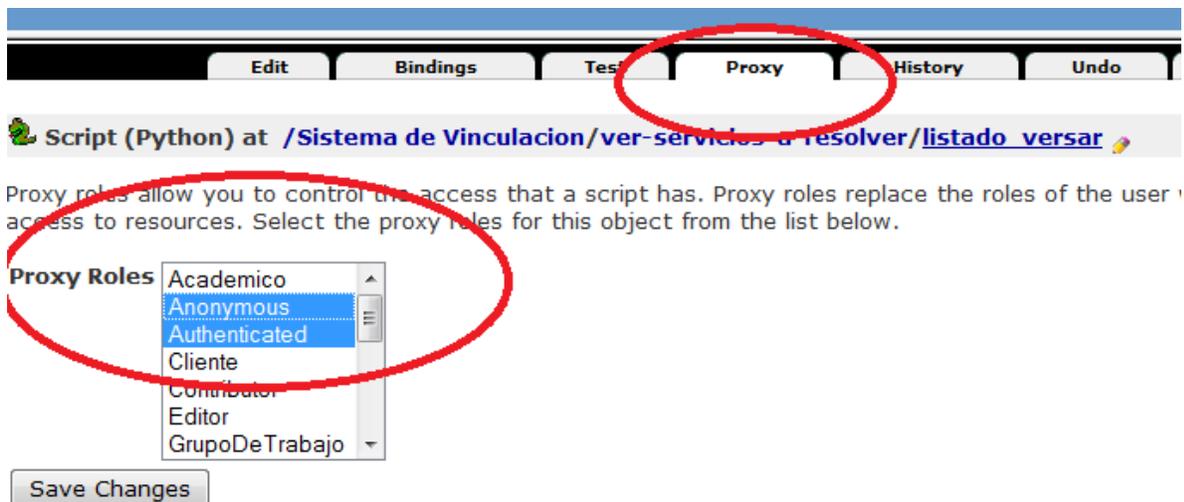


Figura 6.4: Cambios en el uso de cada contenido via proxy por cada rol

6.4 Desarrollo del sistema

Para el desarrollo del sistema en el paradigma de los gestores de contenido fue importante entender el contexto, las herramientas, y la forma de crear e interpretar la información. Esto implicó conocer y entender muchas cosas de fondo, otras fueron tratadas de manera más superficial a fin de no perder el enfoque del trabajo y avanzar en la construcción del sitio.

VENTAJAS

Los puntos que se pueden resaltar más en el desarrollo de esta aplicación bajo este entorno ya citado con anterioridad, son aquellos relacionados con el manejo y uso de toda la infraestructura de Plone, esto es, que bajo sus diferentes niveles de integración,

productos, lenguaje python, administración de Zope, la misma capa de Plone y/o la capa de la base de datos ZODB, es factible trabajar en todas ellas; inclusive hay ciertos niveles para poder desarrollar diversos productos que son útiles en el sistema. Es decir, que para poder brindar una solución no solo se puede remitir al último nivel de Plone, sino que explorando y revisando como es que funcionan estas capas se puede dar un resultado utilizando o no, una o varias capas de su estructura. Esto hace de la adaptabilidad, mantenabilidad, modificabilidad aspectos mas manejables.

Existen ciertas tareas que facilitan la creación del sistema como: validadores de campo, registro de usuarios, manejo de seguridad y varios puntos más que ya tienen un desarrollo realizado, lo cual permite emplear el tiempo en otros puntos del sistema, que tal vez requieran mayor esfuerzo debido a su complejidad. Esto habla de un gran valor de la implementación, de un modelado físico más automatizado, de la interoperabilidad y coexistencia de los productos ya construidos.

Tal vez una de las ventajas más puntuales al usar Plone para la construcción de sistemas, es saber utilizar productos ya hechos para la reutilización del software por un lado, por otro una vez montados, ser capaz de realizar los ajustes necesarios para modificar lo que viene en el producto y hacer que respondan conforme lo necesita el sistema. Esto requiere conocimiento y aprendizaje del sistema, gran portabilidad, adecuación y facilidad de adaptación.

Otro punto a resaltar durante el desarrollo es la facilidad de corrección de errores, ya que en cuanto se comete un error, la visión integral del sistema se ve comprometida y por ende el desarrollador se da cuenta de que existe un error. Este error es fácilmente buscado en una bitácora donde se catalogan cada uno de los errores cometidos, además de ayudar al desarrollo del sistema, se convierte en una herramienta personal o de equipo que resulta interesante como fuente de datos a analizar, ya que se puede medir la frecuencia de errores y catalogarlos para encontrar debilidades de construcción y desarrollo.

Que todo en Plone sea manejado como un objeto hace que el manejo de la información tenga una estructura bien definida y es difícil salirse de ello por ende da forma y unifica visiones de diferentes entidades, lo cual mejora la cohesión de los datos y eso hace a los scripts adaptables a cambios.

DESVENTAJAS

Durante el desarrollo de este sistema las desventajas que se presentaron fueron originadas por varios aspectos, uno de ellos fue la documentación existente dentro de la comunidad que utiliza Plone, ya que aunque si existe en muchas ocasiones está basada en los productos que se desean explicar y no tanto en la estructura misma de Plone, es decir, al tener un problema que se desea abordar en la documentación generalmente se asocia a un producto en el cual está inmersa la solución de ese problema de alguna manera. Este hecho afecta en tiempo y esfuerzo para llegar a un aprendizaje más profundo ya que nunca se trata de fondo la solución directa de los planteamientos, sino que al tener un contexto se da una posible solución; esto implica aprender a usar el producto, instalarlo, adecuarlo a las necesidades y este camino se vuelve un tanto tortuoso y complicado. Sin embargo, no toda la documentación es así, existen ciertos textos que si se liberan de los productos y tratan de manera directa los planteamientos de mayor demanda en el proceso de creación de un sitio web utilizando este CMS.

Otro punto es la granularidad en como el sistema en Plone debe ser montado, es decir, las capa del lenguaje de programación, la aplicación y motor Zope, la base de datos ZODB, los productos nativos y aquellos instalados posteriormente, a todos estos se les puede realizar modificaciones en las cuales sus efectos se darán al fin de cuentas en el sistema. Esto crea una desventaja ya que teniendo que realizar modificaciones en varios niveles, o en uno en específico se puede tornar difícil saber cuál será el mejor método al modificar, hasta que nivel hacerlo o cuantos niveles involucrar en el cambio. Esto se puede traducir en dos cosas; una, la curva de aprendizaje de la forma de crear sistemas en CMS es tardada y dos, tomar la decisión no solo implica tener previamente el aprendizaje de los niveles, su funcionamiento y fusión, además implicaría saber para cada decisión todas las alternativas y saber las características que las destacarían sobre las otras o bien las razones para no ser elegidas. Esto sin duda es una tarea que genera una desventaja.

Hablando sobre la forma en como se trabajan los objetos, esto es, como manejar o configurar sus relaciones con flujos de trabajo, estado de publicación, seguridad heredada y de bajo nivel, es decir, en las acciones dependiendo el tipo de objeto que se aborde; todo esto si bien es útil una vez dominado el punto, puede retrasar mucho la creación del sitio.

Capítulo 7

Plan de Pruebas y Datos

En este capítulo “plan de pruebas y datos”, se va a diseñar y construir las pruebas necesarias para medir y cuantificar las características a resaltar de ambos sistemas, con el fin de establecer un banco de datos que serán analizados posteriormente.

En este apartado lo importante es hacer un plan de pruebas que se ajuste al contexto ya antes expuesto en los sistemas realizados en las plataformas RDBMS y CMS, para ello es necesario tomar en cuenta características que no estén relacionadas directamente con la forma de trabajar con alguna de las dos plataformas, ya que de no ser así se estaría sesgando desde un inicio los resultados de estas pruebas.

Como base de las pruebas se tiene, como ya se había mencionado con anterioridad el trabajo propuesto en [17], dado que es un enfoque a la información almacenada y también tratado para sistemas con alcances medio y/o pequeño, tal y como es el caso en este trabajo.

Entonces el plan de pruebas es dividido en dos grandes secciones, una que involucra cuestiones como funcionalidad y desempeño del sistema y otra enfocada a la integridad de los datos y seguridad de los mismos. Con estas dos secciones bien definidas se podrán realizar pruebas que se enfoquen solo en estas características y darle al plan de pruebas una estructura bien definida, y con un gran soporte que permita crear uniformidad en la revisión de las plataformas, ya que estas propiedades del sistema y de la información no son derivadas a partir de un producto o de la tecnología misma, más bien, están fuertemente involucradas con el manejo propio de la información y su utilidad.

7.1 Funcionalidad y desempeño del sistema

En referencia con [17], la funcionalidad del sistema, verifica que el sistema funciona para lo que fue creado, muchas podrían ser las funciones que realiza este sistema, en esta ocasión el trabajo se delimita a probar la inserción y la consulta masiva de información, haciendo de esta prueba de funcionalidad a la vez una prueba de estrés; lo anterior se logra al tener procesos de inserción y consulta de varios miles de registros. La respuesta del sistema, es una propiedad que mide la relación entre el funcionamiento del sistema y los recursos utilizados. Esta eficiencia del sistema va relacionada con el rendimiento. La respuesta del sistema es la cantidad de tiempo que toma en hacerse una tarea, o bien, en el caso del rendimiento cuantas tareas son realizadas en un intervalo de tiempo determinado.

El desempeño del sistema es la eficiencia de los procesos realizados en el sistema los cuales administran la información almacenada en el mismo; aquí las pruebas se enfocarán en las propiedades ACID (atomicidad, durabilidad, aislamiento y consistencia) de las transacciones. A continuación se detalla cada una de ellas (de acuerdo con [15]), descartando en las pruebas únicamente Aislamiento, ya que pruebas sobre concurrencia de transacciones no serán realizadas.

Atomicidad

Esta propiedad verifica que una transacción se realiza o no, es decir, no puede quedarse a la mitad, esto garantiza preservar las propiedades intactas de la información realizando por completo o no cualquier la tarea.

Durabilidad

La durabilidad es aquella característica que se enfoca en garantizar que una vez terminada una transacción que afecte a la información almacenada, los cambios realizados a esta estarán reflejados desde el momento de terminar la transacción y no se verán afectados hasta que corra otra transacción que utilice dicha información para realizar nuevas modificaciones.

Aislamiento

El aislamiento verifica que las transacciones que corran de manera simultánea deberán valerse de los recursos de concurrencia de cada plataforma para garantizar que ambas transacciones serán tratadas de manera única y que la información sea estable durante y después de estaz.

Consistencia

La consistencia de los datos se verifica y prevalece antes y después de que una transacción afecte la información dentro de un sistema.

Las pruebas realizadas en este trabajo se enfocaron en los procesos de inserción y consulta masiva de datos.

La funcionalidad y el desempeño del sistema abarcan propiedades de la información tales como la complejidad, la funcionabilidad, la fiabilidad, la accesibilidad, la eficiencia y la oportunidad. Dichas propiedades son comprometidas si el desempeño y la funcionalidad del sistema, dada una tarea, son deficientes.

El desempeño de la respuesta del sistema será una de las pruebas a realizar en este plan, debido a que se trata de una característica que se puede cuantificar perfectamente, al tratarse precisamente del tiempo de interacción del usuario con la información almacenada, la cual es recolectada, procesada y mostrada al usuario en un cierto intervalo de tiempo desde el momento en que el usuario solicita dicha información hasta que se puede ver a través de la pantalla en los reportes ofrecidos en el sistema, o bien a través de una clase dada.

Para realizar dicho experimento se realizarón las mediciones de tres reportes distintos; el reporte de proyectos (PRO), el reporte de servicios a resolver (SAR) y el reporte de servicios ofrecidos (SO). Para cada reporte las pruebas se medieron con 3 cantidades de registros (RDBMS) o archivos-objetos (CMS) distintos: 1,000, 3,000 y 5,000, esto con el propósito de tener un banco de datos aceptable que permita inferir el comportamiento de ambas plataformas (CMS y RDBMS); dicha medición se realizó de dos modos distintos una llamada de usuario final y otra llamada de desarrollador, la de usuario final es aquella que realiza cualquier persona, es decir, un reporte dentro del sistema, con imágenes y flujos normales de la página; el modo de desarrollador tiene la característica que trae los mismos datos, es decir, la información es preservada en ambos modos, pero en esta modalidad no es un flujo del sistema común, más bien se trata de una clase “intermedia”, cuya función es extraer sólo la información como texto que pueda servir como entrada de otra clase, de esta forma existen dos mediciones, una nos dice como es el desempeño de un reporte con todo lo que conlleva presentarlo como una pantalla dentro del sistema y el otro solo se enfoca en la extracción de la información alojada en el contenedor de información. Estas 3 mediciones se repitieron

cada una de ellas cinco veces, eliminando al final las más rápida y la más lenta de ellas para contrarrestar un posible sesgo y tener 3 consultas de tiempo aceptables para cada reporte por cada plataforma. Al final se tomaron esos datos y se analizaron a través de tablas y gráficas comparativas.

Implícitamente a esta prueba está la prueba de inserción de datos masiva, la cual se hizo en los tres reportes en 3 partes, 1000 registros la primera vez, y las últimas dos veces con 2,000 registros cada una.

7.1.1 Configuración

Los experimentos fueron concebidos en un servidor dell con procesador Intel Xeon Core 4 CPU a 1.6 GHz, memoria RAM 2.0 GiB y disco duro de 129.1 GiB. Se instaló Ubuntu 10.04, Gnome 2.30.2.

Para la version RDBMS se instaló base de datos Postgres 8.4, lenguaje de programación Java, ambiente de desarrollo NetBeans 6.5 y servidor de Aplicaciones: Apache Tomcat.

Para la version CMS se instaló ambiente de desarrollo Plone 4.1.0.9, dentro del cual existe: Repositorio de Objetos ZODB, lenguaje de programación Python 2.6.7, servidor de aplicaciones Zope 2.13.8 y además se instalaron dos productos básicos para el desarrollo y funcionamiento adecuados: PloneFormGen 1.7.0 y PloneFormGen Save Data to Content 1.3.1. Además para subir datos masivos se instaló el producto Atreal.Massloader.

7.1.2 Plataforma RDBMS

Para la inserción masiva de los datos se utilizaron plantillas en hojas de cálculo las cuales se transformaron en scripts sql, los cuales fueron gradualmente insertados de 1,000 registros en una primera fase y de 2,000 en las dos subsecuentes. Los registros están compuestos de la siguiente manera:

Proyecto - PROY

insert into proyecto (nombre,fechainicio,costo,estatusproyecto,numeroregistro,objetivo,

idusuario_alta,idusuario_modificacion,fecha_alta,fecha_modificacion, eliminada)
 values ('proyecto1','2012-03-25','200000',1,'1','objetivo',6,6,'2012-03-25','2012-03-25',0);

Donde los campos 'nombre' y 'numeroregistro' son los únicos que cambian entre todos los registros al aumentar un número consecutivo, esto debido a las restricciones mismas planteadas en la base de datos. De esta manera en cada registro se forma un identificador idproyecto, el cual también un número consecutivo.

El tamaño de estos campos es el siguiente.

idproyecto INT(11)
 fechainicio DATE
 costo DECIMAL(9,2)
 estatusproyecto DECIMAL(1,0)
 numeroregistro VARCHAR(45)
 objetivo TEXT
 idusuario_alta INT(11)
 idusuario_modificacion INT(11)
 fecha_alta DATE
 fecha_modificacion DATE
 eliminada DECIMAL (1,0)

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 8 figuras las cuales se detallan en la tabla 7.1.

Tabla 7.1: Características de las imágenes del reporte PROY - RDBMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|-----------|----------------------|
| lupa.png | 764 bytes | 33 x 27 |
| modifica.png | 1 Kb | 20 x 20 |
| aporta.png | 6 Kb | 50 x 50 |
| calc.png | 3 Kb | 50 x 50 |
| estrella.png | 4 Kb | 32 x 32 |
| entrega.png | 2 Kb | 48 x 48 |
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Servicio a Resolver - SAR

insert into servicioaresolver (nombre,descripcion,idusuario,fechaalta)
 values ('servicioaresolver1','5','2012-03-20');

Donde el campo ‘nombre’ es el único que cambian entre todos los registros al aumentar un número consecutivo, esto debido a las restricciones mismas planteadas en la base de datos. De esta manera en cada registro se forma un identificador idservicioaresolver, el cual también un número consecutivo.

El tamaño de estos campos es el siguiente.

idservicioaresolver INT(11)
 nombre VARCHAR(100)
 descripcion TEXT
 idusuario INT(11)
 fechaalta DATE

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 2 figuras las cuales se detallan en la tabla 7.2.

Tabla 7.2: Características de las imágenes del reporte SAR - RDBMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|--------|----------------------|
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Servicio Ofrecido - SO

```
insert into servicioofrecido(nombre,descripcion,proveedor_idusuario,fechaalta,eliminada)
values ('servicioofrecido1','',6,'2012-03-21',0);
```

Donde el campo ‘nombre’ es el único que cambian entre todos los registros al aumentar un número consecutivo, esto debido a las restricciones mismas planteadas en la base de datos. De esta manera en cada registro se forma un identificador idservicioofrecido, el cual también un número consecutivo.

El tamaño de estos campos es el siguiente.

idservicioofrecido INT(11)
 nombre VARCHAR(100)
 descripcion TEXT
 proveedor_idusuario INT(11)
 fechaalta DATE

eliminada DECIMAL(1,0)

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 2 figuras las cuales se detallan en la tabla 7.3.

Tabla 7.3: Características de las imágenes del reporte SO - RDBMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|--------|----------------------|
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Los datos sobre el desempeño de la inserción en esta plataforma pueden ser consultados en la tabla 7.4.

Cota superior

Cabe mencionar que el primer objetivo de esta prueba era medir el estrés del funcionamiento y desempeño con archivos que contuvieran registros de cientos de miles de datos hasta llegar a la cifra de 1,000,000 de registros, sin embargo, para la plataforma RDBMS se presentó un problema, no en el proceso de inserción la cual puede soportar esos datos y más, sin embargo, para la parte de consulta la plataforma se quedó sin respuesta de su servidor de aplicaciones, esto debido a los recursos mismos mencionados en la configuración 7.1.1. Se realizaron varias pruebas para encontrar una cota superior que fuera un margen para continuar con las pruebas, las cifras que fueron probadas y que fallaron durante los experimentos fueron : 250000, 50000, 30000, 20000 y por último 10000 registros, estas pruebas se ejecutaron para la consulta del reporte de proyectos. La figura 7.1 muestra la respuesta del sistema

Tabla 7.4: Tiempos (ms) para la inserción masiva en la plataforma RDBMS

| Entidad | 1,000 registros | 2,000 registros - 3,000 en total | 2,000 registros - 5,000 en total |
|----------------------|-----------------|----------------------------------|----------------------------------|
| Servicios a resolver | 435 | 387 | 390 |
| Servicios ofrecidos | 214 | 427 | 417 |
| Proyectos | 343 | 613 | 583 |

Ahora bien, para las pruebas de usuario final se mostrarán los datos de las 5 pruebas para las 3 cantidades de registros, sobre las tres consultas de reportes. Dichos datos, se pueden consultar en la tabla 7.5. Los datos con la marca † son eliminados

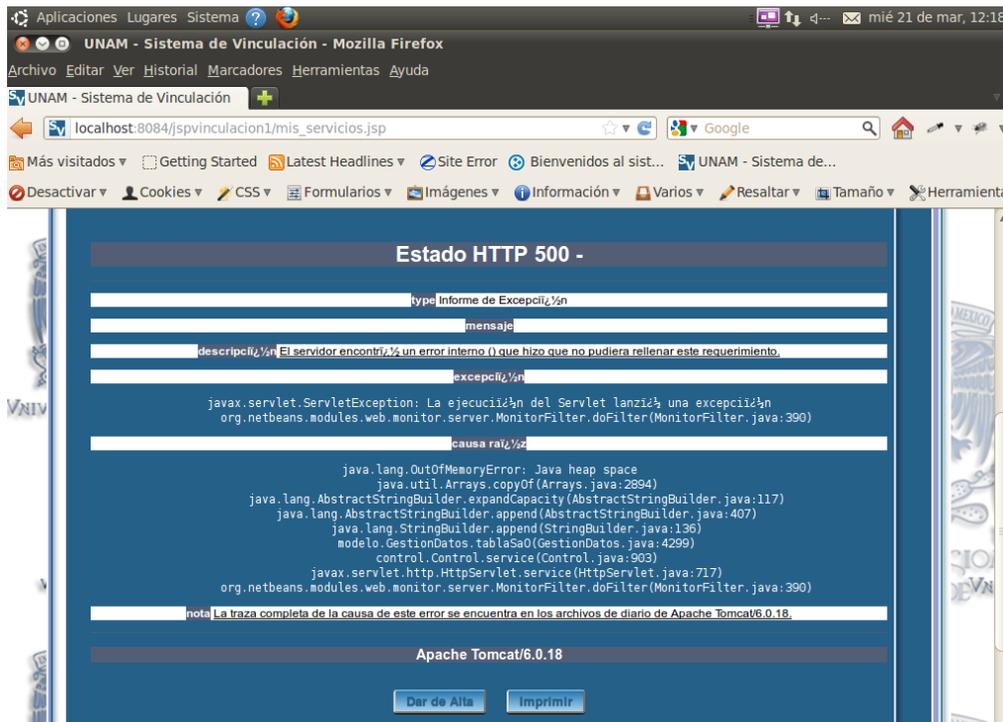


Figura 7.1: Cota superior en proceso de consulta de proyectos

para el promedio por ser el dato mas alto y el mas bajo de los 5 intentos.

Tabla 7.5: Tiempos (ms) para todas las pruebas en la plataforma RDBMS.

| 1000 registros | | | | | | |
|----------------------|----------|----------|----------|----------|----------|----------------------|
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 2281 | 2317 | † 2319 | 2252 | † 2236 | 2283.3333333333 |
| Servicios ofrecidos | † 1991 | 2013 | 2012 | 2009 | † 2061 | 2011.3333333333 |
| Proyectos | † 20775 | 20433 | 20476 | 20394 | † 20375 | 20434.3333333333 |
| 3000 registros | | | | | | |
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | † 26716 | 26365 | 26393 | 26414 | † 26363 | 26390.6666666667 |
| Servicios ofrecidos | † 23055 | † 22752 | 22842 | 22754 | 22772 | 22789.3333333333 |
| Proyectos | † 243118 | 242986 | 242762 | † 234926 | 235048 | 240265.3333333333 |
| 5000 registros | | | | | | |
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 86334 | † 87078 | 85968 | 83946 | † 83930 | 85416 |
| Servicios ofrecidos | † 72735 | † 72835 | 72795 | 72764 | 72759 | 72772.6666666667 |
| Proyectos | † 878357 | 775814 | † 713463 | 714211 | 713556 | 734527 |

Por último, se muestran los resultados de las pruebas de desarrollador, los datos de las 5 pruebas para las 3 cantidades de registros, sobre las 3 consultas de reportes.

Dichos datos, se pueden consultar en la tabla 7.6. Los datos con la marca † son eliminados para el promedio por ser el dato mas alto y el mas bajo de los 5 intentos.

Tabla 7.6: Tiempos (ms) para todas las pruebas en la plataforma RDBMS.

| 1000 registros | | | | | | |
|----------------------|----------|----------|----------|----------|----------|----------------------|
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 113 | † 110 | † 120 | 116 | 113 | 114 |
| Servicios ofrecidos | 133 | † 141 | 128 | † 116 | 120 | 127 |
| Proyectos | † 160 | 130 | 106 | 127 | † 105 | 121 |
| 3000 registros | | | | | | |
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 912 | 907 | † 906 | † 924 | 921 | 913.3334 |
| Servicios ofrecidos | 883 | 883 | † 882 | 882 | † 894 | 882.6667 |
| Proyectos | † 856 | 723 | 692 | † 673 | 686 | 700.3334 |
| 5000 registros | | | | | | |
| Entidad | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 3276 | † 4199 | 2596 | 2527 | † 2519 | 2799.6667 |
| Servicios ofrecidos | † 2694 | 2527 | 2466 | 2459 | † 2414 | 2484 |
| Proyectos | † 1911 | 1908 | 1839 | † 1837 | 1838 | 1861 |

A continuación se describe el proceso para cada consulta para el proceso que el usuario final sigue. Esto con el fin de detallar los procesos de consulta y tener más claro el nivel de desempeño medido.

