



Universidad Central de Venezuela  
Facultad de Ciencias  
Escuela de Computación

**Prototipo para un Archivo de Documentos  
Digitales provenientes de la Web.  
Panel de Control, Virtualización  
y Programación en la Nube**

Trabajo Especial de Grado presentado ante la Ilustre  
Universidad Central de Venezuela

Por la Bachiller:

**Arévalo S., Vanessa C.**

Para optar al título de Licenciado en Computación

Tutor:

**Prof. Andrés Sanoja**

Caracas, mayo 2011



---

## Resumen

Es muy conocido que la Web es una fuente de datos con gran cantidad de información que es utilizada por la mayoría de las personas, contenido que es necesario mantener integrado por diversas razones. Basado en esto, se propone un proyecto compuesto por una serie de herramientas que apoyen el intercambio de datos, con la finalidad de contar con una base de datos hipertextual así como patrimonial orientada a Venezuela o con información relacionada con el país. El proyecto consta de los siguientes componentes: Portal, Panel de Control (Cpanel), el Modulo de Control, Manejador de Trabajos en Background (Robot); Analizador de Documentos, Generación de Vistas Previas en varios formatos y Spiders, estos tres últimos soportados por una plataforma de Cloud Computing. Finalmente los componentes para el almacenamiento y gestión de datos soportado por un repositorio de documentos siguiendo el enfoque No-Sql, y un Datawarehouse para la consulta y gestión de versiones históricas. Particularmente, el desarrollo del Panel de Control es con el fin de gestionar las direcciones web a incluir en el repositorio, manejo de usuarios, configuraciones, grupos entre otras funcionalidades. Además se realizaron las siguientes actividades: instalación y gestión de ambientes de virtualización para una plataforma de Cloud Computing, procesamiento de tareas en segundo plano, así como también un filtro bayesiano para clasificación semiautomática de documentos. El resultado fue el de una aplicación con las funcionalidades mencionadas anteriormente. Este enfoque es un prototipo como paso inicial para la construcción de un Archivo de Documentos Digitales provenientes de la Web como patrimonio nacional mediante la preservación de la información, algo que sería de gran utilidad y brindaría beneficios tanto a las generaciones actuales como a las futuras, sobre todo el de mantener y esparcir el conocimiento.

**Palabras Clave** Cloud Computing, Web Archiving, Datawarehouse, Virtualización, Aplicaciones Web



---

## Índice general

<b>Introducción</b> .....	1
<b>1. Propuesta de TEG</b> .....	5
1.1. Planteamiento del Problema .....	5
1.2. Objetivos .....	6
1.2.1. Objetivo General .....	6
1.2.2. Objetivos Específicos .....	6
1.3. Solución/Arquitectura .....	7
1.4. Diseñó Lógico .....	7
1.5. Propuesta de la solución .....	8
1.6. Alcance .....	9
<b>2. Marco Conceptual</b> .....	11
2.1. Preservación Digital .....	11
2.1.1. Perspectivas de Gestión .....	12
2.1.2. Política adecuada de Preservación .....	15
2.2. Internet Archive .....	15
2.3. The Wayback Machine .....	16
2.3.1. ¿Cómo funciona?[18] .....	16
2.3.2. ¿Por que Wayback? .....	17

2.3.3. Formatos de Archivos .....	18
2.4. Ruby on Rails .....	18
2.4.1. Lenguaje de Programación Ruby .....	18
2.4.2. Gemas de Ruby (Rubygems) .....	19
2.4.3. Framework Rails .....	19
2.4.4. Ventajas de Rails .....	19
2.4.5. Filosofía de Rails: Principios de Desarrollo .....	20
2.4.6. El Patrón de Diseño Modelo - Vista - Controlador (MVC) ...	22
2.4.7. ActiveRecord .....	24
2.4.8. Action Controller .....	25
2.4.9. Action View .....	25
2.5. HTML Dinámico .....	26
2.5.1. Definición .....	26
2.5.2. jQuery .....	27
2.6. Virtualización .....	28
2.6.1. ¿Cómo funciona? .....	28
2.6.2. Ventajas .....	29
2.6.3. Tipos de Virtualización[37] .....	30
2.7. Computación en la Nube .....	32
2.7.1. Evolución de Cloud Computing .....	32
2.7.2. Beneficios y riesgos .....	33
2.7.3. Implementación .....	33
2.7.4. Niveles de servicio .....	36
<b>3. Marco Metodológico .....</b>	<b>39</b>
3.1. Programación Extrema(XP) .....	39
3.2. Adaptación del Proceso de Desarrollo XP .....	40
3.2.1. Historias de Usuario .....	41