Proyecto (PRO)

El proceso para la extracción de esta consulta una vez oprimido el botón de Proyectos en el menú de administradores (color rosa), es que inicia con la carga de la página proyectos.jsp; ahí por medio de AJAX se pasa el manejo de la descarga al control por medio de la petición “control?operacion=tablaProyecto” esa opción inicia el contador de tiempo y a su vez manda llamar al proceso de gestión de datos por medio de “dts.tablaProyecto(sesion.getAttribute(‘stipousuario’).toString(), sesion.getAttribute(‘sidusuario’).toString());” este proceso envía las variables de sesión tipo de usuario e identificador de usuario. Una vez en el archivo GestionDatos, se extrae la información, con dos tipos de consultas. La consulta principal es

```
“SELECT * FROM Proyecto WHERE eliminada=0 ORDER BY nombre;”
```

de la cual se van extrayendo la mayoría de los datos y la siguiente se utiliza para las características posteriores del proyecto como aportaciones, pagos, documentos, etc. Esta segunda consulta que está contenido dentro de la primera está estructurada de la siguiente manera:

```

queryValido = "SELECT idresponsable as idusu FROM Proyecto WHERE idproyecto="
+rs.getString('idproyecto')
+ " UNION "
+"SELECT idresponsable AS idusu FROM entregable WHERE eliminada=0 AND id-
proyecto=" +
rs.getString('idproyecto')
+ " UNION "
+"SELECT idusuario AS idusu FROM pagosfunciones WHERE eliminada=0 AND
idproyecto=" +
rs.getString('idproyecto')
+ " UNION "
+"SELECT idusuario AS idusu FROM servicioaresolver WHERE eliminada=0 AND
idservicioaresolver="+rs.getString('idservicioaresolver');

```

Después el proceso se encarga de ir armando la tabla con las imágenes, datos y referencias necesarias, para llevarlas de regreso hasta la vista

Servicio a Resolver (SAR)

El proceso para la extracción de los servicios a resolver es como primer paso dar click al botón (verde) del menú de clientes “Mis Servicios a Resolver” la cual redirige a la página misproblemas.jsp, una vez hecho esto, como el proceso del proyecto, pasa el control a la clase de java “control.java” a través de la instrucción solicitada con AJAX “ ‘control?operacion=tablaSaRidusuario='+idusuario7” y ahí se inicia el cronometro de cuanto tiempo tarda en ejecutar la petición y con la opción dts.tablaSaR(idusuario); cede el control al proceso de gestión de datos. Este proceso envía la variable de identificador de usuario. Dentro de GestionDatos, se conecta a la Base de datos y se extrae la información a través de la siguiente consulta:

```

query = “ ‘ SELECT * FROM ServicioaResolver WHERE eliminada=0 and idusuario=’
+idusuario”;

```

Aquí se extrae la información, se llena la tabla y se pasa el control de la ejecución a la clase control donde se detiene el reloj y se envía la información a la vista.

Servicio Ofrecido (SO)

Lo mismo que las consultas anteriores se tiene que iniciar solicitando la página `misservicios.jsp` a través del botón (naranja) Mis Servicios Ofrecidos del menú de proveedores, dentro de la vista por medio de AJAX se pasa el manejo de la descarga al control por medio de la petición “`control?operacion=tablaSaOidusuario=+idusuario7`”, una vez en el control, se inicia el cronometro y a su vez manda llamar al proceso de gestión de datos por medio de “`dts.tablaSaO(idusuario)`” este proceso envía la variable de identificador de usuario. Una vez en la clase `GestionDatos`, se extrae la información, con la siguiente consulta.

```
“ ‘ SELECT * FROM ServicioOfrecido WHERE eliminada=0 and proveedoridusuario=’  
+idusuario;”
```

Aquí se obtiene la información, se llena la tabla y se pasa el control de la ejecución a la clase control donde se detiene el reloj y se envía la información a la vista.

Las consultas para el tipo de usuario desarrollador se describen a continuación:

Proyecto (PRO)

Una vez iniciado sesión el usuario puede consultar dicha clase con la ruta “`localhost:8084/jspvinculacion1/control?operacion=cuantos`” esa opción inicia el contador de tiempo y a su vez llama al proceso de gestión de datos por medio de “`dts.cuantos('6')`” este proceso envía el `idusuario 6` que es la inserción de master en las tabla proyecto. Una vez en el archivo `GestionDatos`, se extrae la información, con dos tipos de consultas. La consulta principal es

```
“SELECT * FROM Proyecto WHERE eliminada=0 ORDER BY nombre;”
```

Después se imprime un contador para verificar el número de registros y el nombre del proceso.

Servicio a Resolver (SAR)

Después de ingresar un usuario y contraseña al sistema se puede ir directamente a esta ruta “`localhost:8084/jspvinculacion1/control?operacion=cuantossar`” una

vez ahí, se inicia el contador y redirige al proceso de gestión de datos a través de “dts.cuantossar('5')”. Este proceso envía el identificador de usuario 5, que es el cliente prueba con el que se ligaron los datos de inserción. Ya dentro de GestionDatos, se conecta a la Base de datos y se extrae la información a través de la siguiente consulta:

```
query = “SELECT * FROM ServicioaResolver WHERE eliminada=0 ORDER BY nombre;”
```

Después el imprime un contador para verificar el número de registros y el nombre del proceso.

Servicio Ofrecido (SO)

Lo mismo que las consultas anteriores una vez ingresado un usuario válido, se consulta la ruta: ”localhost:8084/jspvinculacion1/control?operacion=cuantoso” una vez aquí, se inicia el contador y redirige al proceso de gestión de datos a través de “dts.cuantoso('6')”. Este proceso envía el identificador de usuario 6, que es el usuario con el que se ligaron los datos de inserción. Ya dentro de GestionDatos, se conecta a la Base de datos y se extrae la información a través de la siguiente consulta:

```
“SELECT * FROM ServicioOfrecido WHERE eliminada=0 ORDER BY nombre;”
```

Después el imprime un contador para verificar el número de registros y el nombre del proceso.

7.1.3 Plataforma CMS

Para la inserción masiva de los datos se utilizaron archivos zip con objetos con propiedades adecuadas para cada consulta, los cuales fueron gradualmente insertados de 1,000 registros en una primera fase y de 2,000 en las próximas dos. Los objetos están compuestos de la siguiente manera:

Proyecto - PROY

El objeto es tomado de la estructura del siguiente archivo.

url:<http://localhost:8080/Sistema de Vinculacion/menu-de-administradores/formulario>

proyectos/proyectos/proyecto1
meta_type:FormSaveData2ContentEntry
title:
content_type:text/plain
auth:Basic YWRtaW46MFZUNnZob3A= cookie:fontsize=
tree-s="eJzTyCkw5NLIKTDiClZ3hANXW3WuAmOuRKCECUjWFFn
WCSxrxpUIBHoAAMYp5A"
id: proyecto1
subject:
relatedItems:
location:
language: es-mx
effectiveDate: 2012/03/28 19:55:00 US/Central
expirationDate: None
creation_date : 2012/03/28 19 : 56 : 16.128987US/Central
modification_date : 2012/03/28 20 : 43 : 49.511385US/Central
creators : admin
contributors :
rights :
allowDiscussion : False
excludeFromNav : False
nombre : proyecto1
descripcion : descripcionproyecto1
fecha – inicio : 2012/03/28 00 : 00 : 00US/Central
fecha – fin : 2014/11/17 00 : 00 : 00US/Central
costo – total : 200000
estatus : Atrasado
numero – de – registro : 1
objetivo : nada
responsable : usuPA
servicioresolver : 2012 – 01 – 25.3596214614
Content – Type : text/plain

Donde los atributos ‘url’, ‘id’, ‘nombre’, ‘descripcion’ y ‘numero-de-registro’ son los atributos que cambian entre todos los objetos al aumentar un número consecutivo esto debido a las restricciones de identificación de cada objeto.

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 8 figuras las cuales se detallan en la tabla 7.7.

Tabla 7.7: Características de las imágenes del reporte PROY - CMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|-----------|----------------------|
| lupa.png | 764 bytes | 33 x 27 |
| modifica.png | 1 Kb | 20 x 20 |
| aporta.png | 6 Kb | 50 x 50 |
| calc.png | 3 Kb | 50 x 50 |
| estrella.png | 4 Kb | 32 x 32 |
| entrega.png | 2 Kb | 48 x 48 |
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Servicio a Resolver - SAR

url:http://localhost:8080/Sistema de Vinculacion/menu-de-clientes/formulario mis-servicios-a-resolver/servicio-a-resolver/servicioaresolver1

meta_type : *FormSaveData2ContentEntry*

title :

content_type : *text/plain*

auth : *BasicYW RtaW46MFZUNnZob3A =*

cookie : *fontsize =; tree-s = "eJzTyCkw5NLIKTDiClZ3hANXW3WuAmOuRK CECUjWFFnWCSxr xpUIBHoAAMP5A"*

id : *servicioaresolver1*

subject :

relatedItems :

location :

language : *es - mx*

effectiveDate : *2012/03/2021 : 10 : 00US/Central*

expirationDate : *None*

creation_date : *2012/03/2021 : 09 : 19.661667US/Central*

modification_date : *2012/03/2020 : 36 : 50.127812US/Central*

creators : *usuCPF*

contributors :

rights :

allowDiscussion : *False*

excludeFromNav : *False*

nombre : *Servicioaresolver1general*

descripcion : descripciondelservicioaresolver1general
presupuestoempresarial : 200000
campodeconocimiento : 2012 – 01 – 17.0510586965
 2012 – 01 – 17.0642818711
 2012 – 01 – 17.0810135190
 2012 – 01 – 17.0969298851

Content – Type : text/plain

Donde los atributos ‘url’, ‘id’, ‘nombre’ y ‘descripcion’ son los atributos que cambian entre todos los objetos al aumentar un número consecutivo esto debido a las restricciones de identificación de cada objeto.

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 2 figuras las cuales se detallan en la tabla 7.8.

Tabla 7.8: Características de las imágenes del reporte SAR - CMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|--------|----------------------|
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Servicio Ofrecido - SO

url:http://localhost:8080/Sistema de Vinculacion/menu-de-proveedores/formulario mis-servicios-ofrecidos/servicios-ofrecidos/servicioofrecido1

meta_type : *FormSaveData2ContentEntry*

title :

content_type : *text/plain*

auth : *BasicYWRtaW46MFZUNnZob3A=*

cookie : *fontsize =; tree-s = "eJzTyCkw5NLIKTDiClZ3hANXW3WuAmOuRKCECUjWFFnWCSxrxpUIBHoAAMP5A"*

id : *servicioofrecido1*

subject :

relatedItems :

location :

language : *es – mx*

effectiveDate : *2012/03/2015 : 30 : 00US/Central*

expirationDate : None
creation_date : 2012/03/2015 : 29 : 50.991275US/Central
modification_date : 2012/03/2021 : 09 : 44.280115US/Central
creators : admin
contributors :
rights :
allowDiscussion : False
excludeFromNav : False
nombre : Servicioofrecidogeneral1
descripcion : descripcionservicioofrecidogeneral1
campodeconocimiento : 2012 – 01 – 17.0510586965
 2012 – 01 – 17.0642818711
 2012 – 01 – 17.0969298851

Content – Type : text/plain

Donde los atributos ‘url’, ‘id’, ‘nombre’ y ‘descripcion’ son los atributos que cambian entre todos los objetos al aumentar un número consecutivo esto debido a las restricciones de identificación de cada objeto.

Además en el modo usuario final, es decir, cuando se sigue el flujo normal del sistema, a cada carga de registro es necesario agregar 2 figuras las cuales se detallan en la tabla 7.9.

Tabla 7.9: Características de las imágenes del reporte SO - CMS

| Nombre | Tamaño | Dimensión en pixeles |
|---------------|--------|----------------------|
| modificar.png | 1 Kb | 20 x 20 |
| eliminar.png | 2 Kb | 24 x 23 |

Los datos sobre la inserción en esta plataforma pueden ser consultados en la tabla 7.10.

Cota superior

Cabe mencionar que el primer objetivo de esta prueba era medir como en la plataforma

RDBMS, el estrés del funcionamiento y desempeño con archivos que contuvieran cientos de miles de objetos hasta llegar a la cifra de 1,000,000 de ellos, sin embargo, con la prueba de la cota superior en la plataforma RDBMS se probaron los mismos datos, es decir, se probó llegar a una cota superior hasta que el sistema quedó sin respuesta de su servidor de aplicaciones, esto debido a las recursos mismos mencionados en la configuración 7.1.1. Se realizaron varias pruebas para encontrar dicha cota, las cifras que fueron probadas y que fallaron durante los experimentos fueron : 250000, 50000. Para 30000 el reporte de proyectos tardó aproximadamente de 30 a 40 minutos en responder, sin embargo, tenía que tomarse los mismos parámetros que con la plataforma RDBMS. La figura 7.2 muestra la respuesta del sistema con la inserción de 50,000 objetos de servicios ofrecidos.

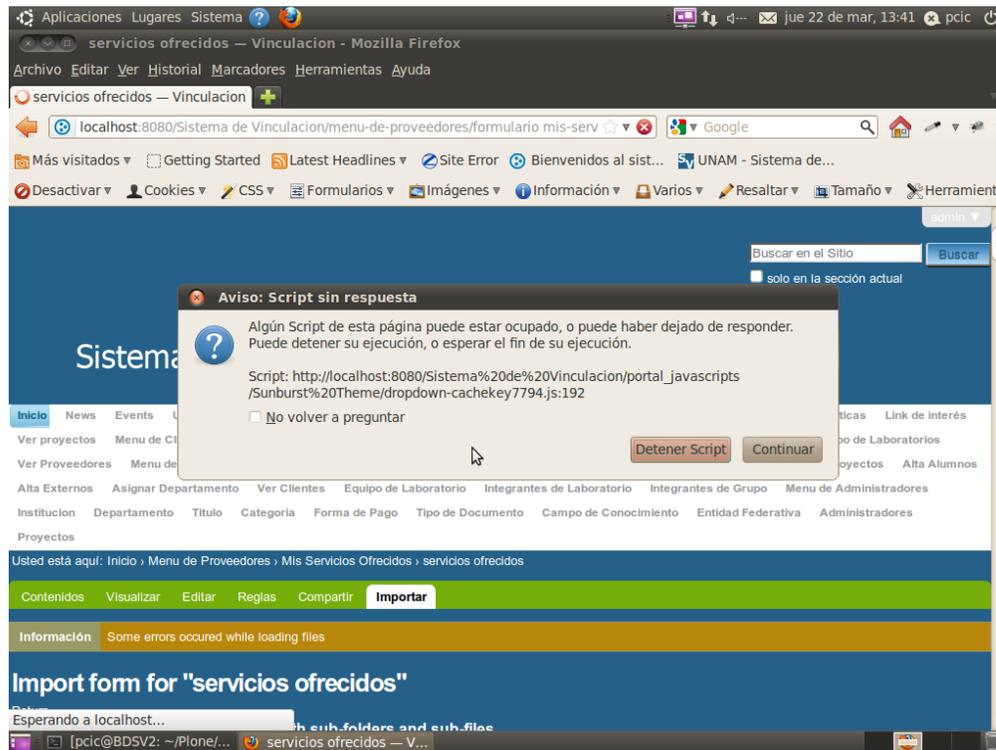


Figura 7.2: Cota superior en proceso de inserción de servicios ofrecidos

Tabla 7.10: Tiempos (ms) para la inserción masiva en la plataforma CMS

| Objeto | 1,000 archivos | 2,000 archivos - 3,000 en total | 2,000 archivos - 5,000 en total |
|----------------------|----------------|---------------------------------|---------------------------------|
| Servicios a resolver | 103812 | 307432 | 605338 |
| Servicios ofrecidos | 123674 | 203343 | 550491 |
| Proyectos | 183019 | 793267 | 819928 |

Ahora bien, para las pruebas de usuario final en el sistema, se mostrarán los datos de las 5 pruebas para las 3 cantidades de registros, sobre las tres consultas. Dichos datos, se pueden consultar en la tabla 7.11. Los datos con la marca † son eliminados para el promedio por ser el dato mas alto y el mas bajo de los 5 intentos.

Tabla 7.11: Tiempos (ms) para todas las pruebas en la plataforma CMS.

| 1000 archivos | | | | | | |
|----------------------|----------|----------|----------|----------|----------|----------------------|
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | † 30391 | 29049 | 29389 | † 28941 | 29412 | 29283.3333333333 |
| Servicios ofrecidos | 26609 | † 25607 | † 27073 | 25724 | 26196 | 26176.3333333333 |
| Proyectos | † 7640 | 7622 | 7609 | † 7534 | 7610 | 7613.6666666667 |
| 3000 archivos | | | | | | |
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | † 90330 | 88375 | † 88318 | 88758 | 88464 | 88532.3333333333 |
| Servicios ofrecidos | † 80819 | 78800 | 78646 | 79708 | † 78508 | 79051.3333333333 |
| Proyectos | † 24540 | 22656 | 22609 | † 22572 | 22700 | 22655 |
| 5000 archivos | | | | | | |
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 151200 | † 301824 | † 149534 | 149792 | 150092 | 150361.3333333333 |
| Servicios ofrecidos | † 134530 | 132925 | 133438 | † 132578 | 132907 | 133090 |
| Proyectos | † 40690 | 39198 | † 38132 | 38220 | 38956 | 38791.3333333333 |

De la misma forma para las pruebas de modo desarrollador, se mostrarán los datos de las 5 pruebas para las 3 cantidades de registros, sobre las tres consultas. Dichos datos, se pueden consultar en la tabla 7.12. Los datos con la marca † son eliminados para el promedio por ser el dato mas alto y el mas bajo de los 5 intentos.

Tabla 7.12: Tiempos (ms) para todas las pruebas en la plataforma CMS.

| 1000 archivos | | | | | | |
|----------------------|----------|----------|----------|----------|----------|----------------------|
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 20017 | † 20334 | † 19957 | 20295 | 19969 | 20093.6667 |
| Servicios ofrecidos | † 20721 | 19618 | 19733 | 19670 | † 19597 | 19673.6667 |
| Proyectos | † 3839 | 2944 | 2940 | † 2945 | 2944 | 2942.6667 |
| 3000 archivos | | | | | | |
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | 60489 | 60550 | 60435 | † 60083 | † 60644 | 60491.3334 |
| Servicios ofrecidos | † 61908 | 59358 | † 59226 | 59284 | 59785 | 59475.6667 |
| Proyectos | † 10674 | 10291 | 10176 | † 10786 | 10216 | 10227.6667 |
| 5000 archivos | | | | | | |
| Objeto | Prueba 1 | Prueba 2 | Prueba 3 | Prueba 4 | Prueba 5 | Promedio (3 pruebas) |
| Servicios a resolver | † 95549 | 93056 | 93271 | 93437 | † 92476 | 93254.6667 |
| Servicios ofrecidos | 101806 | † 102398 | † 101073 | 102078 | 101316 | 101733.3334 |
| Proyectos | † 18051 | 14741 | 14851 | † 14723 | 14748 | 14780 |

Para las consultas para el modo usuario final, se utilizaron los siguientes procedimientos (script Python):

Proyecto (PRO)

Para dar inicio al reporte de proyectos, el usuario debe localizarse en la sección de Proyectos, dentro del menú de administradores y dar click en el botón “Administrar Proyectos”. Al hacer eso, el sistema, manda llamar a un Page Template de nombre AdminsitrarProyectos y dentro de este se llama un script de Python “Administrar proyectos” ahí se empieza a medir el tiempo a través de `DateTime().PreciseTime()`. En ese mismo script se manda a llamar a los objetos necesarios para obtener los datos necesarios, a continuación se detalla este llamado de objetos:

```
1.. path="/Sistema de Vinculacion/menú-de-administradores/formulario proyectos/"
2.. pathobj=context.restrictedTraverse(path)
3.. document = getattr(pathobj, 'proyectos')
...
4.. for content in document.objectValues():
...
```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 sirve para establecer la ruta absoluta de donde está la carpeta padre de la información. La línea 2 convierte esta ruta en objeto para que se puedan manipular sus atributos, que a su vez pueden volver a ser objetos. La línea 3 recupera los objetos proyectos. y los devuelve en un arreglo llamado document. La línea 4 recorre todos los elementos de document con el fin de barrer todos los objetos que resultaron de la extracción.

Servicio a Resolver (SAR)

Cuando un cliente está en el menú de clientes, dentro del formulario de servicios a resolver, puede dar click en el botón “Listado Mis servicios a Resolver”. Al hacer eso, el sistema, manda llamar a un Page Template de nombre “ListadoMisSerRes” y dentro de este se llama un script de Python “listadomsare”, donde se empieza a medir el tiempo a través de `DateTime().PreciseTime()`. En el script se manda a llamar a los objetos necesarios para obtener los datos requeridos, a continuación se detalla este llamado de objetos:

```
1... pm = getToolByName(context, "portalmembership")
2... usuconectado=pm.getAuthenticatedMember()
...
3... path="/Sistema de Vinculacion/menú-de-clientes/formulario mis-servicios-a-resolver/"
```

```

4... pathobj=context.restrictedTraverse(path)
5... document = getattr(pathobj, 'servicio-a-resolver')
6... for content in document.objectValues():
7...     auxstr=content.Creators()
8...     for cadena in auxstr:
9...         if str(usuconectado) == str(cadena):

```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 se conecta con el portal donde existe información sobre los usuarios registrados (portalmembership), en la línea dos a través del método `getAuthenticatedMember()` obtiene que usuario está conectado al momento de hacer la petición. Las líneas 3, 4 y 5 son como la consulta anterior sirven para establecer la ruta, ubicar el objeto padre y recoger los objetos y atributos necesarios. La línea 6 recorre los resultados, pero como no se quieren todos sino aquellos que sean propiedad del usuario conectado en cada objeto se pregunta por el `Creator` (líneas 7,8 y 9), de esta manera se garantiza que al ser igual que el usuario conectado se muestre el objeto.

Servicio Ofrecido (SO)

Cuando un Proveedor: Laboratorio, grupo de trabajo o académico está en el menú de proveedores dentro del formulario de servicios ofrecidos, puede dar click en el botón “Listado Mis servicios Ofrecidos”. Al hacer eso, el sistema, manda llamar a un Page Template de nombre “ListadoMisSerOf” y dentro de este se llama un script de Python “listadomso” ahí se empieza a medir el tiempo a través de `DateTime().PreciseTime()`. En el script se manda a llamar a los objetos necesarios para obtener los datos requeridos, a continuación se detalla este llamado de objetos:

```

1... pm = getToolByName(context, "portalmembership")
2... usuconectado=pm.getAuthenticatedMember()
...
3... path="/Sistema de Vinculacion/menú-de-proveedores/formulario mis-servicios-
ofrecidos/"
4... pathobj=context.restrictedTraverse(path)
5... document = getattr(pathobj, 'servicios-ofrecidos')
...
6... path2="/Sistema de Vinculacion/menú-de-proveedores/agregar-solucion/agregar-
solucion/"
7... pathobj2=context.restrictedTraverse(path2)

```

```
8... sol = getattr(pathobj2, 'solucion')
...
9... for content in document.objectValues():
10... auxstr=content.Creators()
11... for cadena in auxstr:
12... if str(usuconectado) == str(cadena):

...
13... for content in sol.objectValues():
14... auxstr=content.Creators()
15... for cadena in auxstr:
16... if str(usuconectado) == str(cadena):
```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 se conecta con el portal donde existe información sobre los usuarios registrados (portalmembership), en la línea 2 a través del método `getAuthenticatedMember()` obtiene que usuario está conectado al momento de hacer la petición. Los paquetes de líneas 3, 4 y 5 por un lado y por otro las líneas 6, 7 y 8 sirven para establecer la ruta, ubicar el objeto padre y recoger los objetos y atributos necesarios, el primero utilizado para servicios que no son solución, y los segundos para ubicar a las que son soluciones. Las líneas del 9 al 12 y del 13 al 16 recorren los resultados, pero como no se quieren todos sino aquellos que sean propiedad del usuario conectado en cada objeto se pregunta por el `Creator`, de esta manera se garantiza que al ser igual que el usuario conectado se muestre el objeto.

Para las consultas para el modo desarrollador, se utilizaron los siguientes procedimientos (script Python):

Proyecto (PRO)

Para dar inicio al reporte de proyectos, el desarrollador debe localizarse en la sección de Proyectos, dentro del menú de administradores y consultar el script Python “cuantos”. Al hacer eso, el sistema empieza a correr el script, donde se empieza a medir el tiempo a través de `DateTime().PreciseTime()`. Ahí se manda a llamar a los objetos necesarios para obtener los datos necesarios, a continuación se detalla este llamado de objetos:

```
1.. path="/Sistema de Vinculacion/menú-de-administradores/formulario proyectos/"
```

```

2.. pathobj=context.restrictedTraverse(path)
3.. document = getattr(pathobj, 'proyectos')
...
4.. for content in document.objectValues():
...

```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 sirve para establecer la ruta absoluta de donde está la carpeta padre de la información. La línea 2 convierte esta ruta en objeto para que se puedan manipular sus atributos, que a su vez pueden volver a ser objetos. La línea 3 recupera los objetos proyectos. y los devuelve en un arreglo llamado document. La línea 4 recorre todos los elementos de document con el fin de barrer todos los objetos que resultaron de la extracción.

Servicio a Resolver (SAR)

Cuando un desarrollador está en el menú de clientes dentro del formulario de servicios a resolver, puede consultar un script de Python “cuantosar” donde se empieza a medir el tiempo a través de DateTime().PreciseTime(). En el script se manda a llamar a los objetos necesarios para obtener los datos necesarios, a continuación se detalla este llamado de objetos:

```

1... pm = getToolByName(context, “portalmembership”)
2... usuconectado=pm.getAuthenticatedMember()
...
3... path=“/Sistema de Vinculacion/menú-de-clientes/formulario mis-servicios-a-resolver/”
4... pathobj=context.restrictedTraverse(path)
5... document = getattr(pathobj, 'servicio-a-resolver')
6... for content in document.objectValues():
7... auxstr=content.Creators()
8... for cadena in auxstr:
9... if str(usuconectado) == str(cadena):

```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 se conecta con el portal donde existe información sobre los usuarios registrados (portalmembership), en la línea dos a través del método getAuthenticatedMember() obtiene que usuario esta conectado al momento e hacer la petición. Las líneas 3, 4 y 5 son como la consulta anterior sirven para establecer la ruta, ubicar

el objeto padre y recoger los objetos y atributos necesarios. La línea 6 recorre los resultados, pero como no se quieren todos sino aquellos que sean propiedad del usuario conectado en cada objeto se pregunta por el Creator, de esta manera se garantiza que al ser igual que el usuario conectado se muestre el objeto.

Servicio Ofrecido (SO)

Cuando un desarrollador está en el menú de proveedores dentro del formulario de servicios ofrecidos, puede consultar un script de Python `cuantosos.py`, ahí se empieza a medir el tiempo a través de `DateTime().PreciseTime()`. En el script se manda a llamar a los objetos necesarios para obtener los datos necesarios, a continuación se detalla este llamado de objetos:

```
1... pm = getToolByName(context, "portalmembership")
2... usuconectado=pm.getAuthenticatedMember()
...
3... path="/Sistema de Vinculacion/menú-de-proveedores/formulario mis-servicios-
ofrecidos/"
4... pathobj=context.restrictedTraverse(path)
5... document = getattr(pathobj, 'servicios-ofrecidos')
...
6... path2="/Sistema de Vinculacion/menú-de-proveedores/agregar-solucion/agregar-
solucion/"
7... pathobj2=context.restrictedTraverse(path2)
8... sol = getattr(pathobj2, 'solucion')
...
9... for content in document.objectValues():
10... auxstr=content.Creators()
11... for cadena in auxstr:
12... if str(usuconectado) == str(cadena):
...
13... for content in sol.objectValues():
14... auxstr=content.Creators()
15... for cadena in auxstr:
16... if str(usuconectado) == str(cadena):
```

Con estas líneas se puede extraer la información de los objetos de ZODB desde Python y Plone. La línea 1 se conecta con el portal donde existe información sobre los

usuarios registrados (portal membership), en la línea 2 a través del método `getAuthenticatedMember()` obtiene que usuario está conectado al momento e hacer la petición. Lo paquetes de líneas 3, 4 y 5 por un lado y por otro las líneas 6, 7 y 8 sirven para establecer la ruta, ubicar el objeto padre y recoger los objetos y atributos necesarios, el primero utilizado para servicios que no son solución, y los segundos para ubicar a las que son soluciones. Las líneas del 9 al 12 y del 13 al 16 recorren los resultados, pero como no se quieren todos sino aquellos que sean propiedad del usuario conectado en cada objeto se pregunta por el `Creator`, de esta manera se garantiza que al ser igual que el usuario conectado se muestre el objeto.

7.2 Integridad y seguridad de la información

La integridad de la información es una propiedad que es deseable que tenga toda la información almacenada en un sistema, esto debido a que dicha propiedad brinda la certeza y seguridad a los usuarios de que la información es real y correcta, que contiene datos oportunos y cuyo valor no ha sido decrecido por el manejo mismo de la información sino todo lo contrario. Como se menciona en [17], la integridad puede ser entendida como la garantía de un tratamiento sin errores de aquellas funciones y/o procesos que administran y controlan los datos. La integridad de la información se puede separar en varios aspectos y se puede clasificar de la siguiente manera:

Integridad de dominio

Es aquella integridad que se refiere a que los datos pertenezcan a un conjunto de valores en específico (dominio), reglas o restricciones a la información que entra al sistema, estas permiten establecer un dominio bien delimitado. Los atributos de los objetos/registros de ambas plataformas deben cumplir con estas características.

Integridad referencial

Esta integridad puntualiza que cierta información que va a almacenarse en el sistema puede estar relacionada con datos previamente insertados en el sistema, pero que no pertenecen a una misma entidad, solo se relacionan entre ellos. Para la plataforma CMS se refiere a la relación entre objetos y para la plataforma RDBMS concretamente a información contenida como llave foránea dentro de una tabla la cual hace referencia a otra.

Integridad de entidad

La integridad de entidad se puede entender como cuidar la información almace-

nada para una entidad; es decir, para un objeto o tabla en particular, debe reflejar un tipo de información en específico, por ejemplo para este sistema, un usuario, un servicio a resolver, un proyecto, un equipo de laboratorio son entidades bien definidas. Debe ser factible extraer cualquier información e identificarla de manera única, esto se logra a través de dos formas, en la plataforma CMS con la propiedad de Id(), y para la plataforma RDBMS es la llave primaria, los campos, atributos o información que componen una o la otra deben reflejar de manera única la información contenida y por tanto no puede ser nulos, repetidos o atributos carentes de información.

Integridad definida por el usuario

Son reglas donde se cuidan los requerimientos del usuario a través de transacciones, bloqueos y aquellos procesos que van sumados a otros procesos de tratamiento de información habitual del sistema.

La integridad de la información va ligada a propiedades como la seguridad, la oportunidad, la libertad de errores, la representación concisa, conocimiento del dominio, cohesión de los datos, credibilidad, efectividad y relevancia. Si la integridad de la información se ve comprometida en alguno de sus aspectos, estas características de la información se pueden ver vulneradas. Para evitar ello se debe tener una certeza sobre la calidad de la integridad de la información en un sistema.

Como se mencionó anteriormente la integridad va ligada de cierta forma con la seguridad de la información, esto debido a que mecanismos aplicados en diferentes niveles de las capas de cada plataforma brinda seguridad, estos mecanismos hacen que la información sea respaldada adecuadamente y con la certeza de que solo ciertos tipo de usuarios podrá tener permisos específicos sobre ella. El tema de la seguridad es extenso, y se puede hablar de seguridad de la información, del sistema, sobre el ingreso y cierre de sesión, seguridad de protección de la estructura, sobre ataques externos y/o internos, problemas de inyección de información falsa, problemas en la red. Sin embargo para delimitar el tema de seguridad en este trabajo se enfocará a la seguridad como aquella característica que protege la información de ser consultada, almacenada, modificada y/o eliminada por diferentes tipos de usuarios.

Por un lado, esta delimitación permitió enfocarse en la integridad de la información junto con un nivel de seguridad de uso de la misma, y complementando estas pruebas la verificación en la inserción, eliminación y modificación de datos, la garantía de la preservación de las propiedades ACID: atomicidad, durabilidad y consistencia, dejando sin probar por tratarse de un problema de concurrencia la propiedad de ais-

lamiento.

La configuración queda exactamente igual que en el apartado de configuración de desempeño, 7.1.1. En general las pruebas de integridad se van a realizar en los primeros tres tipos de integridad: de dominio, de entidad y referencial. Ya que no hubo ninguna regla o restricción definida por el usuario. Por otro lado dichas pruebas se realizaron con varios roles y/o tipos de usuario con el fin de probar la seguridad implementada en cada plataforma de manera más amplia.

7.2.1 Plataforma RDBMS

Con la intención de describir mejor el tipo de pruebas a realizar se presentan las tablas 7.13, 7.14, 7.15 y 7.16, las cuales incluyen columnas de tipos de usuario y los renglones son las acciones realizadas, y en la intersección están los lugares del sitio donde se realizó la prueba.

Tabla 7.13: Matriz 1 de Pruebas de integridad y seguridad para proveedores

| Acción | Administrador 1 | Académico 1 | Laboratorio 1 | Grupo de Trabajo 1 |
|----------------|----------------------------|---------------------------|-----------------------------|---------------------------|
| Inserción 1 | Instituto - Departamento * | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Eliminación 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Modificación 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Consulta 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Inserción 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio * | Integrante de Laboratorio |
| Eliminación 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Modificación 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Consulta 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Inserción 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio * | Equipo de laboratorio |
| Eliminación 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Modificación 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Consulta 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Inserción 4 | Integrante de grupo | Integrante de grupo | Integrante de grupo | Integrante de grupo * |
| Eliminación 4 | Integrante de grupo | Integrante de grupo | Integrante de grupo | Integrante de grupo |
| Modificación 4 | Integrante de grupo | Integrante de grupo | Integrante de grupo | Integrante de grupo |
| Consulta 4 | Integrante de grupo | Datos personales | Datos personales | Datos personales |
| Inserción 5 | Datos personales | Datos personales | Datos personales | Datos personales |
| Eliminación 5 | Datos personales | Datos personales | Datos personales | Datos personales |
| Modificación 5 | Datos personales | Datos personales | Datos personales | Datos personales |
| Consulta 5 | Datos personales | Datos personales | Datos personales | Datos personales |

Nota: El integrante de Laboratorio será un Alumno, y el integrante de Grupo será un externo

Para realizar estas pruebas se dieron de alta los usuarios que se muestran en la tabla 7.17 con el fin de usarlos para las pruebas antes detalladas.

Tabla 7.14: Matriz 1 de Pruebas de integridad y seguridad para clientes

| Acción | Persona física 1 | Persona moral 1 |
|----------------|---------------------------|---------------------------|
| Inserción 1 | Instituto - Departamento | Instituto - Departamento |
| Eliminación 1 | Instituto - Departamento | Instituto - Departamento |
| Modificación 1 | Instituto - Departamento | Instituto - Departamento |
| Consulta 1 | Instituto - Departamento | Instituto - Departamento |
| Inserción 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Eliminación 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Modificación 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Consulta 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Inserción 3 | Equipo de laboratorio | Equipo de laboratorio |
| Eliminación 3 | Equipo de laboratorio | Equipo de laboratorio |
| Modificación 3 | Equipo de laboratorio | Equipo de laboratorio |
| Consulta 3 | Equipo de laboratorio | Equipo de laboratorio |
| Inserción 4 | Integrante de grupo | Integrante de grupo |
| Eliminación 4 | Integrante de grupo | Integrante de grupo |
| Modificación 4 | Integrante de grupo | Integrante de grupo |
| Consulta 4 | Integrante de grupo | Datos personales |
| Inserción 5 | Datos personales * | Datos personales |
| Eliminación 5 | Datos personales | Datos personales |
| Modificación 5 | Datos personales | Datos personales |
| Consulta 5 | Datos personales | Datos personales |

Nota: El integrante de Laboratorio será un Alumno, y el integrante de Grupo será un externo.

Tabla 7.15: Matriz 2 de Pruebas de integridad y seguridad para proveedores

| Acción | Administrador 2 | Académico 2 | Laboratorio 2 | Grupo de Trabajo 2 |
|----------------|---------------------------|---------------------------|---------------------------|---------------------------|
| Eliminación 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Modificación 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Consulta 1 | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento | Instituto - Departamento |
| Inserción 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Eliminación 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Modificación 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Consulta 2 | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio | Integrante de Laboratorio |
| Eliminación 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Modificación 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Consulta 3 | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio | Equipo de laboratorio |
| Eliminación 4 | Integrante de grupo | Integrante de grupo | Integrante de grupo | Integrante de grupo |
| Modificación 4 | Integrante de grupo | Integrante de grupo | Integrante de grupo | Integrante de grupo |
| Consulta 4 | Integrante de grupo | Datos personales | Datos personales | Datos personales |
| Eliminación 5 | Datos personales | Datos personales | Datos personales | Datos personales |
| Modificación 5 | Datos personales | Datos personales | Datos personales | Datos personales |
| Consulta 5 | Datos personales | Datos personales | Datos personales | Datos personales |

Nota: El integrante de Laboratorio será un Alumno, y el integrante de Grupo será un externo

7.2.2 Plataforma CMS

Con la intención de describir mejor el tipo de pruebas a realizar se presentan las tablas [7.13](#), [7.14](#), [7.15](#) y [7.16](#), las cuales incluyen columnas de roles de usuario y los renglones son las acciones realizadas, y en la intersección están los lugares del sitio donde se realizó la prueba.

Para realizar estas pruebas se dieron de alta los usuarios de la tabla [7.18](#) con el fin de usarlos para las pruebas antes detalladas.

Tabla 7.16: Matriz 2 de Pruebas de integridad y seguridad para clientes

| Acción | Persona física 2 | Persona moral 2 |
|----------------|---------------------------|---------------------------|
| Eliminación 1 | Instituto - Departamento | Instituto - Departamento |
| Modificación 1 | Instituto - Departamento | Instituto - Departamento |
| Consulta 1 | Instituto - Departamento | Instituto - Departamento |
| Eliminación 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Modificación 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Consulta 2 | Integrante de Laboratorio | Integrante de Laboratorio |
| Eliminación 3 | Equipo de laboratorio | Equipo de laboratorio |
| Modificación 3 | Equipo de laboratorio | Equipo de laboratorio |
| Consulta 3 | Equipo de laboratorio | Equipo de laboratorio |
| Eliminación 4 | Integrante de grupo | Integrante de grupo |
| Modificación 4 | Integrante de grupo | Integrante de grupo |
| Consulta 4 | Integrante de grupo | Datos personales |
| Eliminación 5 | Datos personales | Datos personales |
| Modificación 5 | Datos personales | Datos personales |
| Consulta 5 | Datos personales | Datos personales |

Nota: El integrante de Laboratorio será un Alumno, y el integrante de Grupo será un externo.

Tabla 7.17: Usuarios utilizados en las pruebas de la plataforma RDBMS

| Usuario | Password | Tipo de Usuario |
|-----------|----------|------------------|
| usuAca1 | 12345678 | Académico |
| usuLab1 | 12345678 | Laboratorio |
| usuGpo1 | 12345678 | Grupo de Trabajo |
| usuPF1 | 12345678 | Persona Física |
| usuPM1 | 12345678 | Persona Moral |
| usuAdmon1 | 12345678 | Administrador |
| usuAca2 | 12345678 | Académico |
| usuLab2 | 12345678 | Laboratorio |
| usuGpo2 | 12345678 | Grupo de Trabajo |
| usuPF2 | 12345678 | Persona Física |
| usuPM2 | 12345678 | Persona Moral |
| usuAdmon2 | 12345678 | Administrador |

Tabla 7.18: Usuarios utilizados en las pruebas de la plataforma CMS

| Usuario | Password | Grupo de Usuario |
|-----------|----------|------------------|
| usuAca1 | 12345678 | Académico |
| usuLab1 | 12345678 | Laboratorio |
| usuGpo1 | 12345678 | Grupo de Trabajo |
| usuPF1 | 12345678 | Persona Física |
| usuPM1 | 12345678 | Persona Moral |
| usuAdmon1 | 12345678 | Administrador |
| usuAca2 | 12345678 | Académico |
| usuLab2 | 12345678 | Laboratorio |
| usuGpo2 | 12345678 | Grupo de Trabajo |
| usuPF2 | 12345678 | Persona Física |
| usuPM2 | 12345678 | Persona Moral |
| usuAdmon2 | 12345678 | Administrador |

Capítulo 8

Análisis de Resultados

Con los datos y resultados arrojados en el capítulo anterior se llevó a cabo en base a las características de los sistemas, un análisis concreto de cada una de las pruebas realizadas, en este capítulo se presentarán en forma comparativa y por separado todos los resultados que pongan en evidencia el funcionamiento y procesos del sistema tanto en la plataforma RDBMS como en la plataforma CMS.

8.1 Desempeño de la respuesta del sistema

Como se mencionó en el capítulo anterior de las pruebas dedicadas a las propiedades ACID se midieron y analizaron las propiedades de durabilidad, atomicidad y consistencia, dejando por involucrar concurrencia a la propiedad de aislamiento. Para estas pruebas se tomaron en cuenta los experimentos hechos con las cotas superiores, es decir, con los procesos que no funcionaron y los experimentos con las inserciones masivas. A continuación se puntualizan cada una de las propiedades para las dos plataformas.

Atomicidad

- ▷ CMS. Para esta plataforma la propiedad de Atomicidad se cumplió en todos los intentos fallidos de inserción al no realizar ningún cambio parcial en la base de datos, por otro lado se realizaron completos y de manera satisfactoria todos las inserciones positivas de 1000, y 2000 objetos realizados.

- ▷ RDBMS. Analizando la atomicidad en la plataforma RDBMS se encontró que la propiedad se verificó en todos los intentos fallidos de inserción al no realizar ningún cambio parcial en la base de datos y por otro lado, se realizaron completos

y de manera satisfactoria todos las inserciones positivas de 1000, y 2000 registros realizados.

Durabilidad

- ▷ CMS. El paradigma de los CMS pasó satisfactoriamente la revisión de durabilidad, ya que no importando si fue realizada o no la inserción masiva, la durabilidad se presentó en todo momento dejando la base de datos con la información correspondiente.
- ▷ RDBMS. Para todas las pruebas la plataforma RDBMS respondió positivamente y aunque hubo pruebas fallidas y otras completadas en su totalidad, siempre los datos fueron preservados sin importar el resultado del proceso.

Consistencia

- ▷ CMS. En un estado consistente transcurrieron cada una de las pruebas de inserción masiva para este paradigma.
- ▷ RDBMS. No se encontró incidentes respecto a la consistencia de la información en ninguna de las pruebas realizadas.

8.2 Funcionalidad del sistema

Cabe aclarar que en ambos modos (usuario final y desarrollador) y para ambas plataformas (CMS y RDBMS) con 1000, 3000 y 5000 registros la funcionalidad se verificó, es decir, que todo funciona correctamente para lo que fue contruido, por lo que sólo se detallaron los tiempos de respuesta obtenidos. Antes de presentar las gráficas de resultados, se pone la selección de color para cada prueba a fin de identificarla y poder hacer distinción de entre otras pruebas, dicha relación se puede consultar en la tabla [A.13](#) ubicada en los apéndice A.

Gráficas para modo Usuario Final

Primeramente se mostrarán las gráficas que resultan necesarias para el análisis detallado. La gráfica [8.1](#) muestra las variaciones de tiempo de los 3 reportes en sus

5 mediciones para la plataforma CMS (Plone). Como se puede observar, los datos en 1000, 3000 y 5000 registros para PRO (Proyectos) son mas bajos, y los de SO (Servicios ofrecidos) y SAR (Servicios a Resolver) son un tanto mas parecidos entre ellos. Esto indica que la forma de extracción entre los dos servicios y el proyecto, es considerable. La existencia de 5 pruebas y el proceso de eliminar la más rápida y la más lenta no se justifica mejor que en el caso de los 5000 registros para el caso de SAR-2. En este experimento en específico se dispara de un promedio de 150361 ms a casi el doble, con una respuesta de 301824 ms. Fuera de este caso, las mediciones en general son muy parecidas.

Analizando la gráfica 8.2, la cual es la misma en condiciones a la anterior, pero distinta plataforma, se puede ver que igual que en la otra los reportes de SO y SAR muestran similitudes en tiempo, la que sigue siendo distinta es PRO (Proyectos), pero ahora en vez de tardarse menos, como con la plataforma CMS, aquí su demora es mucho mayor.

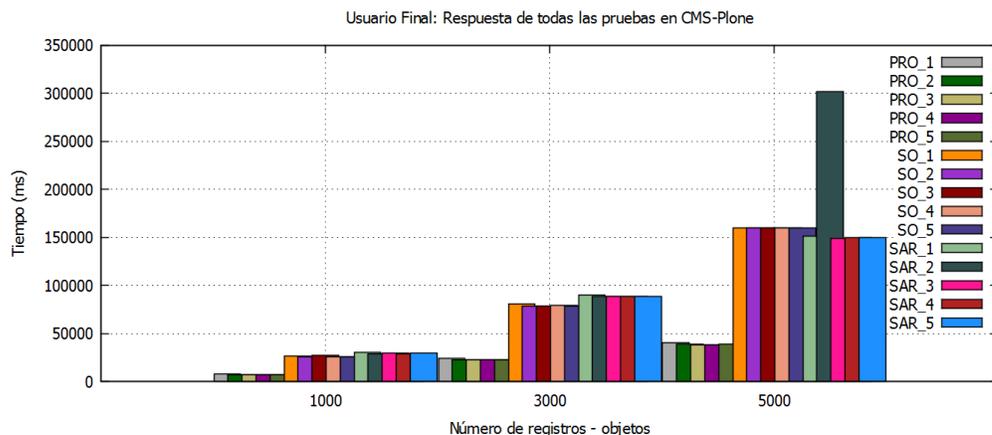


Figura 8.1: Usuario Final: Todas las pruebas en CMS

En las gráficas 8.3 de Servicios a Resolver y 8.4 de Servicios Ofrecidos, se puede ver que en los reportes de SO y SAR existe una amplia diferencia entre las repuestas de los sistemas, siendo el de RDBMS mejor en desempeño, ya que responde en los eventos de 1000, 3000 y 5000 más rápido que la plataforma CMS.