	Índice general	IX
3.2.2. Fases de desarrollo .....		41
3.2.3. Análisis General del Sistema .....		44
3.2.4. Metáfora del Sistema .....		48
<b>4. Marco Aplicativo .....</b>		<b>49</b>
4.1. Plan de Iteración .....		49
4.1.1. Iteración 0 .....		49
4.1.2. Iteración 1 .....		53
4.1.3. Iteración 2 .....		56
4.1.4. Iteración 3 .....		59
4.1.5. Iteración 4 .....		63
4.1.6. Iteración 5 .....		67
4.1.7. Iteración 6 .....		71
4.1.8. Iteración 7 .....		75
4.1.9. Iteración 8 .....		77
4.2. Resultados .....		79
<b>Conclusiones y Recomendaciones .....</b>		<b>86</b>
<b>Referencias .....</b>		<b>93</b>





---

## Índice de figuras

1.1. Visión General de la Solución/Arquitectura .....	7
1.2. Diseñó Lógico del Sistema .....	8
2.1. Algunos de los servidores del Archivo Internet, en su ubicación original en San Francisco, hacia 2002[11]. .....	16
2.2. Página de la UCV el día 5 de mayo de 2009 .....	17
2.3. Implementación en Rails del Patrón MVC .....	23
2.4. Mapeo relacional de objetos (ORM) .....	24
2.5. Emulación del Hardware. ....	30
2.6. La virtualización completa utiliza un hipervisor para compartir el hardware subyacente .....	31
2.7. La paravirtualización comparte el proceso con el SO alojado (Guest OS) .....	31
2.8. La virtualización en el nivel del sistema operativo aísla a los servidores .....	31
3.1. Desarrollo en XP .....	41
3.2. Metáfora del Sistema .....	48
4.1. Modelo de datos inicial .....	51
4.2. Formulario para crear nuevo grupo. ....	54

4.3. Código de crear nuevo proceso. ....	54
4.4. Componente javascript para agregar direcciones web a un proceso. ....	57
4.5. Código para agregar o eliminar mediante ajax las direcciones web. ....	57
4.6. Vista menú principal .....	60
4.7. Código de autenticación .....	64
4.8. Código de cambiar contraseña .....	65
4.9. Modelo correo .....	66
4.10. Arquitectura de BackgroundDRb[39] .....	68
4.11. Modelo donde se crea la imagen thumbnail .....	69
4.12. Workers creado con BackgroundDRb .....	70
4.13. Modelo de Datos del Controlador Central .....	72
4.14. Menú principal .....	79
4.15. Lista de grupos creados .....	79
4.16. Creación de grupo nuevo .....	80
4.17. Dirección agregada a un grupo nuevo .....	80
4.18. Lista de direcciones que contiene un grupo .....	81
4.19. Opción de modificar grupo de direcciones .....	81
4.20. Lista de configuraciones .....	82
4.21. Lista de módulos de la aplicación .....	82
4.22. Modificar módulo seleccionado .....	83
4.23. Lista de usuarios registrados .....	84
4.24. Agregar nuevo usuario .....	84
4.25. Cambiar clave de usuario .....	85
4.26. Arquitectura detalla del proyecto .....	91

---

## Índice de tablas

2.1. Visión general de cómo Rails implementa el framework de diseño	
MVC .....	23
3.1. Formato de Registro para Historias de Usuario .....	41
3.2. Formato de planificación de cada iteración .....	42
3.3. Formato de Pruebas de Aceptación .....	43



---

## Introducción

En los últimos tiempos se ha visto como la automatización de sistemas y procesos son un elemento clave para la evolución de las organizaciones. Igualmente, gracias a la gran influencia que ha tenido internet se han logrado muchos avances en las Tecnologías de Información. Debido a estos avances, la Web se ha convertido en una fuente de datos donde se puede encontrar una gran cantidad de información, lo que se traduce en un importante recurso que la mayoría de las personas puede consultar. Sin embargo, mucho de éste contenido se ha perdido en el tiempo, debido a la falta de políticas y estructuras adecuadas para mantener integrada la información.