En la gráfica 8.5 vemos a diferencia de las dos anteriores, que aquí la plataforma CMS responde mucho mejor que su similar RDBMS, esto podría sonar contradictorio, ya que siguen siendo reportes similares y las condiciones no han cambiado, pero lo distinto es que mientras en los reportes anteriores se manejaban solo 2 imágenes por

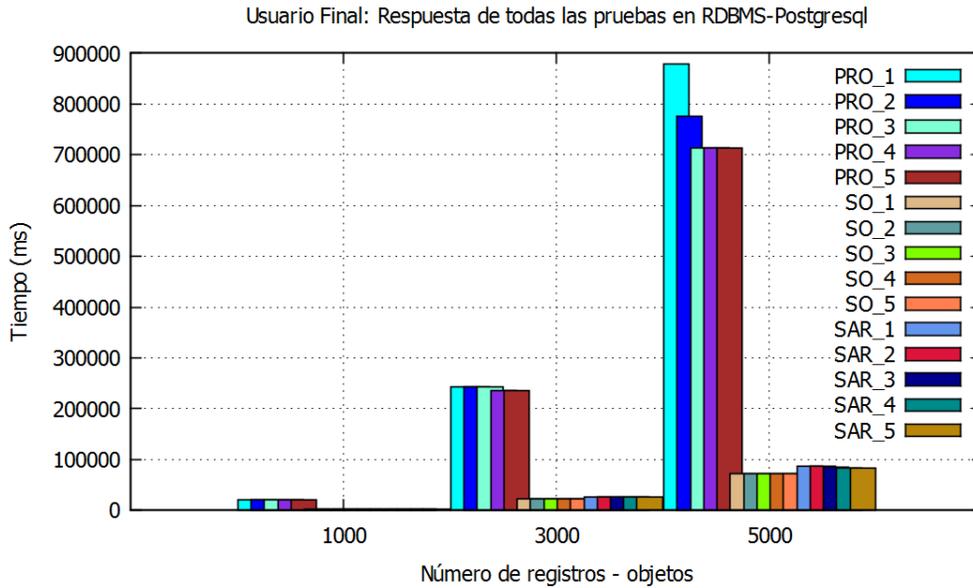


Figura 8.2: Usuario Final: Todas las pruebas en RDBMS

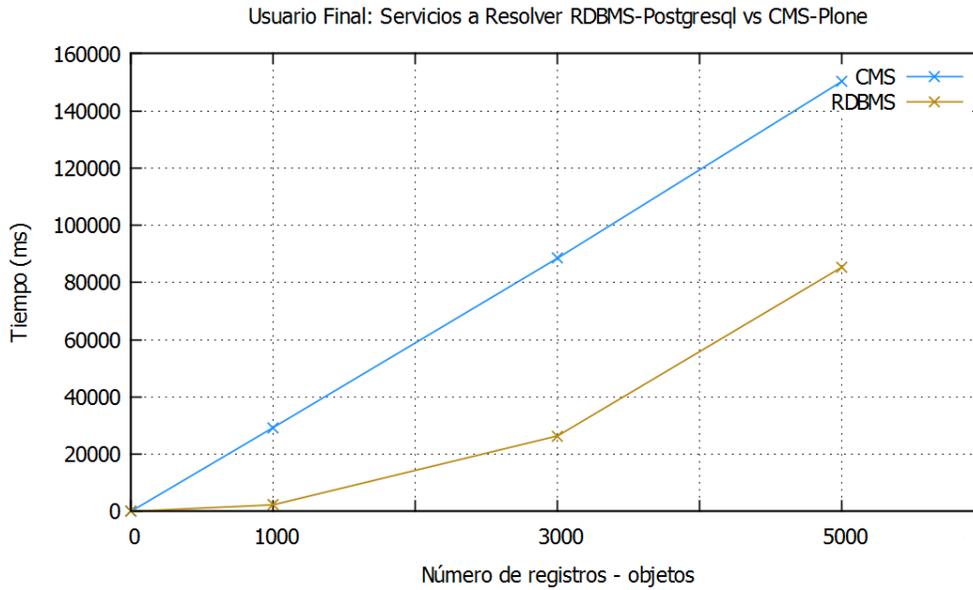


Figura 8.3: Usuario Final: SAR RDBMS vs CMS

registro, en proyectos se imprimen 8 imágenes por renglón haciendo más que un problema de volcado de texto, un problema de manejo de imágenes, ya que es una consulta que tiene su punto más crucial en el manejo de estas. Por esta razón se decidió realizar las mismas mediciones desde el punto de vista de un desarrollador, haciendo de la consulta una clase intermedia que pudiera reflejar el comportamiento de las plataformas

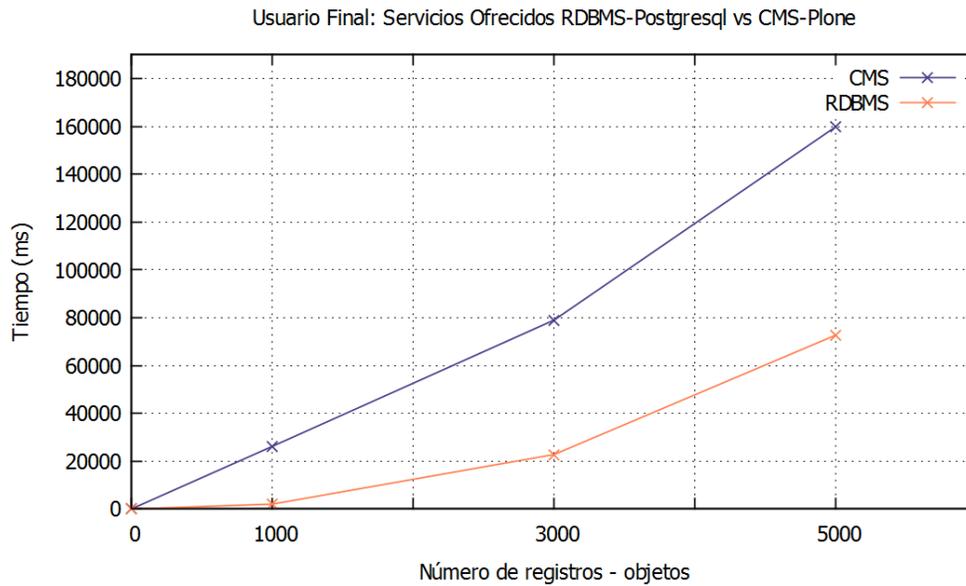


Figura 8.4: Usuario Final: SO RDBMS vs CMS

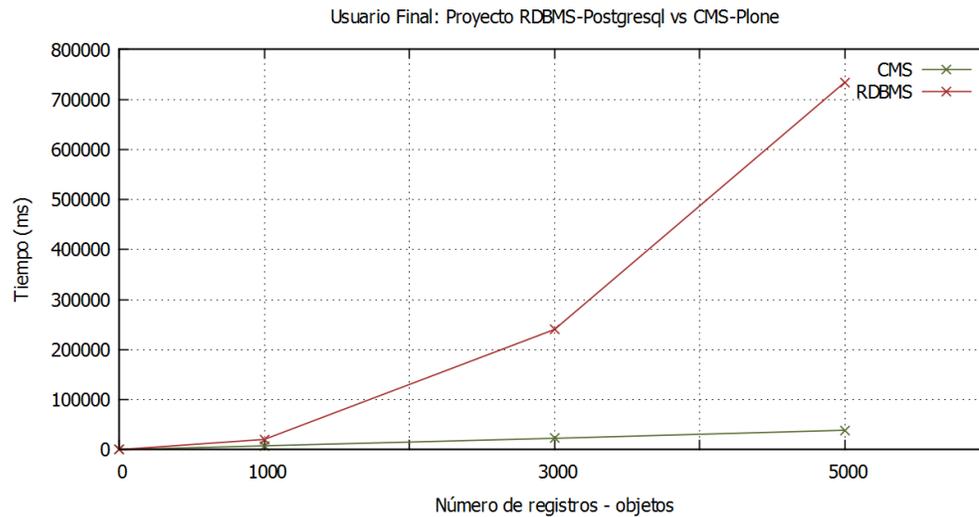


Figura 8.5: Usuario Final: PROY RDBMS vs CMS

sin la intervención y uso del manejo de imágenes.

Para terminar la parte de los experimentos de usuario final, se tiene la gráfica 8.6, la cual es una vista general de todos los experimentos, aquí se observa como las pruebas de SO y SAR realizados en la plataforma RDBMS son los más rápidos, y las pruebas SO y SAR de la plataforma CMS son más lentos, dejando a un lado las repuestas del reporte de PRO (proyectos), ya que en realidad estas son al revés, es decir, la más rápida

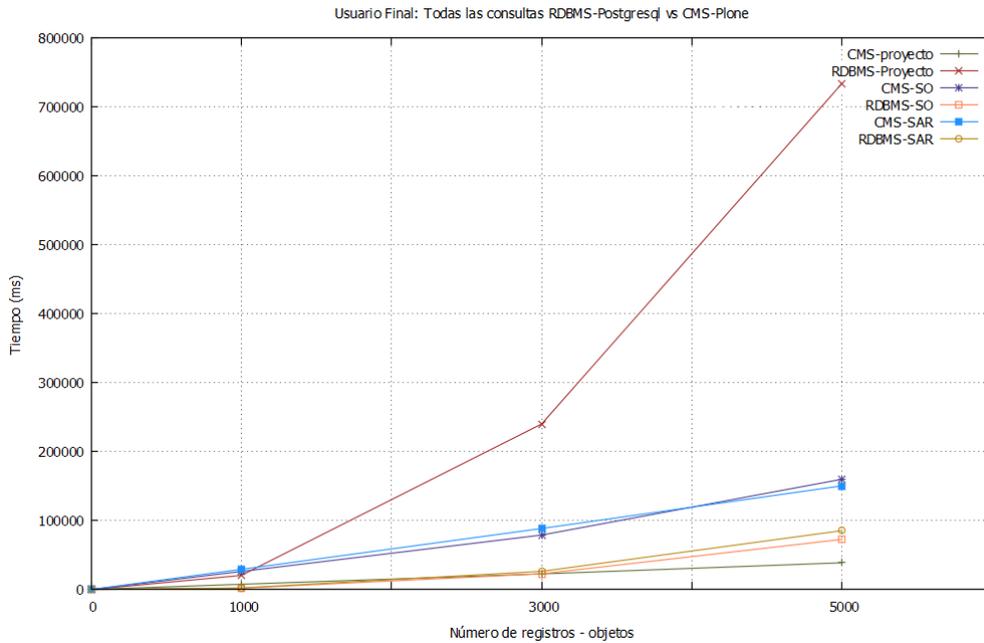


Figura 8.6: Usuario Final: Todos las consultas RDBMS vs CMS

(CMS) y la más lenta (RDBMS), estas últimas son también la más rápida y la más lenta de todo el experimento, respectivamente. Aquí lo interesante es observar como el manejo de imágenes hacen un buen desempeño para la plataforma CMS y en cambio para la plataforma RDBMS hace verse como una plataforma de desempeño menor.

Por ese motivo a continuación se observarán los resultados de las pruebas de desempeño en modo desarrollador y finalmente se podrán visualizar todos los resultados para un mejor entendimiento del experimento.

Gráficas para modo Desarrollador

Del mismo modo que se explicaron las gráficas del modo usuario final, se llevará a cabo una reseña con los mismos puntos para el modo desarrollador.

La gráfica 8.7 contiene el resumen de todas las pruebas hechas en la plataforma CMS, aquí se observa como los tres reportes en cada una de sus 5 repeticiones son muy similares entre sí, es decir, no hay grandes saltos de tiempo en sus repeticiones y se ve como en casos anteriores que los reportes de Servicios Ofrecidos (SO) y Servicio a Resolver (SAR) son entre ellos también similares solo dejando al reporte de proyecto (PRO) como el reporte más rápido ya que se observa una respuesta más rápida que

los dos anteriormente mencionados. También se puede notar que el desempeño de la plataforma es proporcional al aumento de 1000, 3000 y 5000 registros.

Ahora bien, en la gráfica 8.8 es un tanto similar el comportamiento, ya que bajo esta plataforma (RDBMS) la uniformidad también se ve en todas las repeticiones a excepción de algunos casos como las repeticiones 1 y 2 de servicios a resolver y el primero de servicio ofrecido, sin embargo, no es algo que genere una alerta o una atención mayor, ya que son eliminadas por estar fuera del rango promedio para tomar las mediciones promedio. En esta prueba la plataforma RDBMS, con sus reportes SO, SAR y PRO se manejan a diferencia de la plataforma CMS con mayor parecido en sus tiempos, es decir, aquí no existe diferencia tan significativa entre el reporte de proyectos y sus otros dos semejantes; aunado a esto, el tiempo de respuesta es definitivamente mucho más rápido que con la otra plataforma.

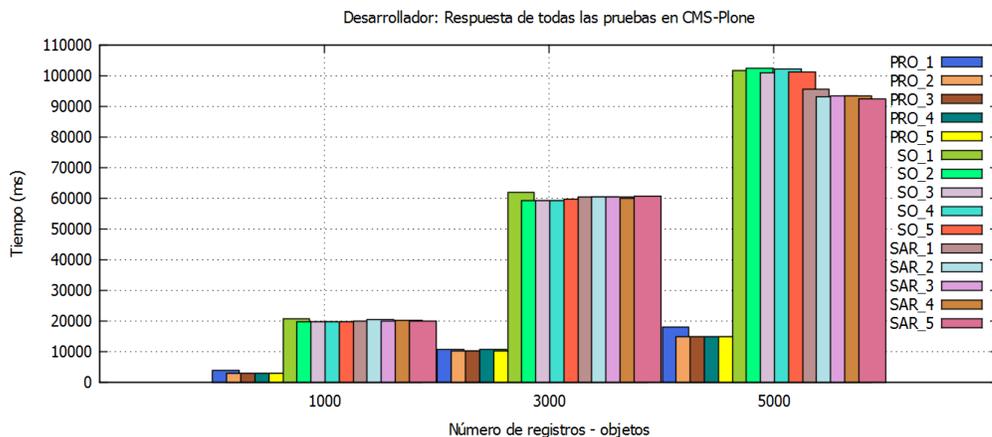


Figura 8.7: Desarrollador: Todas las pruebas en CMS

Observando los resultados de la gráfica 8.9 se puede ver que la diferencia en cada punto medido y en general la tasa de crecimiento del desempeño en ambas plataformas no deja duda de cual es para el reporte la opción que respondió mejor. De igual forma podemos ver el mismo comportamiento para las gráficas 8.10 y 8.11

Al final se muestra en la gráfica 8.12 el orden o mas bien, el tiempo de cada reporte en las dos plataformas y se ve 6 líneas cada una representando los promedios de cada experimento, marcando los tres reportes de la plataforma RDBMS un mejor tiempo que sus similares en la plataforma CMS.

Finalmente para cerrar el tema de desempeño de ambas plataformas en la última

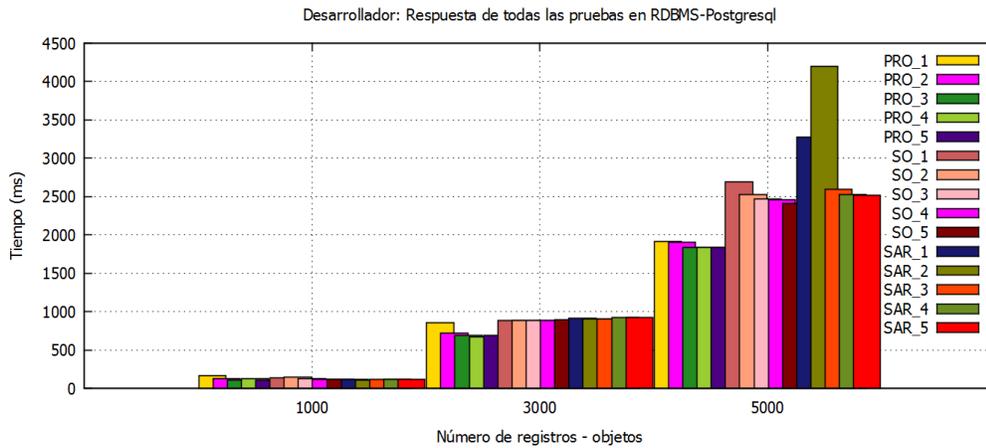


Figura 8.8: Desarrollador: Todas las pruebas en RDBMS

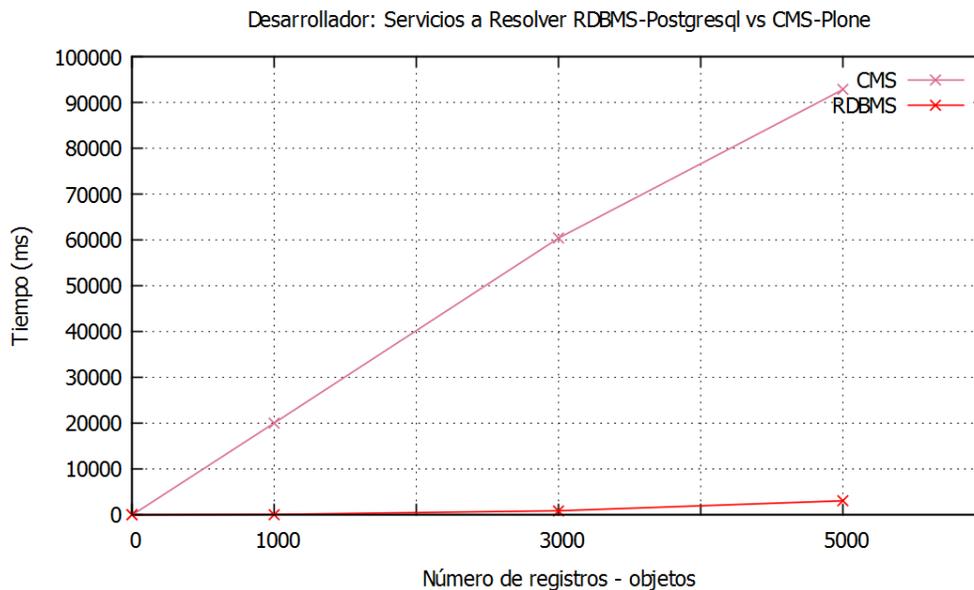


Figura 8.9: Desarrollador: SAR RDBMS vs CMS

gráfica 8.13 se intenta plasmar todas las pruebas en ambas plataformas y en ambos modos, poniendo como sufijo “txt” en caso de experimentos en modo desarrollador y un sufijo “img” en caso de modo Usuario final, de igual forma los acrónimos RDBMS y CMS dan la información sobre la plataforma en que corrieron los experimentos. Como un breve resumen gráfico, se ve en la ilustración que el experimento PRO en la plataforma RDBMS para el modo usuario final tomo un gran tiempo y se disparó de los demás, además que si se analiza con detalle se observa los tiempos mas altos son en la plataforma CMS y una vez que pasan todos, vienen en mejores tiempos los experi-

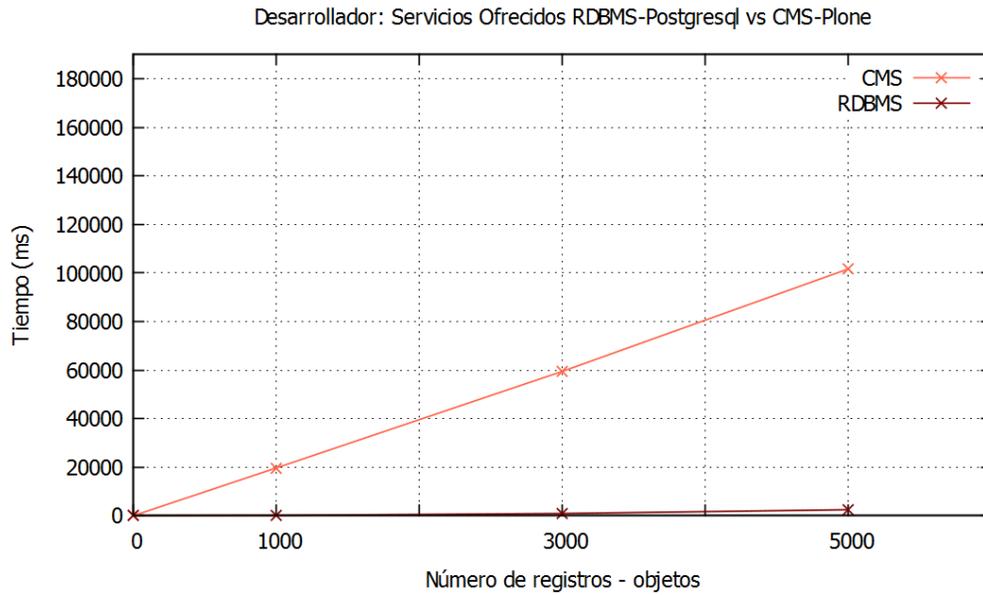


Figura 8.10: Desarrollador: SO RDBMS vs CMS

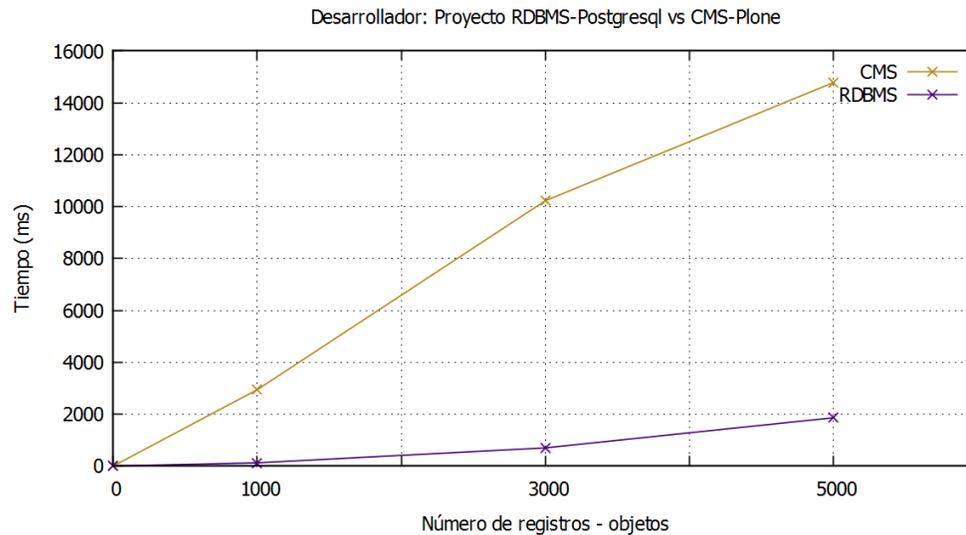


Figura 8.11: Desarrollador: PROY RDBMS vs CMS

mentos realizados en la plataforma RDBMS a excepción del primer reporte nombrado y del mismo reporte pero en la plataforma contraria, siendo este ultimo inclusive mejor que algunos reportes de la plataforma RDBMS.

Otro punto a resaltar de la última gráfica es que los tiempos de los experimentos que involucran imágenes son más lentos en comparación a los que tienen relación con

el uso exclusivo de la información almacenada.

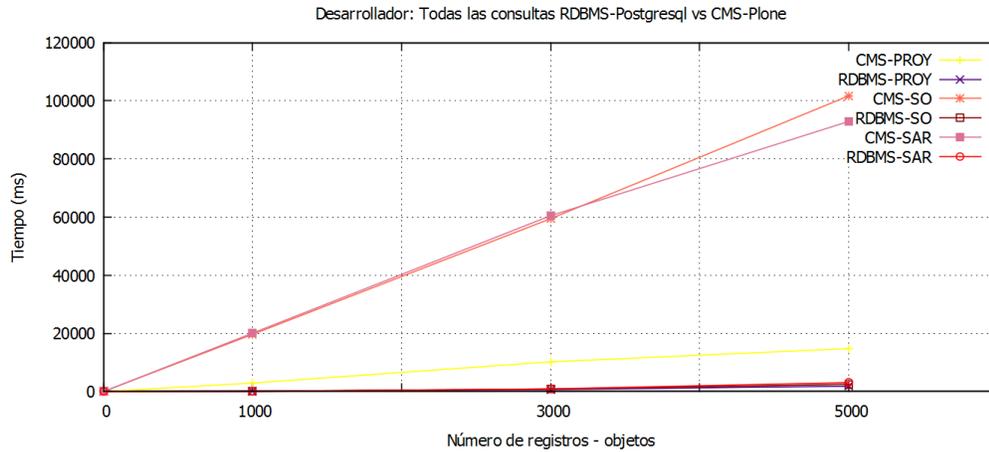


Figura 8.12: Desarrollador: Todos las consultas RDBMS vs CMS

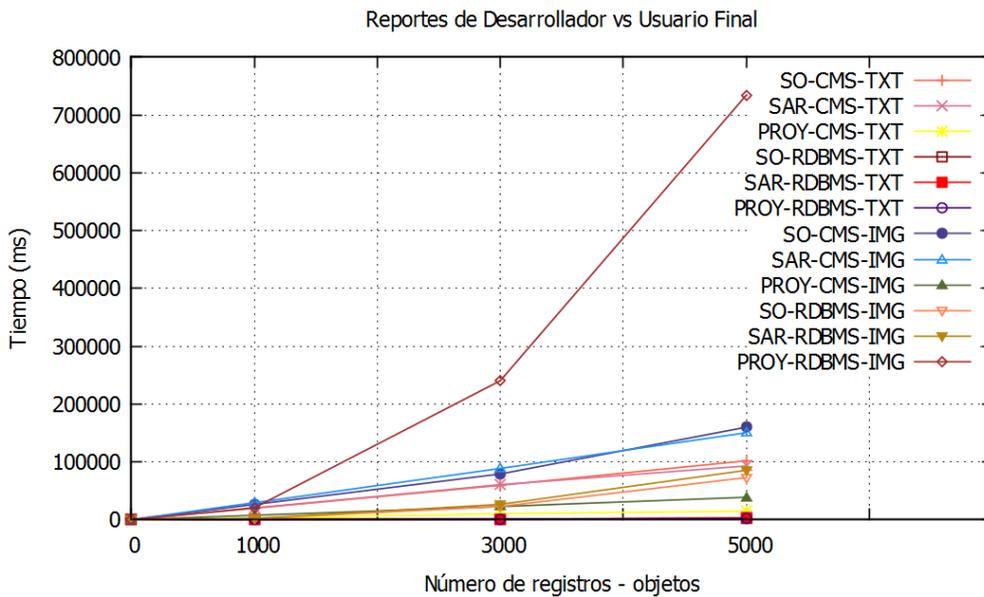


Figura 8.13: Todos los experimentos y sus combinaciones

8.3 Integridad y seguridad de la información

Los resultados son mostrados separando por usuario, pero presentando ambas plataformas con la intención de que se puedan visualizar lo mejor posible. Para hacer más legible la tabla, la columna uno tendrá las siguientes abreviaturas: I - Inserción, E -

Eliminación, M - Modificación, y C - Consulta, todas ellas varían de caso 1 al 5. Para facilitar la lectura de las columnas de resultados se anotó “SI” en caso de que la prueba fue exitosa, “NO” en caso contrario, o bien un “SI” con “O” (observaciones). Las observaciones se agregaron de forma abreviada en cada caso y representan los hallazgos u observaciones encontradas, para estos se utilizó la letra “O” Observación y un número consecutivo. Dicha lista de observaciones se puede consultar en la tabla 8.1.

Tabla 8.1: Listado de Observaciones

| Observación | Detalle |
|-------------|---|
| O1 | Al eliminar primero el departamento todo funcionó bien. Al eliminar primero el instituto, el departamento queda sin la referencia del instituto. Queda grabado el id del instituto, pero no hay nombre. Problema de Integridad Referencial. |
| O2 | En Semestre actual se ingresó una cadena y dejó hacerlo en vez de recibir un número. Problema de Integridad de Dominio. |
| O3 | La información no se pudo integrar como participante de laboratorio 1, sino al usuAdmon1, lo cual es un error. Problema de Integridad de Entidad. |
| O4 | Si se elimina la integración al laboratorio primero es correcto. Si se elimina primero al Alumno, deja borrarlo pero el reporte de integrantes de laboratorio genera error. Problema de Integridad de Entidad. |
| O5 | No se pudo ingresar el equipo de laboratorio al laboratorio 1, sino al usuAdmon1, lo cual es un error. Problema de Integridad de Entidad. |
| O6 | Se puede poner el rfc con una estructura incorrecta. Problema de integridad de Dominio. |
| O7 | La modificación que esté relacionada con el campo que identifica el save data adapter provoca error, otros no. Problema de Entidad. |
| O8 | No todas las validaciones automáticas se adaptan a las necesidades de todos los campos. Problema de Integridad de Dominio. |
| O9 | Se puede buscar el contenedor de información por medio de la búsqueda automatizada de objetos, pero debido a los permisos de publicación del contenido y permisos propios sobre la creación de objetos está denegada la acción. |
| O10 | Lectura permitida siempre y cuando haya permisos de leer en el objeto contenedor para el usuario y que el objeto a leer se encuentre en un estado de publicado, estas búsquedas aprovechan la jerárquica realizada automáticamente en Plone, o por medio de búsqueda de contenedores padre. |
| O11 | Realiza bien la eliminación, verifica la llave de departamento y manda mensaje en caso de que se quiera eliminar y exista. Si se elimina primero el depto se realiza sin contratiempos. Adecuado manejo de Integridad Referencial. |
| O12 | La inserción de los datos personales en la plataforma RDBMS se hace desde la creación de la cuenta del sistema. |
| O13 | La inserción de la persona es exitosa, pero la inclusión como integrante no se puede realizar. Manejo de Integridad de Entidad. |
| O14 | Se puede eliminar la persona pero no la integración en el grupo o laboratorio por cuestión de tipos o grupos de usuarios. Manejo de Integridad de Entidad. |
| O15 | Se puede modificar la persona pero no la integración en el grupo o laboratorio por cuestión de tipos o grupos de usuarios. Manejo de Integridad de Entidad. |
| O16 | Se puede consultar la persona pero no la integración en el grupo o laboratorio por cuestión de tipos o grupos de usuarios. Manejo de Integridad de Entidad. |
| O17 | La eliminación de un elemento se hizo sin la verificación de la integridad referencial. Error de Integridad referencial. |
| O18 | Esta observación debe tomarse como un error que no es derivado por la estructura o el alcance de las plataformas, sino más bien como errores generados por el alcance del desarrollo y/o de los requerimientos del mismo, es decir, las plataformas tienen los recursos para solventar estos errores. |

En los experimentos realizados con el usuario usuAdmon1 reflejado en la tabla 8.2 se puede observar que existen diferencias entre las plataformas, sin embargo, estas no son tan distintas, detallando un poco este asunto se tiene que la diferencia más relevante se presenta en el equipo de laboratorio, ya que mientras en la plataforma CMS se puede insertar, eliminar, modificar y consultar, en la otra plataforma no se permite. Esto pareciera decir, que tienen comportamientos diferentes, pero en realidad no es así, esto tiene que ver con el manejo de la funcionalidad que adquiere un Site Administrator y/o Administrator dentro de los roles en el perfil CMS y como se concibe en la plataforma RDBMS, además, la observación en la primer columna, aclara un poco el punto. Es decir, no tiene razón de ser el hecho de crear un equipo de laboratorio para un perfil que no coincide con el de laboratorio. Sin embargo, no se bloqueó del todo el proceso, se pudo haber hecho pero no se consideró por tratarse de un cambio que no afecta de manera importante el funcionamiento del sistema. La otra diferencia más visible se tiene en los datos personales, ya que, mientras en la primer plataforma se puede hacer todo, en la plataforma RDBMS no se permite eliminar, esto tiene que ver también más que con las posibilidades de una plataforma u otra, tiene que ver con la forma como se crean las cuentas de registro, ni siquiera los datos personales; es decir, en el sitio hecho con el CMS, viene una opción automática para crear cuentas de usuario, pero no incluyen los datos personales, por ello son creados un tanto aparte de su primer registro, y esto es lo que permite poder eliminarlo, por otro lado, en el portal construido con RDBMS, el registro de cuentas está ligado desde un inicio con el tipo de usuario y los datos inherentes, por ello una vez creado no se puede eliminar ya que forma parte de un registro inicial.

Sobre las pruebas hechas para usuAca1 (tabla 8.3) lo que se destaca son dos hechos, uno es la forma como maneja los proveedores en su conjunto ambas plataformas, y el otro es otro proceso automático hecho en CMS, el cual es la búsqueda. Estas pruebas de académico al ser las primeras donde el usuario ya tiene privilegios mas limitados a su actuar dentro del sistema, es obvio ver como empieza a limitarse su poder y accionar en las diferentes pruebas, encontrando muchas más veces un SI-018 o con otras observaciones que un SI. En las acciones C1 y C3, la plataforma CMS pareciera ser la opción más abierta al permitir al académico consultar y la otra plataforma no tanto; sin embargo, lo que hace diferencia es el uso de productos ya muy hechos en este CMS, los cuales realizan una búsqueda a través de sus objetos padres y/o contenedores, pudiendo a través de ellos y no de reportes dentro del sistema propio consultar cierta información, esto no hace menos o mas relevante a una plataforma u a otra, es simplemente distintas formas de controlar o permitir el uso de la información, pudiendo bloquear esos permisos u otorgarlos en ambas plataformas con distintos procesos, según sea el caso. El otro

Tabla 8.2: Resultados para el usuario usuAdmon1 - Administrador 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|----------------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Fac. de Filosofía - Filosofía ** | SI | SI |
| E1 | Instituto-Depto. | Fac. de Filosofía - Filosofía | SI- O1 | SI -O11 |
| M1 | Instituto-Depto. | Fac. de Filosofía - Filosofía | SI | SI |
| C1 | Instituto-Depto. | Fac. de Filosofía - Filosofía | SI | SI |
| I2 | Alumno-integrante Lab. | Arturo Díaz Gómez | SI - O2,O3 | SI - O13 |
| E2 | Alumno-integrante Lab. | Arturo Díaz Gómez | SI - O4 | SI |
| M2 | Alumno-integrante Lab. | Arturo Díaz Gómez | SI | SI |
| C2 | Alumno-integrante Lab. | Arturo Díaz Gómez | SI | SI - 018 |
| I3 | Equipo de Lab. | equipo de fusión | SI - O5 | SI - 018 |
| E3 | Equipo de Lab. | equipo de fusión | SI | SI - 018 |
| M3 | Equipo de Lab. | equipo de fusión | SI | SI - 018 |
| C3 | Equipo de Lab. | equipo de fusión | SI | SI - 018 |
| I4 | Externo-Integrante Gpo. | Carlos Bonavides | SI - O6 | SI - O13 |
| E4 | Externo-Integrante Gpo. | Carlos Bonavides | SI | SI |
| M4 | Externo-Integrante Gpo. | Carlos Bonavides | SI - O7 | SI |
| C4 | Externo-Integrante Gpo. | Carlos Bonavides | SI | SI - 018 |
| I5 | Datos personales | Datos personales de usuAdmon1 | SI - O8 | SI - O12 |
| E5 | Datos personales | Datos personales de usuadmon1 | SI | SI - 018 |
| M5 | Datos personales | Datos personales de usuAdmon1 | SI | SI |
| C5 | Datos personales | Datos personales de usuAdmon1 | SI | SI |

Tabla 8.3: Resultados para el usuario usuAca1 - Académico 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Fac. de Economía - Estadística | SI - O9 | SI - 018 |
| E1 | Instituto-Depto. | Fac. de Economía - Estadística | SI - O9 | SI - 018 |
| M1 | Instituto-Depto. | Fac. de Economía - Estadística | SI - O9 | SI - 018 |
| C1 | Instituto-Depto. | Fac. de Economía - Estadística | SI - O10 | SI - 018 |
| I2 | Alumno-integrante Lab. | Karla Martínez | SI - O9, O13 | SI O13 |
| E2 | Alumno-integrante Lab. | Karla Martínez | SI - O9 | SI |
| M2 | Alumno-integrante Lab. | Karla Martínez | SI | SI |
| C2 | Alumno-integrante Lab. | Karla Martínez | SI | SI |
| I3 | Equipo de Lab. | Herramientas de excavación | SI - O9 | SI - 018 |
| E3 | Equipo de Lab. | Herramientas de excavación | SI - O9 | SI - 018 |
| M3 | Equipo de Lab. | Herramientas de excavación | SI - O9 | SI - 018 |
| C3 | Equipo de Lab. | Herramientas de excavación | SI - O10 | SI - 018 |
| I4 | Externo-Integrante Gpo. | Emma Gil | SI - O9, O13 | SI - O13 |
| E4 | Externo-Integrante Gpo. | Emma Gil | SI - O9 | SI |
| M4 | Externo-Integrante Gpo. | Emma Gil | SI | SI |
| C4 | Externo-Integrante Gpo. | Emma Gil | SI | SI - O13 |
| I5 | Datos personales | Datos personales de usuAca1 | SI - O8 | SI - O12 |
| E5 | Datos personales | Datos personales de usuAca1 | SI - 018 | SI - 018 |
| M5 | Datos personales | Datos personales de usuAca1 | SI | SI |
| C5 | Datos personales | Datos personales de usuAca1 | SI | SI |

punto importante se presenta en las acciones E2 y E4, aquí los papeles se invierten, ya que la plataforma RDBMS parece más permisiva ahora, sin embargo, sigue tratándose de la mirada de cada plataforma al proceso que se lleva a cabo, mientras en el CMS eliminar un objeto compete solo a su dueño y a quien tenga permisos sobre el objeto en general y su jerarquía, en la otra plataforma, se convierte en un asunto de privilegios no de tipo de usuario sino de un grupo de ellos, los proveedores aquí pueden compartir responsabilidades sobre alumnos y/o externos para manipular su información, sea

eliminación, consulta, modificación y/o inserción.

Tabla 8.4: Resultados para el usuario usuLab1 - Laboratorio 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|------------------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Instituto de Geofísica - Geofísica | SI -O9 | SI - 018 |
| E1 | Instituto-Depto. | Instituto de Geofísica - Geofísica | SI -O9 | SI - 018 |
| M1 | Instituto-Depto. | Instituto de Geofísica - Geofísica | SI -O9 | SI - 018 |
| C1 | Instituto-Depto. | Instituto de Geofísica - Geofísica | SI -O10 | SI - 018 |
| I2 | Alumno-integrante Lab. | Ana Fuentes ** | SI | SI |
| E2 | Alumno-integrante Lab. | Ana Fuentes | SI - 018 | SI |
| M2 | Alumno-integrante Lab. | Ana Fuentes | SI -O7 | SI |
| C2 | Alumno-integrante Lab. | Ana Fuentes | SI | SI |
| I3 | Equipo de Lab. | Análisis sismografico ** | SI | SI |
| E3 | Equipo de Lab. | Análisis sismografico | SI - 018 | SI |
| M3 | Equipo de Lab. | Análisis sismografico | SI | SI |
| C3 | Equipo de Lab. | Análisis sismografico | SI | SI |
| I4 | Externo-Integrante Gpo. | Verónica Guerrero | SI - O9, O13 | SI - O13 |
| E4 | Externo-Integrante Gpo. | Verónica Guerrero | SI - 018 | SI |
| M4 | Externo-Integrante Gpo. | Verónica Guerrero | SI | SI |
| C4 | Externo-Integrante Gpo. | Verónica Guerrero | SI | SI - O13 |
| I5 | Datos personales | Datos personales de usuLab1 | SI - O8 | SI - O12 |
| E5 | Datos personales | Datos personales de usuLab1 | SI - 018 | SI - 018 |
| M5 | Datos personales | Datos personales de usuLab1 | SI | SI |
| C5 | Datos personales | Datos personales de usuLab1 | SI | SI |

En la tabla 8.4 se encuentran los experimentos realizados para el usuario usuLab1. En esta comparativa se notan las diferencias justo como la tabla anterior en las eliminaciones, en este caso las acciones E3 y E4, y lo mismo para la consulta C1.

Tabla 8.5: Resultados para el usuario usuGpo1 - Grupo de trabajo 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|-----------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Facultad de Química - QBF | SI -O9 | SI - 018 |
| E1 | Instituto-Depto. | Facultad de Química - QBF | SI -O9 | SI - 018 |
| M1 | Instituto-Depto. | Facultad de Química - QBF | SI -O9 | SI - 018 |
| C1 | Instituto-Depto. | Facultad de Química - QBF | SI - O10 | SI - 018 |
| I2 | Alumno-integrante Lab. | Jose Perez | SI - O9,O13 | SI - O13 |
| E2 | Alumno-integrante Lab. | Jose Perez | SI - 018 | SI |
| M2 | Alumno-integrante Lab. | Jose Perez | SI | SI |
| C2 | Alumno-integrante Lab. | Jose Perez | SI | SI |
| I3 | Equipo de Lab. | Equipo microbiología | SI -O9 | SI - 018 |
| E3 | Equipo de Lab. | Equipo microbiología | SI -O9 | SI - 018 |
| M3 | Equipo de Lab. | Equipo microbiología | SI -O9 | SI - 018 |
| C3 | Equipo de Lab. | Equipo microbiología | SI -O10 | SI - 018 |
| I4 | Externo-Integrante Gpo. | Juan Dorantes ** | SI | SI |
| E4 | Externo-Integrante Gpo. | Juan Dorantes | SI O14 | SI |
| M4 | Externo-Integrante Gpo. | Juan Dorantes | SI | SI |
| C4 | Externo-Integrante Gpo. | Juan Dorantes | SI | SI |
| I5 | Datos personales | Datos personales de usuGpo1 | SI - O8 | SI - O12 |
| E5 | Datos personales | Datos personales de usuGpo1 | SI - 018 | SI - 018 |
| M5 | Datos personales | Datos personales de usuGpo1 | SI | SI |
| C5 | Datos personales | Datos personales de usuGpo1 | SI | SI |

El comportamiento del usuario usuGpo1 se ve reflejado en la tabla 8.5 la cual

muestra que en cuestiones de seguridad e integridad de la información, las plataformas se comportan de manera similar, evidentemente tiene sus diferencias, surgidas por el camino distinto en su implementación, más no en su poder de salvaguardar la información y de como se administran los datos y sus procesos de trabajo. Por ejemplo, en la consulta del equipo de laboratorio, la eliminación del integrante de laboratorio y/o alumno, la consulta del instituto o del departamento.

Tabla 8.6: Resultados para el usuario usuPF1 - Persona Física 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|---------------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Fac. de Arq. – Arq. del paisaje | SI -O9 | SI - 018 |
| E1 | Instituto-Depto. | Fac. de Arq. – Arq. del paisaje | SI -O9 | SI - 018 |
| M1 | Instituto-Depto. | Fac. de Arq. – Arq. del paisaje | SI -O9 | SI - 018 |
| C1 | Instituto-Depto. | Fac. de Arq. – Arq. del paisaje | SI -O10 | SI - 018 |
| I2 | Alumno-integrante Lab. | Emilio Cortés | SI -O9 | SI - 018 |
| E2 | Alumno-integrante Lab. | Emilio Cortés | SI -O9 | SI - 018 |
| M2 | Alumno-integrante Lab. | Emilio Cortés | SI -O9 | SI - 018 |
| C2 | Alumno-integrante Lab. | Emilio Cortés | SI -O9 | SI - 018 |
| I3 | Equipo de Lab. | Medición y Calidad del agua | SI -O9 | SI - 018 |
| E3 | Equipo de Lab. | Medición y Calidad del agua | SI -O9 | SI - 018 |
| M3 | Equipo de Lab. | Medición y Calidad del agua | SI -O9 | SI - 018 |
| C3 | Equipo de Lab. | Medición y Calidad del agua | SI -O9 | SI - 018 |
| I4 | Externo-Integrante Gpo. | Patricia Espino | SI -O9 | SI - 018 |
| E4 | Externo-Integrante Gpo. | Patricia Espino | SI -O9 | SI - 018 |
| M4 | Externo-Integrante Gpo. | Patricia Espino | SI -O9 | SI - 018 |
| C4 | Externo-Integrante Gpo. | Patricia Espino | SI -O9 | SI - 018 |
| I5 | Datos personales | Datos personales de usuPF1 ** | SI -O8 | SI -O12 |
| E5 | Datos personales | Datos personales de usuPF1 | SI - 018 | SI - 018 |
| M5 | Datos personales | Datos personales de usuPF1 | SI | SI |
| C5 | Datos personales | Datos personales de usuPF1 | SI | SI |

Las tablas 8.6 y 8.7 son experimentos donde el tipo de usuario es el mas bajo, es decir, si eran personas físicas o personas morales no influyeron en el contexto, mas bien lo que en ambas plataformas les da sentido es el perfil de clientes, es decir, un nivel más abstracto de su tipo de usuario. En ambas tablas los privilegios son limitados y solo pueden afectar su información personal, lo cual es lo pensado dadas las condiciones de integridad y seguridad, lo único extra es la consulta en la plataforma CMS, pero eso deriva de un uso general de búsqueda automática ya previamente creada en los productos básicos de Plone, por tanto no viola ningún aspecto antes considerado. En general son los usuarios con perfiles más bajos y encontrar algo que permita notar diferencias entre las plataformas fue una tarea difícil.

Ahora se muestran las pruebas donde se interactúa con la información que otro usuario creó. En los resultados de la plataforma CMS, se hicieron dos pruebas en cada una, una probando con un estado de estado P:Privado en el objeto a examinar y después se probó con B:Publicado o bien, E:Enviado para Publicación, para dar ambos

Tabla 8.7: Resultados para el usuario usuPM1 - Persona Moral 1

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|-----------------------------|---------------|-----------------|
| I1 | Instituto-Depto. | Fac. Medicina - Cardiología | SI - O9 | SI - 018 |
| E1 | Instituto-Depto. | Fac. Medicina - Cardiología | SI - O9 | SI - 018 |
| M1 | Instituto-Depto. | Fac. Medicina - Cardiología | SI - O9 | SI - 018 |
| C1 | Instituto-Depto. | Fac. Medicina - Cardiología | SI -O10 | SI - 018 |
| I2 | Alumno-integrante Lab. | Monica Hernández | SI -O9 | SI - 018 |
| E2 | Alumno-integrante Lab. | Monica Hernández | SI -O9 | SI - 018 |
| M2 | Alumno-integrante Lab. | Monica Hernández | SI -O9 | SI - 018 |
| C2 | Alumno-integrante Lab. | Monica Hernández | SI -O9 | SI - 018 |
| I3 | Equipo de Lab. | Aparatos nucleares | SI -O9 | SI - 018 |
| E3 | Equipo de Lab. | Aparatos nucleares | SI -O9 | SI - 018 |
| M3 | Equipo de Lab. | Aparatos nucleares | SI -O9 | SI - 018 |
| C3 | Equipo de Lab. | Aparatos nucleares | SI -O9 | SI - 018 |
| I4 | Externo-Integrante Gpo. | Marco Huerta | SI -O9 | SI - 018 |
| E4 | Externo-Integrante Gpo. | Marco Huerta | SI -O9 | SI - 018 |
| M4 | Externo-Integrante Gpo. | Marco Huerta | SI -O9 | SI - 018 |
| C4 | Externo-Integrante Gpo. | Marco Huerta | SI -O9 | SI - 018 |
| I5 | Datos personales | Datos personales de usuPM1 | SI -O8 | SI - O12 |
| E5 | Datos personales | Datos personales de usuPM1 | SI - 018 | SI - 018 |
| M5 | Datos personales | Datos personales de usuPM1 | SI | SI |
| C5 | Datos personales | Datos personales de usuPM1 | SI | SI |

resultados en la misma columna se hizo uso de las abreviaturas antes mencionadas, además de las usadas en el primer bloque de pruebas. En este nuevo bloque solo se contemplan las acciones de E- Eliminación, M- Modificación y C- Consulta, ya que la inserción proviene de otro usuario.

Tabla 8.8: Resultados para el usuario usuAdmon2 - Administrador 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|-------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI; P instituto, E depto:SI | SI |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI; P instituto, E depto:SI | SI |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI; P instituto, E depto:SI | SI |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI -O14; E:SI -O14 | SI - O14 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI -O15; E:SI -O15 | SI - O15 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI -O16; E:SI -O16 | SI - O16 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI; E:SI | SI - 018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI; E:SI | SI - 018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI-O9; E:SI-O9 | SI - 018 |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI; B:SI | SI -O14 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI; B:SI | SI -O15 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI -O16; B:SI -O16 | SI -O16 |
| E5 | Datos personales | usuPF1 ** | P:SI; E:SI | SI - 018 |
| M5 | Datos personales | usuPF1 ** | P:SI; E:SI | SI - 018 |
| C5 | Datos personales | usuPF1 ** | P:SI-O9; E:SI-O9 | SI - 018 |

El experimento realizado con el usuario usuAdmon2, tabla 8.8 es como se esperaba al ser el tipo de usuario con mayor privilegios, este es el experimento que tuvo menos problemas en modificar, consultar, y eliminar información creada por otros usuarios, esto era un resultado esperado. Tal vez la parte de la prueba que puede causar desconcierto es en la manipulación del equipo de laboratorio, la cual en la plataforma

RDBMS no se puede acceder, mientras que la plataforma CMS si se puede utilizar, esto porque de entrada la plataforma CMS determina un super usuario a los grupos y roles Site Administrator y Administrators, además, en las pruebas siguientes como en esta es interesante cotejar además de las pruebas entre plataformas, la diferencia misma de la pataforma CMS con distintos estados de publicación escogidos de manera aleatoria.

Tabla 8.9: Resultados para el usuario usuAca2 - Académico 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|---------------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI - 018 |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI - 018 |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P inst, E depto:SI-09 | SI-018 |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI -O14 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI -O15 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-O16; E:SI-O16 | SI -O16 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI - 018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI - 018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI - 018 |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI - O14 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI - O15, O17 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI -O16; B:SI -O16 | SI -O16 |
| E5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI - 018 |
| M5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI - 018 |
| C5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI - 018 |

Sobre el académico 2, experimento 8.9; se puede resaltar el hecho de que sobre los integrantes del grupo, aquí el académico en la plataforma CMS, tiene dificultades al intentar modificar, o eliminar al externo, mientras que en la plataforma RDBMS no existe conveniente alguno, el porqué de esta diferencia está nuevamente ligado a la forma en como es tratada la información en la primer plataforma pero con algunas modificaciones de seguridad, esto podría revertirse, es decir, no existe diferencia en la capacidad y limitación entre los sistemas, bajo este usuario.

Los experimentos con los usuarios laboratorio 2 y grupo de trabajo 2, tablas 8.10 y 8.11, son similares en estructura, es decir, ambos presentan en la plataforma CMS restricciones para modificar y eliminar, pero no para consultar la información del otro usuario, y en la plataforma RDBMS ambos usuarios tienen libertad en cuanto manipular la información del alumno y externo, pero no así los integrantes de laboratorio y/o grupo. Esto muestra como algunos permisos y manejos de integridad de información son recargados no el tipo de usuario final sino en una capa más arriba en la que engloba a estos dos como proveedores.

Tabla 8.10: Resultados para el usuario usuLab2 - Laboratorio 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|---------------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI - 018 |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI - 018 |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P inst., E depto:SI-09, | SI - 018 |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI -O14 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI - O15 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI -O9; E:SI -O9 | SI -O16 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI - O9; E:OK -O9 | SI-018 |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI -O14 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI -O15 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI -O16; B:SI -O16 | SI -O16 |
| E5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| M5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| C5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |

Tabla 8.11: Resultados para el usuario usuGpo2 - Grupo de trabajo 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|---------------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P inst., E depto:SI-09 | SI-018 |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI -O14 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI -O15 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI -O16; E:SI -O16 | SI -O16 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI - O14 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI - O15 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI - O16; B:SI - O16 | SI - O16 |
| E5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| M5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| C5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |

Tabla 8.12: Resultados para el usuario usuPF2 - Persona Física 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|---------------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P inst., E depto:SI-09 | SI-018 |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI; E:SI | SI |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| E5 | Datos personales | usuPF1 ** | P:SI-018; B:SI-018 | SI-018 |
| M5 | Datos personales | usuPF1 ** | P:SI-018; B:SI-018 | SI-018 |
| C5 | Datos personales | usuPF1 ** | P:SI-018; B:SI-09 | SI-018 |

Por último, las tablas de los clientes, persona física 8.12 y persona moral 8.13. Los permisos en este experimento de consulta, modificación y eliminación de datos de

otro usuario, en cuanto a los clientes es similar, que el primer bloque de pruebas, ya que los privilegios son los mas restringidos entre el segundo bloque, tal y como sucedió la primera, vez, solo un permiso de consulta, lo cual deja la información integra y satisface la forma en como la información debe ser protegida.