Así como existen grandes bibliotecas donde se conservan libros de muchos años, se hace necesario almacenar todo el contenido digital Web y preservarlo en el tiempo. Alcanzar esta meta requiere de las tecnologías actuales así como también el poder adaptarlas eficientemente a las venideras, de tal forma que se pueda garantizar la preservación de un histórico de las páginas Web y así evitar que cualquier tipo de información desaparezca.

Actualmente en algunos países existen estas herramientas de preservación digital. De hecho, la UNESCO establece que se debe ayudar a la preservación y difusión de conocimientos.



## Propuesta de TEG

Este capítulo se enfoca en la propuesta de Trabajo Especial de Grado. Básicamente se propone la solución a un problema, se establecen objetivos, tanto general como específicos, y el alcance que se va a lograr a través de ellos. También se muestra una visión general de la Solución/Arquitectura donde se definen los componentes principales del sistema propuesto. Esto va a permitir entender un poco más la lógica de la solución.

### 1.1. Planteamiento del Problema

Actualmente existen gran cantidad de aplicaciones Web en todo el mundo que diariamente pueden llegar a cambiar en su totalidad o sólo pequeñas partes de ellas. Específicamente en el ámbito nacional se puede llegar a encontrar muchas páginas que cambian constantemente. A pesar de ello, no se cuentan con políticas necesarias que incluyen esfuerzos por mantener y preservar todo la información proveniente de la Web. La utilidad y beneficios que se pudiesen obtener de esto serían de gran envergadura para las generaciones tanto actuales como futuras.

Se requiere la construcción de un sistema que permita obtener información de la internet enmarcada en el ámbito nacional, manteniendo versiones históricas de cada una de las modificaciones y actualizaciones que se ejecuten en las páginas

web con el fin de conservar repositorios que integren toda la información y sea accesible al público en general, evitando así que mucha de esa información se pierda o desaparezca.

Considerando tales exigencias, se ve la necesidad de crear un prototipo como paso inicial y con miras a su consolidación. Para lograr dicha meta, se dispone de tecnologías que se pueden aplicar al desarrollo de este sistema.

## **1.2. Objetivos**

En esta sección se definen los objetivos que surgieron en cuanto a la investigación realizada y con los cuales se pretende lograr el alcance a determinar.

### **1.2.1. Objetivo General**

Desarrollar una aplicación que permita gestionar la actualización de páginas y/o sitios Web de interés para el repositorio así como también aplicar herramientas de virtualización y la implementación de *Cloud Computing*.

### **1.2.2. Objetivos Específicos**

- Definir todas las funcionalidades que va a incluir la aplicación.
- Desarrollar un prototipo de interfaz gráfica que contengan las funcionalidades requeridas.
- Diseñar un modelo de datos para el manejo de la información contenida en la aplicación.
- Desarrollar el prototipo del sistema utilizando las tecnologías seleccionadas.
- Desarrollar las funcionalidades del prototipo del sistema orientada a la computación en al nube.



- Utilizar el método eXtreme Programming para guiar el desarrollo.
- Integrar la aplicación con los demás subsistemas del proyecto en general.
- Realizar pruebas funcionales con los demás subsistemas.

### 1.3. Solución/Arquitectura

Es importante el diseño de una arquitectura que permita entender el alcance y la solución que se requiere para el sistema final. En la figura 1.1 se presenta la arquitectura general donde se aprecian los componentes que constituyen el sistema.