Tabla 8.13: Resultados para el usuario usuPM2 - Persona Moral 2

| Acción | Reporte | Datos | Respuesta CMS | Respuesta RDBMS |
|--------|-------------------------|--------------|---------------------------------------|-----------------|
| E1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| M1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P instituto, E depto:SI-018 | SI-018 |
| C1 | Instituto-Depto. | usuAdmon1 ** | P:SI-018; P inst., E depto:SI-09 | SI-018 |
| E2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C2 | Alumno-integrante Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| E3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| M3 | Equipo de Lab. | usuLab1 ** | P:SI-018; E:SI-018 | SI-018 |
| C3 | Equipo de Lab. | usuLab1 ** | P:SI; E:SI | SI |
| E4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| M4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| C4 | Externo-Integrante Gpo. | usuGpo1 ** | P:SI-018; B:SI-018 | SI-018 |
| E5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| M5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |
| C5 | Datos personales | usuPF1 ** | P:SI-018; E:SI-018 | SI-018 |

Ahora se plantea una reseña de los resultados obtenidos.

Hablando sobre las pruebas de Desempeño, se puede decir que las pruebas de inserción y consulta masiva, dadas las propiedades ACID Atomicidad, Durabilidad y Consistencia; se tiene la siguiente conclusión dadas las plataformas RDBMS y CMS. Las propiedades ACID, son un conjunto de propiedades que hacen que la información tenga sentido y que los datos sean confiables, útiles e íntegros antes, durante y después de su manipulación. Las propiedades ACID en la plataforma CMS se encuentran ubicadas en la base de datos de objetos de Zope, ZODB es una implementación de persistencia de objetos que está estructurada en Python, esté conjunto de herramientas, hacen cumplir las reglas ACID, tal cual como lo haría en la plataforma RDBMS un sistema gestor de bases de datos relacional (en este caso Postgres). Si bien, en las tres propiedades que abarca este experimento ambas plataformas respondieron adecuadamente a las situaciones presentadas, es decir en ningún caso para ninguna de las dos se violaron o quebrantaron estas propiedades, si hubo situaciones donde una plataforma se comportaba mejor dada su estructura interna; aunque en general a manera de aprendizaje de este trabajo se puede decir que ambas plataformas son altamente confiables en cuanto a las propiedades ACID cuando se están realizando transacciones dentro del sistema. Ahora bien, ya que de manera práctica se llegó a la determinación que ambas plataformas son igualmente efectivas, solo resta acompañar dicha aseveración con los

siguientes hechos técnicos. ZODB para el experimento de lecturas es apoyado por una serialización de punteros a objetos, indexación de textos a nivel de la capa de aplicación, jerarquización objetos a través de las propiedades de herencia, y además cuenta con una técnica de “Packing”, la cual realiza un revisión y limpieza sobre el historial de objetos de ZODB. Por otra lado el manejador Postgres tiene un búsqueda de árbol b en las tablas, también utiliza serialización pero es respecto a las transacciones no a los objetos, y aunque el historial de ZODB podría asemejarse a un log de transacciones, también cuenta con un versionamiento de datos, utilizado principalmente para la concurrencia entre transacciones. Como se puede apreciar, ambas tecnologías aportan lo mejor de sí para resolver diversos escenarios en cuanto a la funcionalidad del sistema. Ahora bien otros puntos de los cuales se puede hablar de sus manejadores de información son los siguientes, mientras que en Postgres se puede considerar el escenario de tener réplicas y manejar planes de carga de ejecución de tareas, ZODB es un estilo centralizado. Algo que se hizo en las pruebas de funcionalidad y desempeño al hacer las lecturas para tener un equilibrio y similitud en las plataformas es al momento de seleccionar un objeto y un registro hacerlo de manera similar, esto es en ZODB para obtener una característica del objeto, es necesario tener el objeto listo en su totalidad, es decir si se tiene un objeto de muchas propiedades y por alguna razón se desea obtener solo una de ellas, se tendrán que cargar todas las propiedades antes de poder manipular aquella de interés. Este escenario es completamente distinto en Postgres ya que aquí hablando de registros, se puede acceder a todas las propiedades de una sola vez, o bien, a un subconjunto de ellos. Para las pruebas de esta tesis se eliminó esa ventaja en la plataforma RDBMS al hacer peticiones del registro completo aún y cuando no fuera necesario, esto como se mencionó antes, para dar equidad a las pruebas. Otro punto que no se probó, pero que sin duda alguna es una diferencia a rescatar es que en ZODB existe un proceso llamado “UNDO”, esto es, que dada una transacción se puede deshacer aún y cuando ya haya finalizado la transacción, esto, claramente no puede realizarse en un RDBMS como Postgres, ya que una vez que se termina con una transacción exitosa, esta no puede deshacerse. El Problema con el proceso de “UNDO” es que no existe la propiedad de hacerlo en cascada, es decir, si la realización del “UNDO” se lleva a cabo tiempo o transacciones después de la que se desea trabajar, no garantiza integridad en la información que haya sido manipulada después, por tanto no es una propiedad que se aconseje utilizar.

Las prácticas enfocadas al desempeño, tal vez fueron aquellas en las que se evidenciaron más las diferencias entre las plataformas CMS y RDBMS. Las gráficas y tablas de tiempo ya vistas muestran que hubo diferentes repuestas dado el ambiente a probar; para la prueba de inserción las tablas de tiempo mostraron grandes diferencias,

sin embargo influyó lo que se escogió como método de inserción, ya que aunque ambas opciones fueron elegidas por ser métodos ya implementados, el costo que sufrió de mas por parte de CMS y mucho menor en caso de RDBMS fue evidente al trabajar en capas superiores en caso del primero y en capas mas internas para el último caso. Es decir, una manera de haber comprobado el desempeño en una manera mas equilibrada hubiera sido, probar CMS a nivel de ZODB o bien, implementar un método de inserción dentro del sistema en RDBMS como se hizo con el producto de massloader para Plone. Sin embargo, refiriéndose a los experimentos de lectura se tienen dos grandes vertientes, una consulta donde mayoritariamente aparecen más elementos gráficos que información de texto, o bien, consultas desarrolladas para el tratamiento exclusivo de texto. Aquí se mostró en los resultados del experimento que la forma en como manipula la extracción y presentación del texto es más eficiente en la plataforma RDBMS que en su similar CMS, pero por otro lado cuando la visualización de la información implica carga el proceso de imágenes, la plataforma CMS tiene un desempeño de respuesta mejor que con la plataforma RDBMS. Esto tiene que ver principalmente con el tratamiento que se le da a las imágenes en cada plataforma, por un lado en CMS son inmersas dentro del sistema como un todo y son manejadas como otro objeto más, otorgando todos los beneficios que esto conlleva, en cuestión de almacenamiento, extracción y manipulación. Por otro lado los textos y su indexación, y la manera en como están relacionados permite a la plataforma RDBMS tener un manejo superior en cuanto a la información guardada, que como objetos en la plataforma CMS. Hay que tomar en cuenta que, como se mencionó para la funcionalidad RDBMS tenía una ventaja que fue suprimida al traer toda la información de los registros aún y cuando no fueran necesarios, siendo esto un contexto fuera de la investigación dichas consultas pudieron haberse llevado a cabo con un tratamiento donde deberían ser optimizadas al recoger solo lo necesario y evidentemente el desempeño hubiera sido aún mejor que el mostrado en estas pruebas; haciendo más evidente la separación del tiempo entre este y el desempeño del CMS para textos.

Revisando la Integridad de la información se tiene que lo arrojado en el plan de pruebas fue muy similar entre las plataformas. El experimento abarcó pruebas de Integridad Referencial, pruebas de Integridad de Dominio e Integridad de Entidad; las pruebas fueron vistas en dos fases en esta tesis, una involucraba las pruebas con inserciones de usuarios propios y otra fase con datos insertados por usuarios secundarios. Dichos resultados arrojaron una matriz muy interesante al mezclarse cuestiones de seguridad e integridad. Cabe mencionar que las pruebas de integridad que se realizaron tenían el objetivo de ver como respondían las plataformas a escenarios donde se respetaban o no dichas reglas, y también ver si las plataformas tenían implementadas

de manera automática algunas de estas reglas y cuales eran sus alcances. En ningún momento se pretendió establecer un escenario fuera de anomalías, ya que de ser así las pruebas caían en un ambiente extremo donde cada detalle hubiera sido cuidado y por tanto la verificación de las reglas era innecesario, pero mayor aún el interés de ver como reaccionaban las plataformas hubiera sido sesgado a escenarios positivos únicamente. Analizando las pruebas de Entidad y de Dominio, se observó que en la matriz hubo aciertos al cumplir reglas de integridad, pero también hubo fallas por incumplimiento de las mismas. Ejemplos puntuales son el rfc mal validado, algunos campos de caracteres que permitían datos numéricos o viceversa, código postal mal formado. Y otros datos que funcionaron bien como fechas de registro y nacimiento, valores booleanos en toma de decisiones si/no, combos con respuesta preseleccionadas. En la plataforma CMS la validación de los campos con los productos utilizados se daba en la generación del formulario la cual tenía automatizadas dos procesos, uno el cual ayudaba a la integridad de Entidad y de dominio a la vez, y tiene que ver con la presencia o no de valor, es decir, la aceptación de nulos a través del producto instalado, el cual tiene dentro del proceso de la creación del formulario una manera muy sencilla de verificar si el campo debe ser requerido o no, esto se hace a través de verificar una casilla al momento de crear el campo. De esta forma hacemos de un campo un posible candidato para reconocerlo como identificador, así ayudando a la integridad de Entidad, o bien, contemplándolo como integridad de Dominio y analizar si el campo puede estar vacío o no. La otra implementación que traía el producto es que también en su creación, dependiendo del campo que se trate, trae validaciones automáticas como validación de fechas reales, validación de tamaño de caracteres, de renglones; validación de cadenas, de números, dirección de e-mails. Si por alguna razón no se encuentra la validación que se desea implementar siempre está la opción de incluir dentro del flujo de trabajo del proceso del formulario un validador o script Python que ayude a verificar la correcta entrada de cada campo con algún evento a controlar después de correr dicho script, esto ya es algo que requiere un esfuerzo, tiempo y conocimiento extra. Por otro lado en cuestión de Integridad de Dominio, la plataforma RDBMS, no tiene implementado nada automatizado y toda validación se tiene que realizar al visitar la página o también como en el CMS con un script alterno que recorra todas las necesidades, y meterlo dentro del flujo del proceso para que pueda hacerse valer las reglas expuestas ahí. Existen ciertos frameworks (e.g. Java Server Faces) que no se contemplaron aquí para la plataforma RDBMS que tienen implementados como el CMS algunas validaciones automáticas, sin embargo, las que existen en este trabajo se tuvieron que desarrollar sin ningún tipo de ayuda. Lo que refiere esta explicación es que ambas plataformas cumplen con la identidad de Dominio, en ambas existen opciones automáticas y manuales. En la plataforma CMS la implementación de esta regla queda sujeta en la capa del sistema, es decir, de la

aplicación en su nivel más alto. Por otro lado la plataforma RDBMS tiene dos niveles, en la aplicación en la capa superior, y en el manejador de base de datos, al declarar la estructura de las tablas; así si no se valida la integridad en la capa superior existe otro filtro en el manejador.

Revisando la integridad de Entidad que es la forma de validar que algo pertenece a la entidad referida es de suma importancia en la credibilidad de la información verificar que toda información guardada dentro de la plataforma pueda identificarse de manera única, para esta tarea ambas plataformas respondieron de igual forma, es decir, de manera correcta. En la plataforma RDBMS la forma mas eficiente de hacerlo es respetar la normalización de la base de datos y hacerla valer, de esta forma se asegura y garantiza la manipulación correcta de todos los registros en ella. En la plataforma CMS tienen un identificador de objetos el cual es un id con fecha y numero de identificación, el cual también garantiza su unicidad, pero también existe la opción de ligarse a un identificador dado por el usuario, esto permite al usuario tener una referencia del objetos mas clara en su contexto.

Por último las pruebas de integridad referencial. Estas pruebas son algo en lo que ambas plataformas presentaron aciertos y errores, dentro de estos aciertos y errores, cabe mencionar que la plataforma RDBMS se comportó tal cual debía, es decir, los errores aquí fueron presentados ante situaciones donde la regla simplemente no se validaba, pero la plataforma manipuló los datos hasta el alcance que el manejador de base de datos lo permitió. Por otro lado el CMS también tuvo errores y aciertos, pero a diferencia del primero aquí hubo momentos donde se permitió hacer manipulación con la información que hacía de esta, una información no confiable al destruir vínculos entre ciertos objetos. Esto es posible de corregir a través de una de las propiedades de la forma en como el formulario guarda los datos, pero para el alcance de este trabajo, no se contempló determinar el modo de lograrlo. Sin embargo, eso no implica que en el paradigma de los CMS no puede haber integridad referencial entre objetos.

Las pruebas de integridad fueron de las más densas en cuanto a su contenido y se presentaron varios escenarios que anteriormente han sido resumidos a fin de darle agilidad, coherencia y cohesión a todas las pruebas y su análisis, si bien es cierto que hay diferencias en las tablas de datos arrojados, la forma de verlo es como un campo de oportunidades donde ambas plataformas (CMS y RDBMS) pueden valerse de sus propios medios para responder de manera adecuada ante todo tipo de reglas de integridad propuestas en un desarrollo de sistema.

Finalmente hablando sobre la seguridad una vez terminadas las pruebas en conjunto con la revisión de la integridad, se puede decir que también fueron pruebas donde hubo escenarios diversos y que los resultados fueron variados. Ciertamente, como se implementó la seguridad fue una decisión que se estructuró en los roles y grupos creados: proveedores (administradores, laboratorio, académicos, grupo de trabajo) y clientes (personas físicas y personas morales). Para la plataforma RDBMS la forma en como se implementó la seguridad se vió relacionada en bloquear páginas o columnas en reportes, botones, y eso prácticamente limitó todas las acciones dentro del sistema; tener el tipo de usuario en sesión, permitió discernir entre si el usuario en ese momento podía o no entrar en cierta página, o bien ver o no algún tipo de información. Por otro lado la plataforma CMS, tiene una visión de seguridad más global, que abarca las páginas, las acciones a hacer, quien puede ver la información, quien la manipula, quien cambia el estado de publicación. Existen usuarios dueños de la información, permisos heredados, y una lista de lo que cada objeto puede hacer junto con una matriz personalizada para cada objeto, además se cuentan con herramientas como la búsqueda y la visualización de objetos dentro de ella. Todo este arsenal de configuración crea dentro del sitio más de una posibilidad o escenario, sin duda alguna la tarea de seguridad dentro de la plataforma CMS está desarrollada de manera extensa y aunque muy granular puede de manera sencilla otorgarle al desarrollador y/o administrador herramientas suficientes para tener un control correcto del todo el sistema. Sin embargo, tan abstracto o de alto nivel como este esquema de seguridad, puede llegar a saturar y a dejar ciertas incongruencias al momento de cambiar configuraciones. En este experimento la plataforma RDBMS resultó comportarse tal y como se planteó en la definición de tipos de usuarios. Para la plataforma CMS seguir ese plan fue un poco mas complicado. A veces, esta combinación de resultados, hizo que la plataforma CMS reaccionará de manera favorable haciendo más flexibles los procesos y permitiendo que los roles y/o grupos tuvieran libertades que no afectaran o comprometieran la información y la integridad del sitio; sin embargo, hubo otras ocasiones que estas permisiones o restricciones, hicieron que el flujo de trabajo ya no fuera como el que se planteó y se permitieran algunas anomalías que hicieron que la información no tuviera integridad y perdiera credibilidad al permitir ciertas acciones a usuarios que no tenían que tener dichos privilegios. Entonces en este tema las plataformas funcionaron bien de acuerdo a su configuración dadas y al entorno, es decir, a los objetos y sus estados de publicación (en el caso CMS), es decir, nuevamente no se evalúa el comportamiento de la seguridad, sino como reacciona la plataforma ante eventos positivos o negativos. Se puede agregar que la seguridad en la plataforma RDBMS está presente en el flujo del MVC, entre la base de datos y el Modelo, y dentro de las vistas; mientras que en el CMS está presente en cada objeto y en cada acción, a través de diferentes mecanismos.

A manera de síntesis se puede decir que en seguridad los dos reaccionaron bien, pero es más completo CMS, en integridad reaccionaron bien, hubo problemas en CMS, pero son corregibles, existe un proceso “UNDO” que puede afectar la integridad. RDBMS tiene mayor control en la integridad. Desempeño, RBMS es mejor en inserción y consulta de textos, y CMS es mejor en consulta de imágenes. En cuanto a la funcionalidad, los dos respondieron bien y cuidan las tres propiedades probadas de las propiedades ACID.

En este experimento se probaron cuatro grandes rubros y se expusieron las debilidades de cada uno, se revalorizaron las fortalezas de estas dos plataformas, y hubo una explicación a los resultados encontrados. Todo esto con el fin de decidir entre ambas cuál plataforma puede brindar mayor confianza y certeza de que será un buen contexto para el desarrollo y uso de un sistema, con todo lo que esto conlleva.

Para finalizar este capítulo es conveniente resumir todos los hallazgos encontrados en las pruebas. Dicho resumen podemos encontrarlo en la tabla 8.14, las columnas RDBMS y CMS representan la respuesta de las pruebas en dichas plataformas, para hacerlo de forma general, se encontrará un “SI” en caso de que la plataforma respondiera de manera correcta o adecuada en términos generales, y en caso contrario se encontrará un “NO”. En la última columna llamada “Mejor respuesta” se pondrá el nombre de la plataforma que mejor resultados obtuvo para cada prueba.

Tabla 8.14: Resultados de las pruebas hechas en CMS y RDBMS

| Tipo de prueba | RDBMS | CMS | Mejor respuesta |
|---|-------|-----|-----------------|
| Desempeño - atomicidad - todos los reportes | SI | SI | AMBAS |
| Desempeño - durabilidad - todos los reportes | SI | SI | AMBAS |
| Desempeño - consistencia - todos los reportes | SI | SI | AMBAS |
| Funcionalidad - cota superior - inserción | SI | SI | RDBMS |
| Funcionalidad - cota superior - consulta. Manejo de Imágenes | SI | SI | CMS |
| Funcionalidad - inserción 1000 registros (1) | SI | SI | RDBMS |
| Funcionalidad - inserción 2000 registros (2) | SI | SI | RDBMS |
| Funcionalidad - inserción 2000 registros (3) | SI | SI | RDBMS |
| Funcionalidad - consulta - usuario final - proyecto. Manejo de Imágenes | SI | SI | CMS |
| Funcionalidad - consulta - usuario final - servicio ofrecido | SI | SI | RDBMS |
| Funcionalidad - consulta - usuario final - servicio a resolver | SI | SI | RDBMS |
| Funcionalidad - consulta - desarrollador - proyecto | SI | SI | RDBMS |
| Funcionalidad - consulta - desarrollador - servicio ofrecido | SI | SI | RDBMS |
| Funcionalidad - consulta - desarrollador - servicio a resolver | SI | SI | RDBMS |
| Integridad de dominio - todas las acciones - todos los reportes | SI | SI | RDBMS |
| Integridad de entidad - todas las acciones - todos los reportes | SI | SI | RDBMS |
| Integridad de referencial - todas las acciones - todos los reportes | SI | SI | RDBMS |
| Seguridad - administrador - todas las acciones - todos los reportes | SI | SI | CMS |
| Seguridad - académico - todas las acciones - todos los reportes | SI | SI | CMS |
| Seguridad - laboratorio - todas las acciones - todos los reportes | SI | SI | CMS |
| Seguridad - grupo de trabajo - todas las acciones - todos los reportes | SI | SI | CMS |
| Seguridad - persona física - todas las acciones - todos los reportes | SI | SI | CMS |
| Seguridad - persona moral - todas las acciones - todos los reportes | SI | SI | CMS |

Capítulo 9

Conclusiones y Trabajos Futuros

Con las pruebas realizadas y el análisis de los resultados hechos, en este capítulo se van a dar los resultados finales del experimento, se van a proponer en base a los resultados las posturas sobre cada característica de ambas plataformas y de manera general se va a justificar la decisión de usar una u otra tecnología. De la misma forma se va a hablar de las experiencias obtenidas durante el proceso y de posibles trabajos futuros.

9.1 Conclusiones

Este trabajo está hecho sobre la preocupación por saber discernir correctamente y tomar una decisión que involucre no solamente conocimientos técnicos, sino experiencias sobre desarrollos de sistemas en una vertiente o en otra (plataformas CMS y/o RDBMS). Cada persona tiende a hacer una visión holística del comportamiento de una u otra plataforma, lo cual es adecuado, ya que hay que tratar de ver cada situación por completo y con ello poder determinar fortalezas o debilidades dado un contexto en específico.

Para esta tesis, si bien tratar de detallar cada aspecto medido y tratar de puntualizar lo mejor posible sobre cada aspecto visto durante el desarrollo del trabajo fue una tarea importante, es evidente que se tiene que llegar a una visión general, sin perder la granularidad del detalle, para así tener la habilidad de tomar la mejor decisión en cuanto al uso de una plataforma.

Primeramente se realizó un análisis del contexto de la plataforma CMS, y posibles intentos de realización de estudios sobre la utilización por una parte de CMS y por otra de los RDBMS. De manera alterna se buscó el sustento en determinar un plan

de pruebas que no tuviera ningún sesgo y que otorgara imparcialidad al realizar dichos experimentos sobre ambas plataformas, con el fin de determinar sus comportamientos a través de los resultados arrojados.

Posteriormente la atención fue dirigida a una necesidad real como es el sistema piloto de la Secretaría de Vinculación y tomar este problema como base para la construcción de un sistema que tratara de respetar todas las características en ambas plataformas, exponiendo sus ventajas y desventajas al momento del desarrollo y uso de ambos.

Una vez concluido lo anterior la necesidad se enfocó en darle forma y aterrizar los conocimientos adquiridos en un plan de pruebas específico el cual derivó en dos grandes bloques: Funcionalidad y desempeño del sistema, y por otro lado, integridad y seguridad de la información. Dicho plan de pruebas contempló más de 18 pruebas de inserción masivas, más las contempladas para las cotas superiores; alrededor de 180 consultas masivas distribuidas en 3 reportes y para 3 diferentes cantidades de registros; un total de 300 pruebas para la integridad y seguridad en CMS y 210 para su similar RDBMS.

Con base en lo anterior, se llevó a cabo la recopilación correspondiente de todos estos datos, se les dio un tratamiento para un mejor análisis y visualización; además se interpretaron todos estos resultados y se les dio cohesión con ciertos aspectos técnicos que reforzaron y/o complementaron la información arrojada en las pruebas.

Se recapituló sobre las ventajas y desventajas de desarrollo tanto de una plataforma y otra, y se determinó que aunque cada paradigma aporta sus propias y muy específicas ventajas y desventajas, ninguna de las dos fue tan grande o impactante como para en base a eso preferir una sobre la otra, es decir, aunque se tiene para ciertas características una mejor opción, hablando nuevamente de una versión completa ambas plataformas puede resultar útiles, aunque los caminos puede ser más o menos cortos dependiendo de la experiencia, tiempo y forma de trabajo y la forma como se lleva todo esto a la práctica.

Al final de esta tesis se puede decir que la frialdad de los datos puede ser confusa, es decir, si bien es importante medir y cuantificar los fenómenos ocurridos, la interpretación de los mismos, su comparación y contextualización dentro de una situación hacen que los resultados finales puedan tener un sentido más amplio, esto abre la posibilidad a juzgar con mayor conocimiento cada evento estudiado.

La postura que se plantea es la de utilizar cada plataforma teniendo establecido un interés global del sistema, es decir, ahora se tienen elementos para discutir y valorar temas como la seguridad, integridad, funcionalidad, desempeño y, ventajas y desventajas de desarrollo. sin embargo, lo que hará que las diferencias en estos temas cobre un peso real para tomar una decisión de usar una plataforma u otra, está ligado al uso del sistema.

9.2 Trabajos futuros

Una vez finalizado el experimento y el desarrollo de esta tesis, fue evidente y necesario abrir un espacio para puntualizar sobre aquellos elementos en los que se podría profundizar o bien, dado el contexto de estos dos sistemas, hacer modificaciones con el fin de explorar nuevas características y realizar pruebas con distinto enfoque.

Dicho lo anterior se plantea hacer más pruebas de desempeño y estrés si las condiciones de configuración de hardware crecen. Además se puede contemplar recrear más escenarios dedicados a la seguridad con el fin de abarcar un espectro más amplio.

Se recomienda cubrir la integridad de usuario, y la propiedad ACID de aislamiento. Y una vez tocado el punto de concurrencia plantear pruebas y contextos adecuados para verificar paralelismo y distribución.

Como se puede observar, el sistema puede estar preparado para establecer normas y probar muchas otras características, el estudio comparativo entre sistemas con plataformas RDBMS y CMS es un tema reciente por tanto las áreas de oportunidades son muchas.

Bibliografía

- [1] *Diccionario de la Lengua Española*. Vigésima segunda edición.
- [2] *How to evaluate a content managment system?* http://www.steptwo.com.au/papers/kmc_evaluate/index.html, Última consulta Marzo 2012.
- [3] *Is a hybrid database in your future?* <http://sunsite.uakom.sk/sunworldonline/swol-02-1997/swol-02-objects.html>, Última consulta Marzo 2012.
- [4] *Plone*. <http://www.plone.org>, Última consulta Marzo 2012.
- [5] *Postgresql*. <http://www.postgresql.org/>, Última consulta Marzo 2012.
- [6] *Python*. <http://www.python.org/>, Última consulta Marzo 2012.
- [7] *Sitio web que compara características entre CMS*. <http://www.cmsmatrix.org/matrix/cms-matrix>, Última consulta Marzo 2012.
- [8] *what is a cms?* http://www.steptwo.com.au/papers/kmc_what/index.html, Última consulta Marzo 2012.
- [9] *zodb*. <http://www.zodb.org/>, Última consulta Marzo 2012.
- [10] *zope*. <http://www.zope.org/>, Última consulta Marzo 2012.
- [11] ISO/IEC JTC 1. *ISO/IEC 25012, SQuaRE, Software Engineering - Software Product Quality Requirements and Evaluation - Data quality model*. ISO/IEC, 2008.
- [12] D. Robert Adams. Teaching web application development using zope. *J. Comput. Small Coll.*, 20:63–70, October 2004.
- [13] Boiko Bob. *Content Management Bible, 2nd Edition*. Wiley Publishing, Inc.
- [14] Greg Deitz. Developing a content management system for siguccs. In *Proceedings of the 36th annual ACM SIGUCCS fall conference: moving mountains, blazing trails*, SIGUCCS '08, pages 141–144, New York, NY, USA, 2008. ACM.

-
- [15] Silberschatz Abraham et all. *Database System Concepts, Fifth Edition*. Mc Graw Hill, 2008.
- [16] Tom Gross. *Plone 3 Multimedia*. Packt Publishing, 2010.
- [17] Rodríguez Hernández Jonathan. *Componentes de Bases de Datos para MO-PROSOFT*. UNAM. Tutor Dr. Javier García García, 2008.
- [18] James O. Knowlton. *Python: Create - Modify - Reuse*. Wrox Press Ltd., Birmingham, UK, UK, 2008.
- [19] Sam Knox, Jon Stahl, Martin Aspeli, David Convent, Darci Hanning, Ricardo Newbery, John DeStefano, Clayton Parker, Alex Clark, and Veda Williams. *Practical Plone 3: A Beginner's Guide to Building Powerful Websites*. Packt Publishing, 2009.
- [20] Reuven M. Lerner. At the forge: Databases and zope. *Linux J.*, 2002:9–, May 2002.
- [21] López Rabadán Marco. *Sistema sobre Plone para la captura y recolección de información curricular del instituto de matemáticas*. UNAM. Tutor Dr. Sergio Rajsbaum Gorodezky, 2007.
- [22] Aspeli Martin. *Professional Plone 4 Development*. Wiley Publishing, Inc, 2011.
- [23] Nirav Mehta. *Choosing an Open Source CMS: Beginner's Guide*. Packt Publishing, 2009.
- [24] Serena Pastore. Web content management systems: using plone open source software to build a website for research institute needs. In *Proceedings of the international conference on Digital Telecommunications*, pages 24–, Washington, DC, USA, 2006. IEEE Computer Society.
- [25] A project of the IEEE Computer Society Profesional Practices Committee. *SWE-BOK: Guide to the Software Engineering Body og Knowledge*. IEEE, 2004.
- [26] Chanchal K. Roy and James R. Cordy. Are scripting languages really different? In *Proceedings of the 4th International Workshop on Software Clones, IWSC '10*, pages 17–24, New York, NY, USA, 2010. ACM.
- [27] Doug L. Simpson. Content for one: developing a personal content management system. In *Proceedings of the 33rd annual ACM SIGUCCS fall conference, SIGUCCS '05*, pages 338–342, New York, NY, USA, 2005. ACM.

-
- [28] Jurriaan Souer, Inge Van De Weerd, Johan Versendaal, and Sjaak Brinkkemper. Situational requirements engineering for the development of content management system-based web applications. *Int. J. Web Eng. Technol.*, 3:420–440, July 2007.
- [29] George K. Thiruvathukal and Konstantin Laufer. Plone and content management. *Computing in Science and Engg.*, 6:88–95, July 2004.
- [30] Curiel Díaz Arturo Tlacaoel. *Diseño e Implementación en Plone de un Sistema de Manejo de Solicitudes de Presupuesto Mediante Flujos de Trabajo*. UNAM. Tutor Dr. Sergio Rajsbaum Gorodezky, 2011.
- [31] Espinosa Ávila Eduardo. *Desarrollo de un Sistema de Administración de Procesos en Plone*. UNAM. Tutor Dr. Sergio Rajsbaum Gorodezky, 2009.

Apendice A

Este apéndice A, sirve para complementar la información brindada en los diferentes capítulos de esta tesis. En esta sección se podrán encontrar diferentes gráficas y textos que dan un apoyo extra que de cierta forma aporta información valiosa sobre puntos específicos, pero que fueron apartadas de la estructura general de la tesis para hacer de esta una lectura más fluida y concreta.

ESTRUCTURA DEL SISTEMA DE VINCULACIÓN UTILIZANDO CMS - PLONE



/Sistema de Vinculación

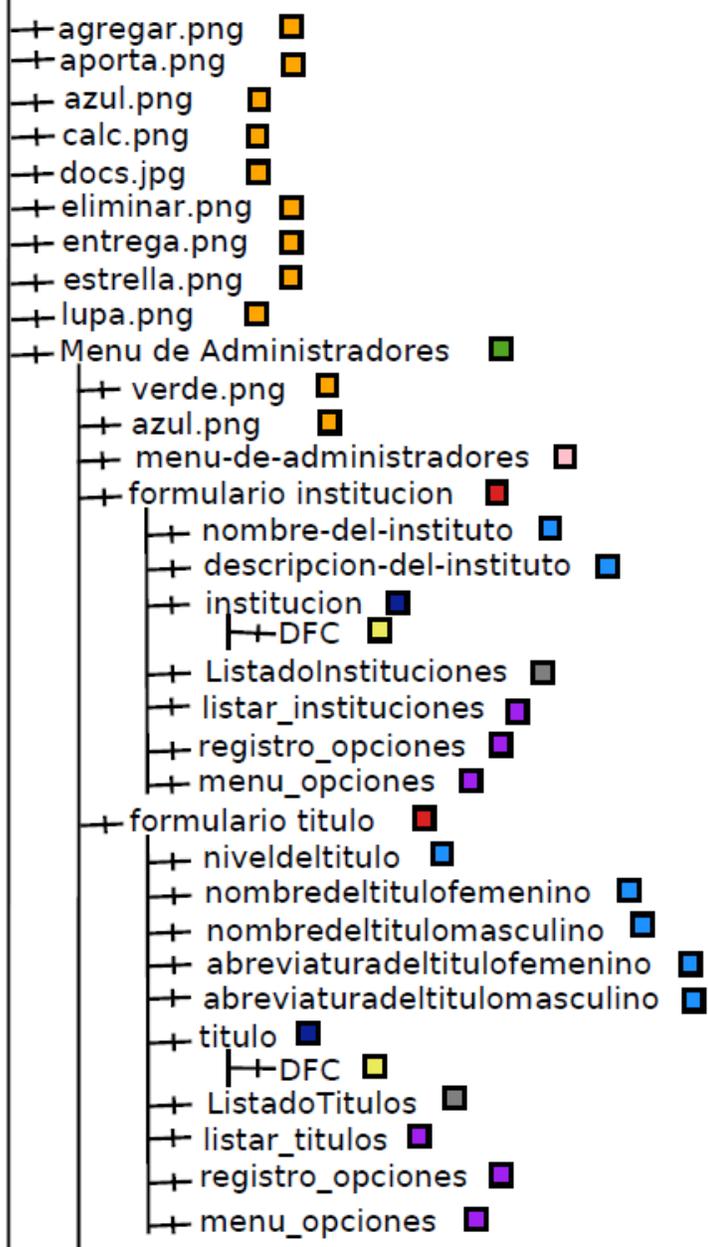


Figura A.2: Estructura del Sitio en Plone

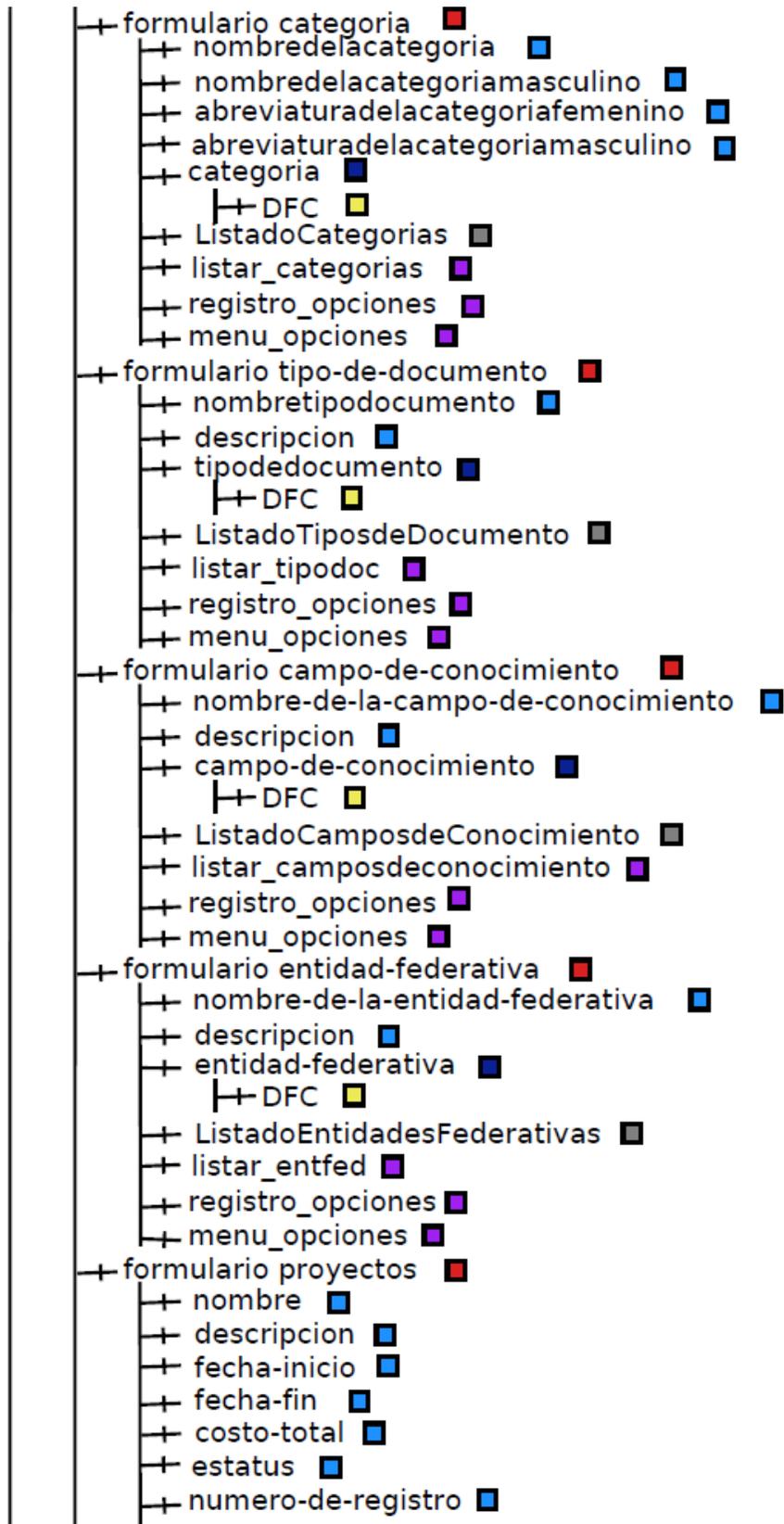


Figura A.3: Estructura del Sitio en Plone

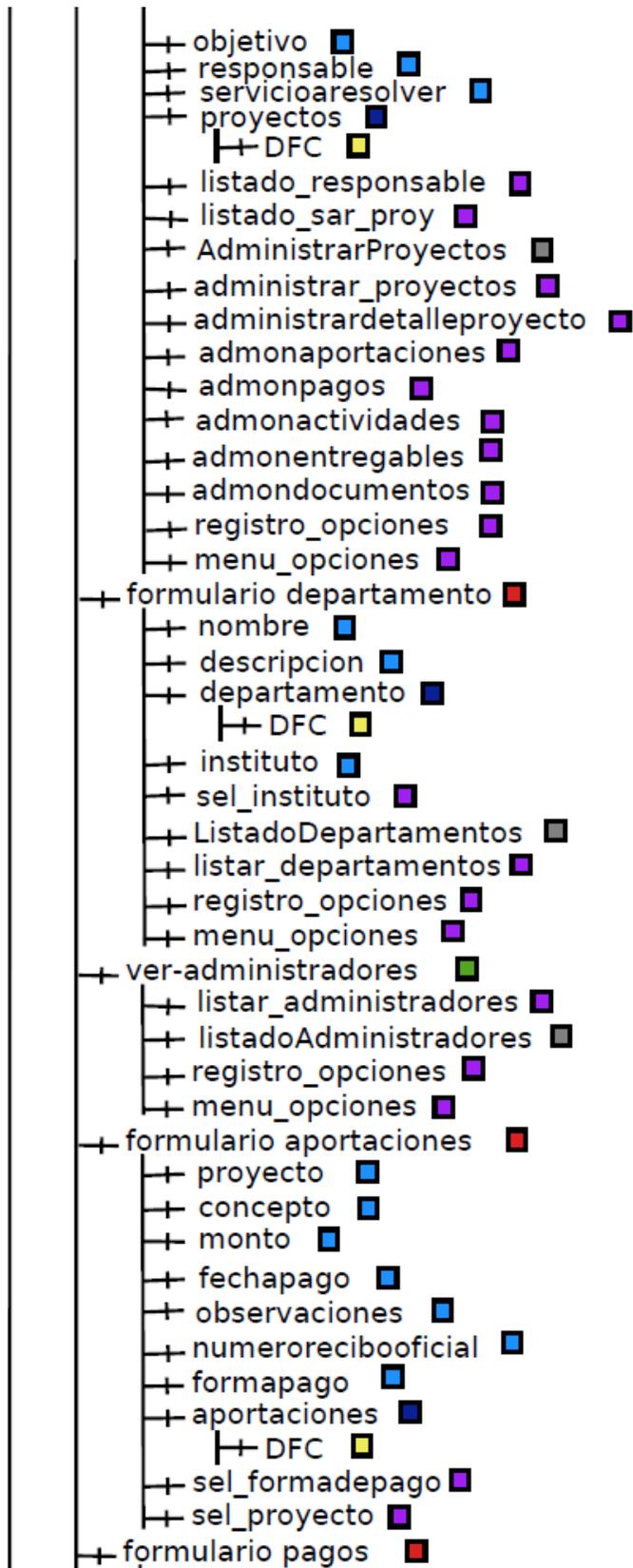


Figura A.4: Estructura del Sitio en Plone

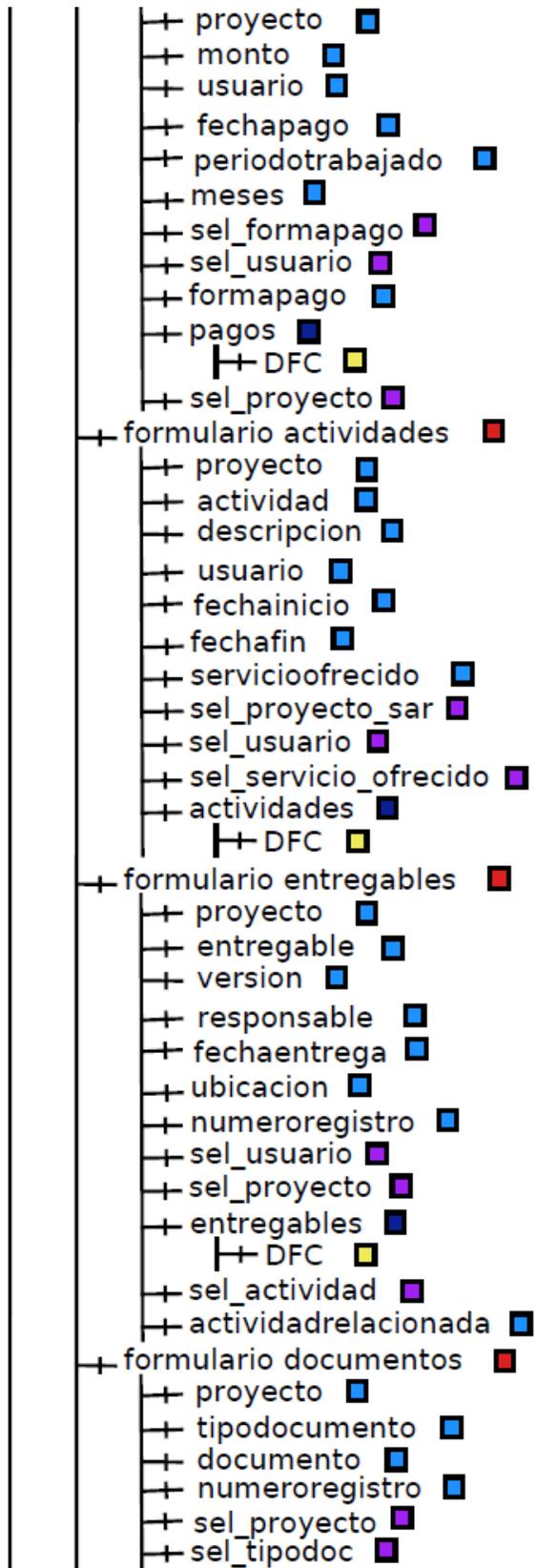


Figura A.5: Estructura del Sitio en Plone

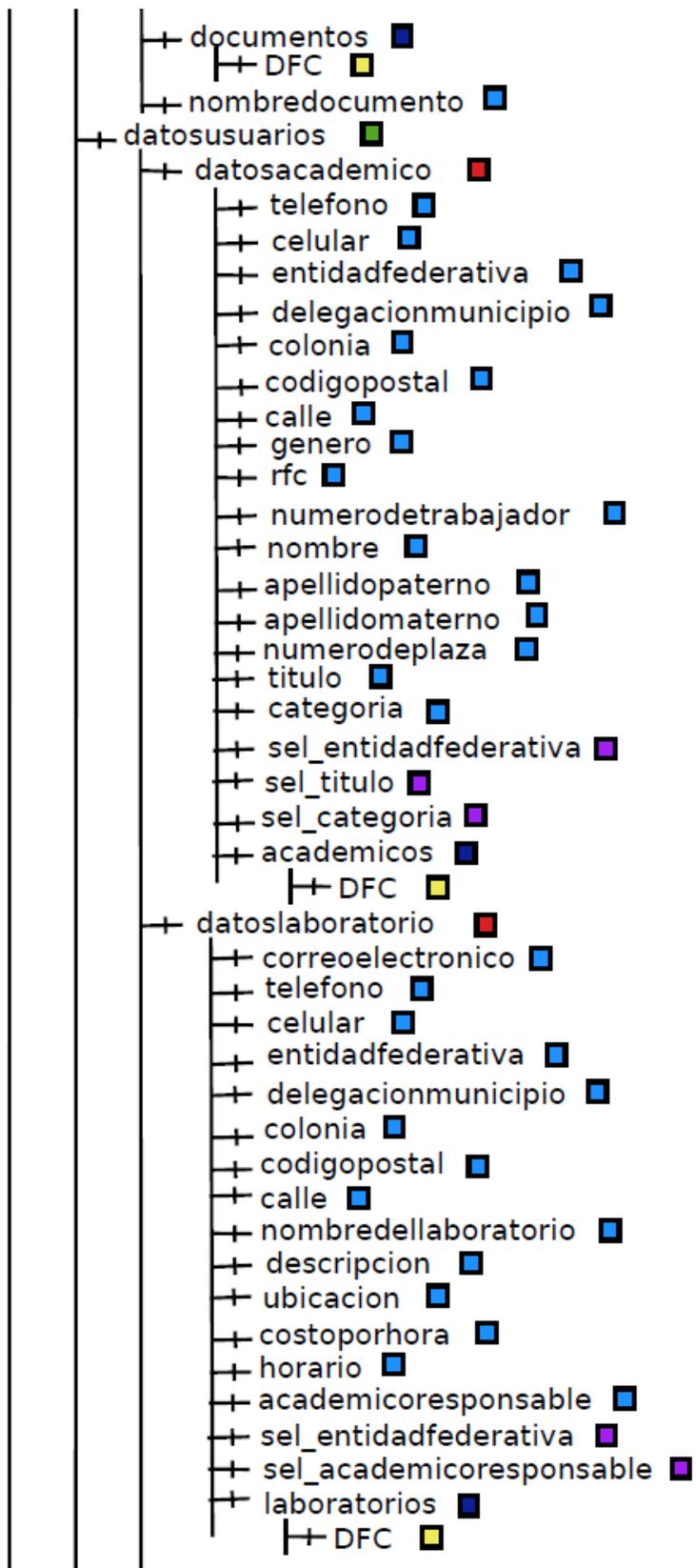


Figura A.6: Estructura del Sitio en Plone

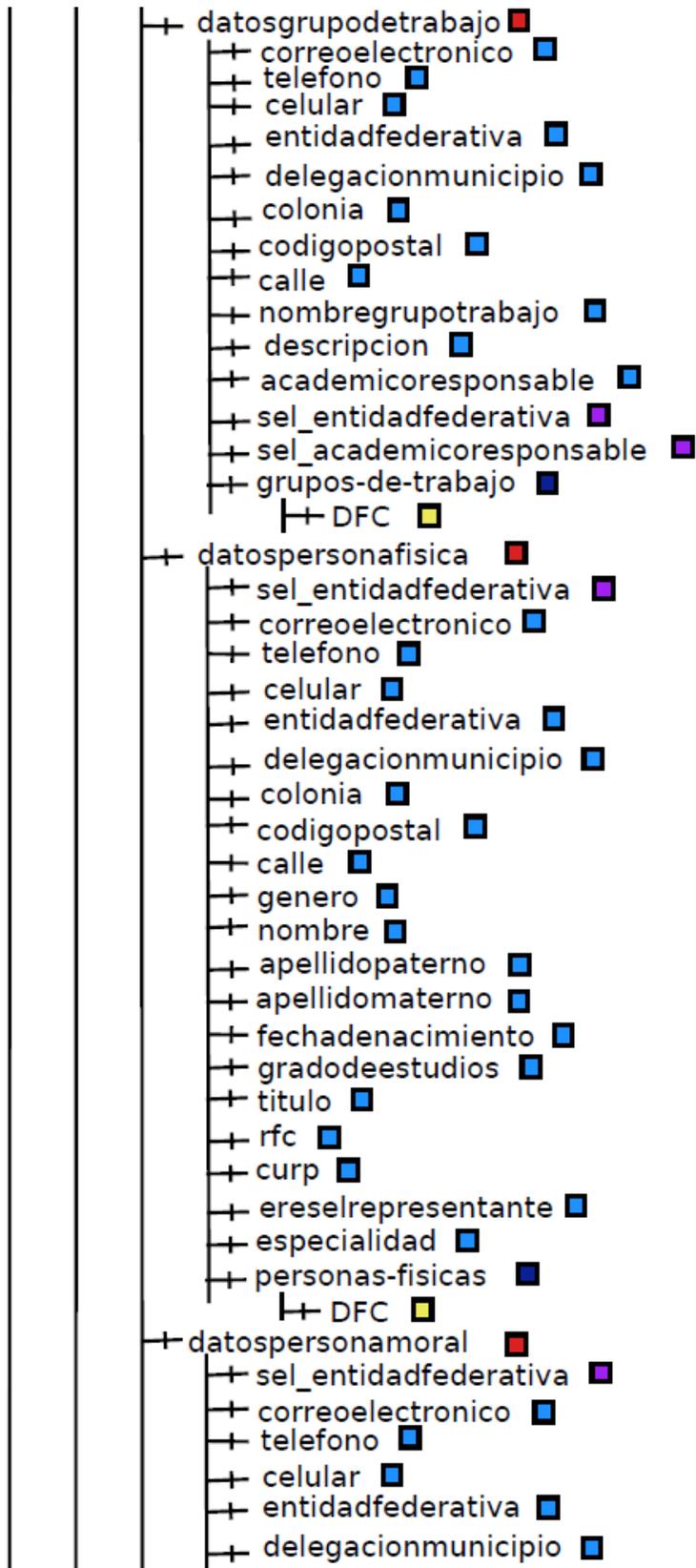


Figura A.7: Estructura del Sitio en Plone

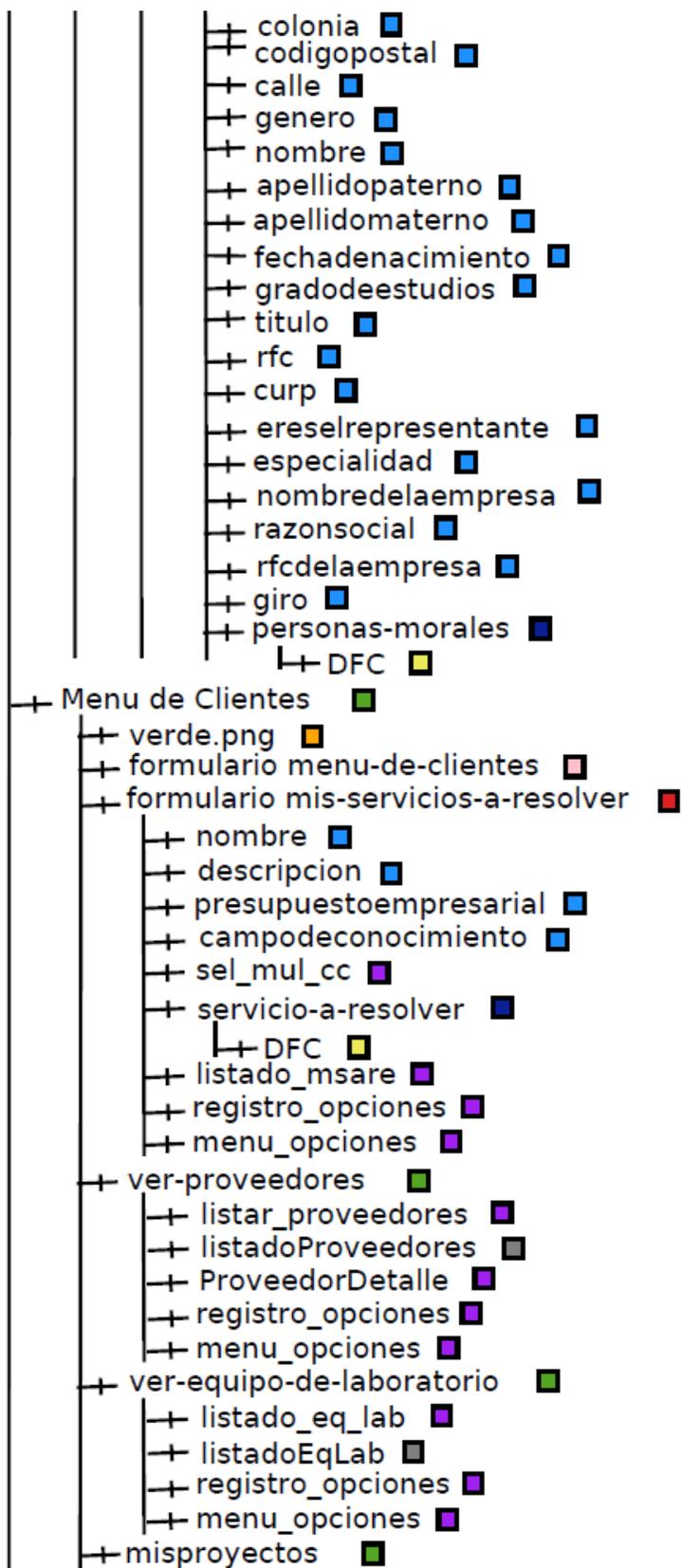


Figura A.8: Estructura del Sitio en Plone

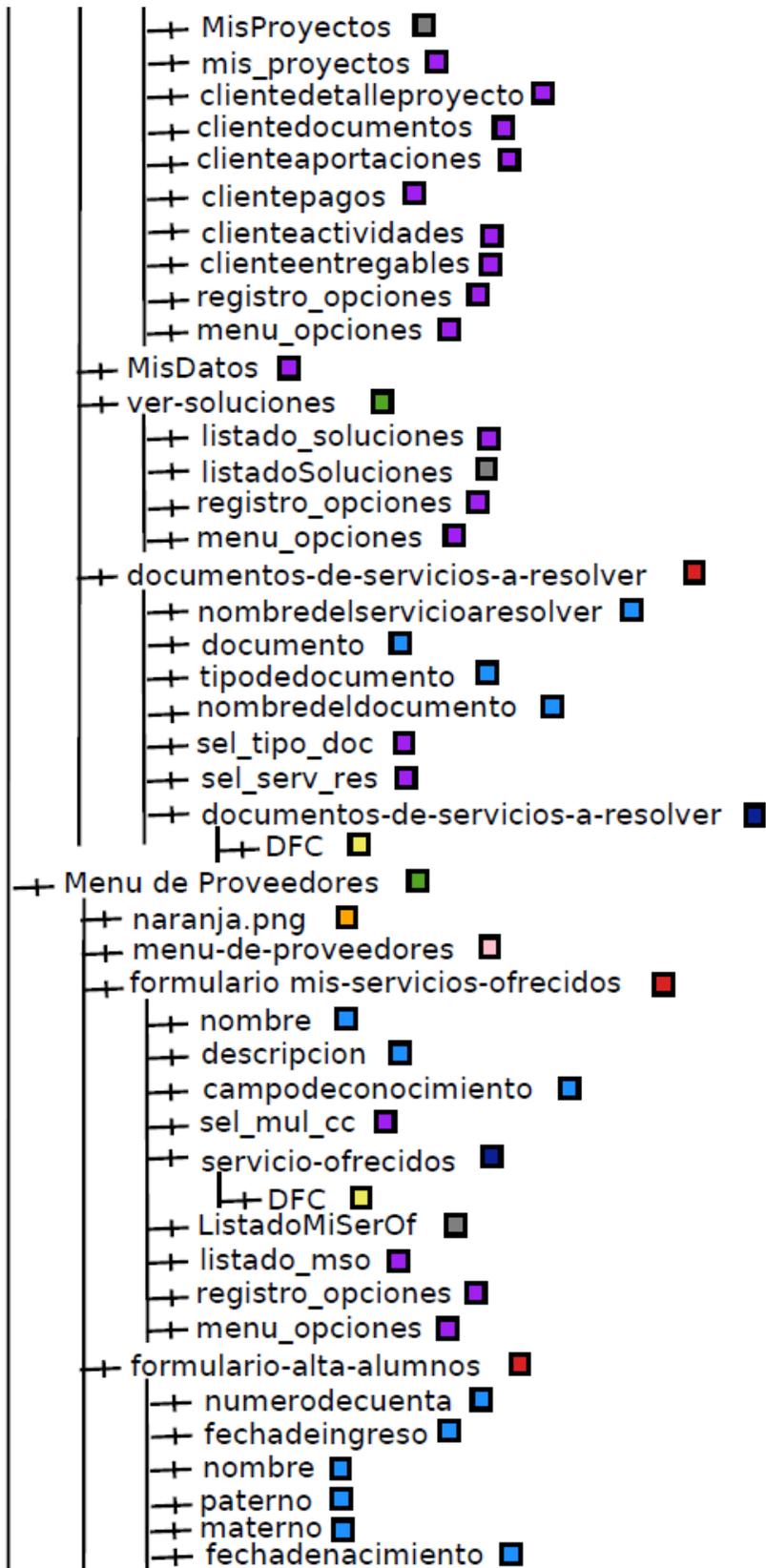


Figura A.9: Estructura del Sitio en Plone

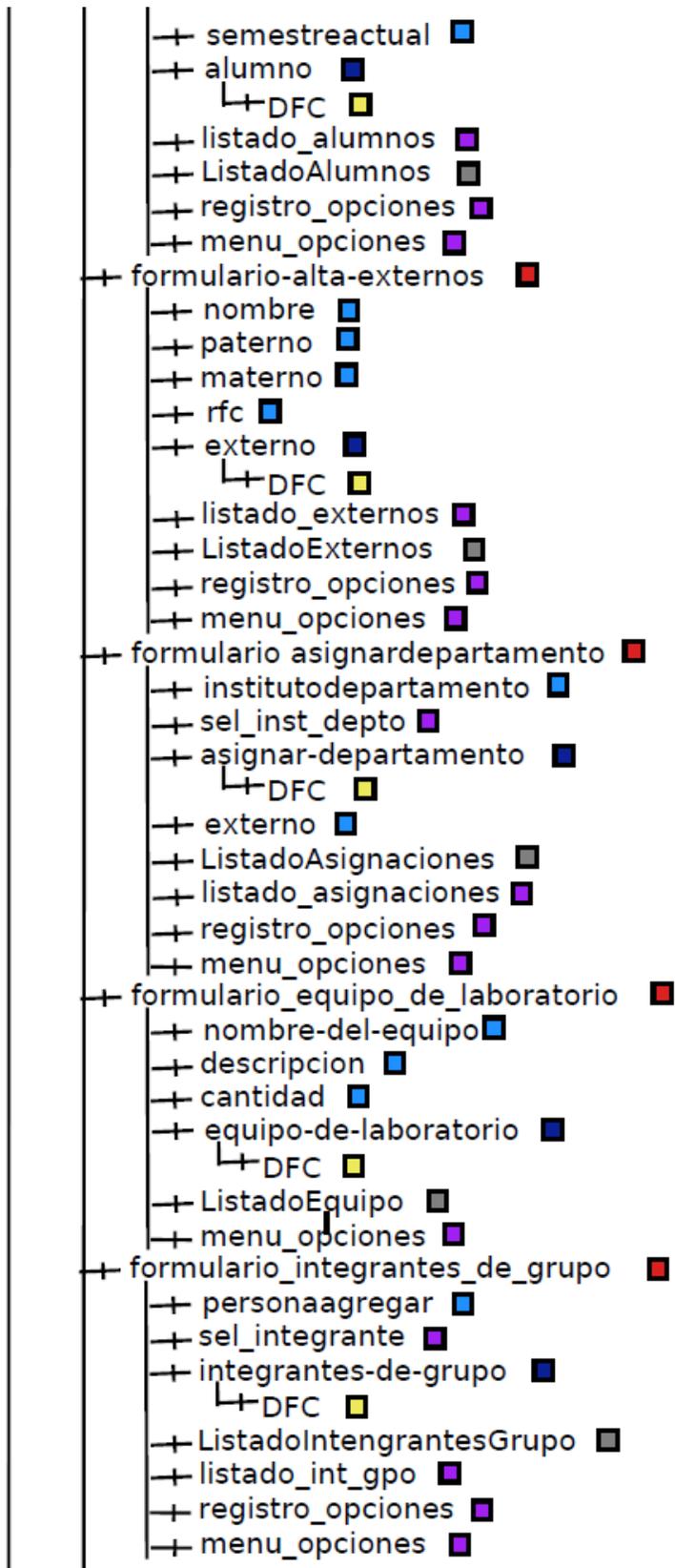


Figura A.10: Estructura del Sitio en Plone

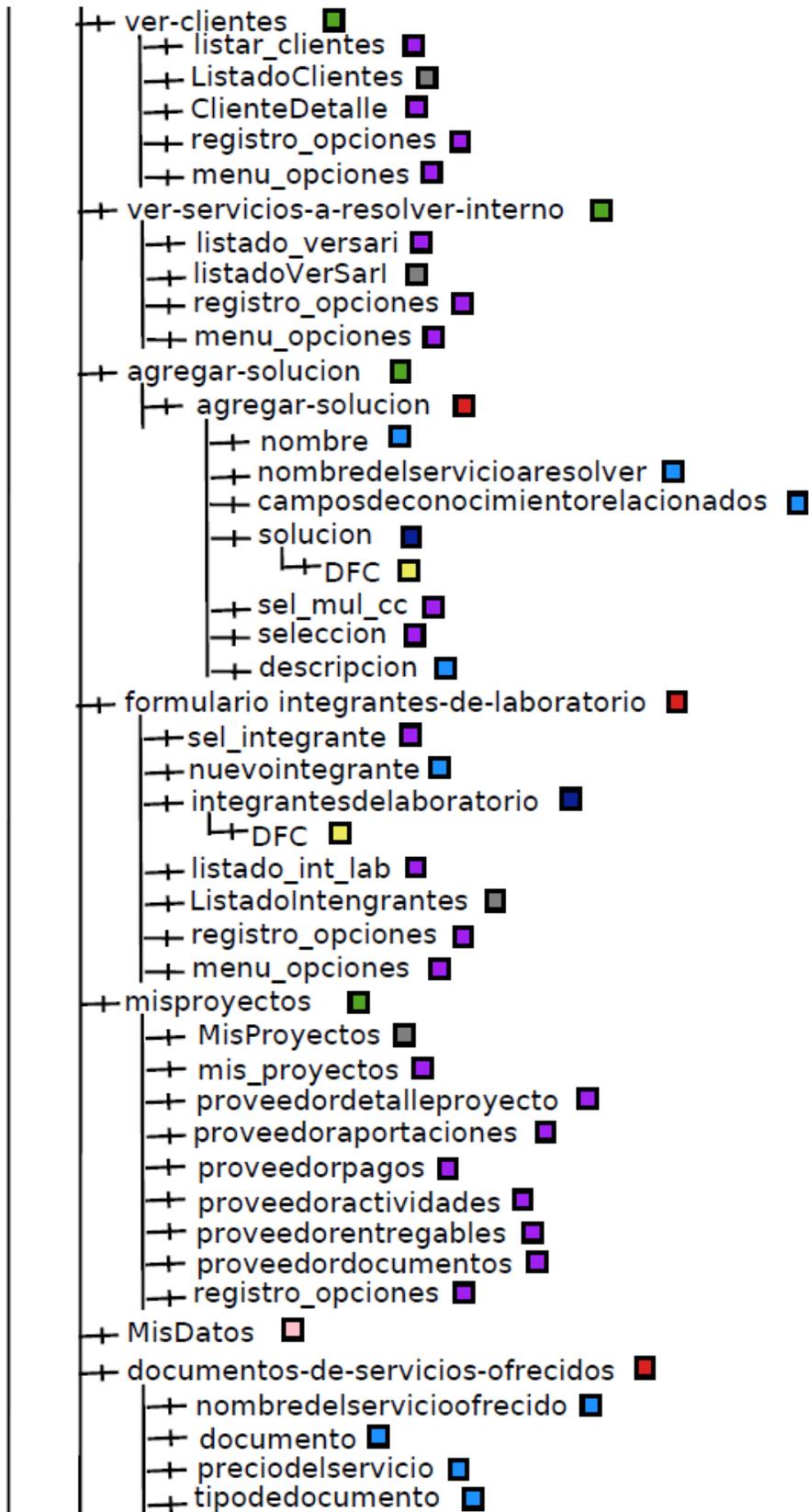


Figura A.11: Estructura del Sitio en Plone

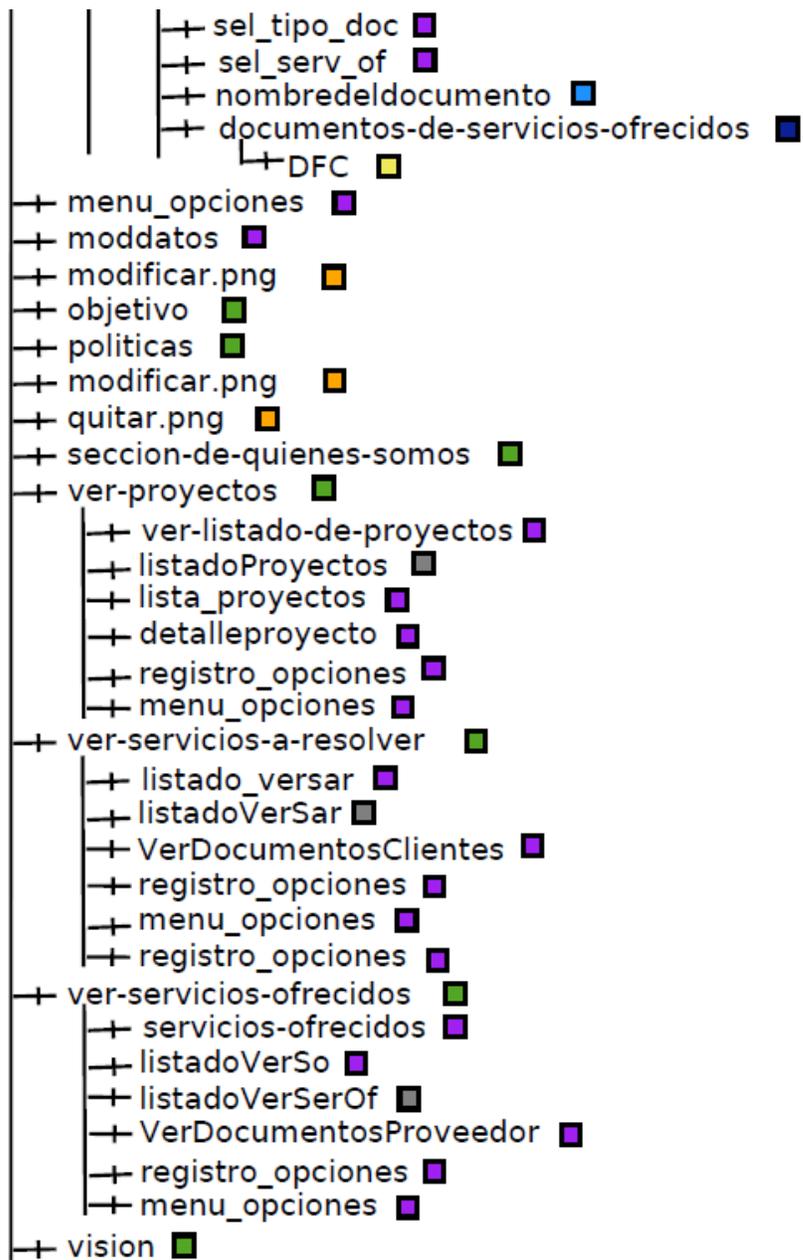


Figura A.12: Estructura del Sitio en Plone

| color - RGB | prueba | vista | color - RGB | prueba | vista |
|-----------------------|-----------------|---|-----------------------|---------------|---|
| aqua 00FFFF | img_rdbms_pro_1 |  | darkgray A9A9A9 | img_cms_pro_1 |  |
| blue 0000FF | img_rdbms_pro_2 |  | darkgreen 006400 | img_cms_pro_2 |  |
| aquamarine 7FFFD4 | img_rdbms_pro_3 |  | darkkhaki BDB76B | img_cms_pro_3 |  |
| blueviolet 8A2BE2 | img_rdbms_pro_4 |  | darkmagenta 8B008B | img_cms_pro_4 |  |
| brown A52A2A | img_rdbms_pro_5 |  | darkolivegreen 556B2F | img_cms_pro_5 |  |
| burlywood DEB887 | img_rdbms_so_1 |  | darkorange FF8C00 | img_cms_so_1 |  |
| cadetblue 5F9EA0 | img_rdbms_so_2 |  | darkorchid 9932CC | img_cms_so_2 |  |
| chartreuse 7FFF00 | img_rdbms_so_3 |  | darkred 8B0000 | img_cms_so_3 |  |
| chocolate D2691E | img_rdbms_so_4 |  | darksalmon E9967A | img_cms_so_4 |  |
| coral FF7F50 | img_rdbms_so_5 |  | darkslateblue 483D8B | img_cms_so_5 |  |
| cornflowerblue 6495ED | img_rdbms_sar_1 |  | darkseagreen 8FBC8F | img_cms_sar_1 |  |
| crimson DC143C | img_rdbms_sar_2 |  | darkslategray 2F4F4F | img_cms_sar_2 |  |
| darkblue 00008B | img_rdbms_sar_3 |  | deeppink FF1493 | img_cms_sar_3 |  |
| darkcyan 008B8B | img_rdbms_sar_4 |  | firebrick B22222 | img_cms_sar_4 |  |
| darkgoldenrod B8860B | img_rdbms_sar_5 |  | dodgerblue 1E90FF | img_cms_sar_5 |  |
| brown A52A2A | img_rdbms_pro |  | darkolivegreen 556B2F | img_cms_pro |  |
| coral FF7F50 | img_rdbms_so |  | darkslateblue 483D8B | img_cms_so |  |
| darkgoldenrod B8860B | img_rdbms_sar |  | dodgerblue 1E90FF | img_cms_sar |  |
| gold FFD700 | txt_rdbms_pro_1 |  | royalblue 4169E1 | txt_cms_pro_1 |  |
| fuchsia FF00FF | txt_rdbms_pro_2 |  | sandybrown F4A460 | txt_cms_pro_2 |  |
| forestgreen 228B22 | txt_rdbms_pro_3 |  | sienna A0522D | txt_cms_pro_3 |  |
| greenyellow 9ACD32 | txt_rdbms_pro_4 |  | teal 008080 | txt_cms_pro_4 |  |
| indigo 4B0082 | txt_rdbms_pro_5 |  | yellow FFFF00 | txt_cms_pro_5 |  |
| indianred CD5C5C | txt_rdbms_so_1 |  | yellowgreen 9ACD32 | txt_cms_so_1 |  |
| lightsalmon FFA07A | txt_rdbms_so_2 |  | springgreen 00FF7F | txt_cms_so_2 |  |
| lightpink FFB6C1 | txt_rdbms_so_3 |  | thistle D8BFD8 | txt_cms_so_3 |  |
| magenta FF00FF | txt_rdbms_so_4 |  | turquoise 40E0D0 | txt_cms_so_4 |  |
| maroon 800000 | txt_rdbms_so_5 |  | tomato FF6347 | txt_cms_so_5 |  |
| midnightblue 191970 | txt_rdbms_sar_1 |  | rosybrown BC8F8F | txt_cms_sar_1 |  |
| olive 808000 | txt_rdbms_sar_2 |  | powderblue B0E0E6 | txt_cms_sar_2 |  |
| orangered FF4500 | txt_rdbms_sar_3 |  | plum DDA0DD | txt_cms_sar_3 |  |
| olivedrab 6B8E23 | txt_rdbms_sar_4 |  | peru CD853F | txt_cms_sar_4 |  |
| red FF0000 | txt_rdbms_sar_5 |  | palevioletred DB7093 | txt_cms_sar_5 |  |
| indigo 4B0082 | txt_rdbms_pro |  | darkgoldenrod B8860B | txt_cms_pro |  |
| maroon 800000 | txt_rdbms_so |  | tomato FF6347 | txt_cms_so |  |
| red FF0000 | txt_rdbms_sar |  | palevioletred DB7093 | txt_cms_sar |  |

Figura A.13: Identificación gráfica de las pruebas de funcionalidad

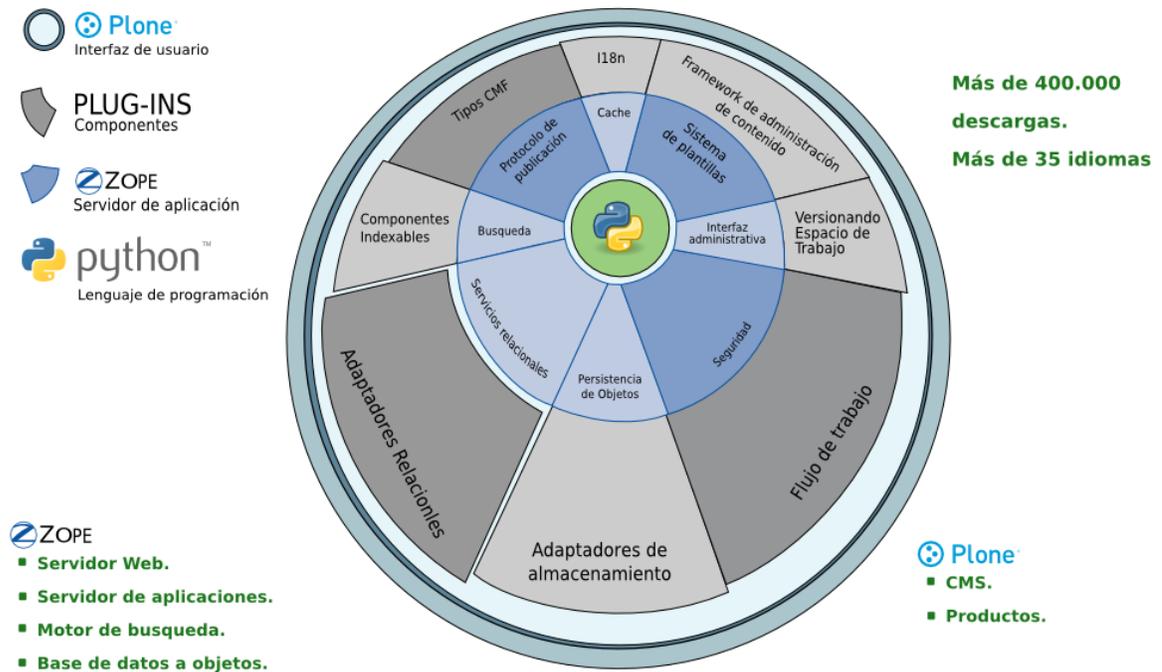


Figura A.14: Estructura Física de Plone. Obtenido del sitio web www.plone.org