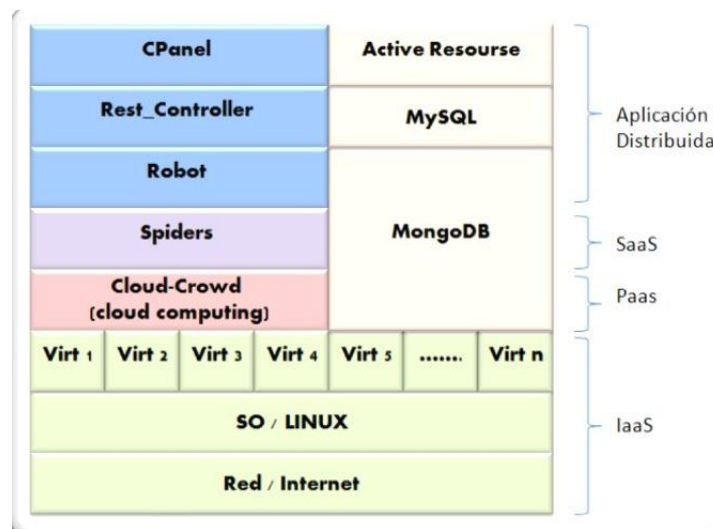
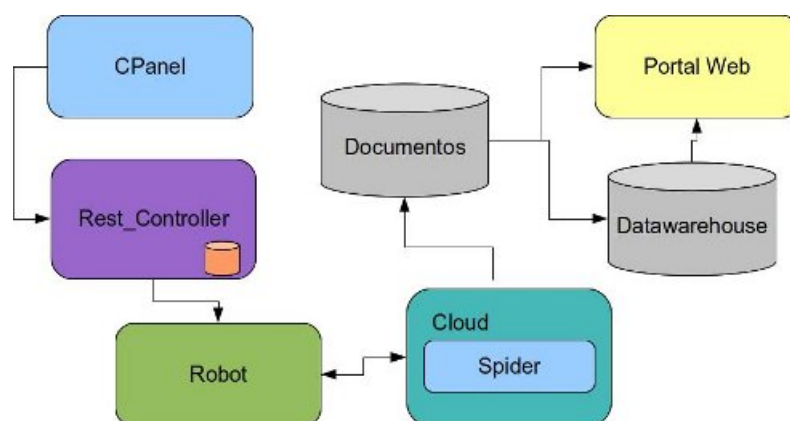


Figura 1.1. Visión General de la Solución/Arquitectura

### 1.4. Diseño Lógico

El sistema se compone de un Portal, Panel de Control (Cpanel), el Modulo de Control, Manejador de Trabajos en Background (Robot); Analizador de Documentos, Generación de Vistas Previas en varios formatos y Spiders, estos tres

últimos soportados por una plataforma basada en Cloud Computing. Finalmente los componentes para el almacenamiento y gestión de datos soportado por un repositorio de documentos siguiendo el enfoque No-Sql, y un Datawarehouse para la consulta y gestión de versiones históricas. En la figura 1.2 se pueden apreciar estos componentes.



**Figura 1.2.** Diseño Lógico del Sistema

## 1.5. Propuesta de la solución

Luego de conocer el problema se requiere de una solución que cumpla con los objetivos planteados mediante una serie de actividades que se presentan a continuación. Cabe destacar que el presente trabajo se enfoca principalmente en el desarrollo del Panel de Control (CPanel).

Se diseñó y desarrolló una aplicación que va a permitir la gestión de las direcciones web. Entre las funcionalidades principales se tienen el crear un grupo el cual va a contener una colección de direcciones que van a ser agregadas por el usuario. Un grupo va a ser determinado por las características comunes que existan entre las direcciones que se asocian a él (por ejemplo, si el nombre del Grupo es UCV entonces este va a contener las páginas web que sean de la universidad). Queda

de parte del usuario como asociar las direcciones a un grupo. Además, se puede hacer la actualización de un grupo agregando nuevas direcciones o eliminando las ya existentes. También, el usuario tiene la posibilidad de eliminar un grupo completo de direcciones si así lo requiere. Sumado a esto, la aplicación permite la gestión de usuarios, de configuraciones, módulos entre otras funcionalidades. Además, las direcciones web se les asocia una serie de etiquetas representativas de los temas a los cuales se puede referir de acuerdo al contenido de su página principal.

Para darle más feedback al usuario, mientras agrega las direcciones se va a generar una imagen preview de la página principal de cada una.

También, para aprovechar las herramientas existentes actualmente y explotar los beneficios que pueden aportar a las diferentes actividades implementadas, se procedió a la instalación y configuración de máquinas virtuales para implementar una plataforma Cloud Computing.

## **1.6. Alcance**

- Construcción de una interfaz de usuario que contenga todas las funcionalidades requeridas por el sistema para la gestión de grupos de direcciones provenientes de la Web.
- Permitir, mediante la aplicación, la gestión de usuarios, configuración entre otros.
- Configurar e implementar ambiente para la ejecución de tareas en background.
- Implementación de un clasificador semiautomático de cada dirección web.
- Configurar y poner en funcionamiento máquinas virtuales.



## **Marco Conceptual**

Este capítulo muestra las bases teóricas necesarias que fundamenta el tema central del Trabajo Especial de Grado. Uno de los tópicos más importantes es lo que actualmente se ha venido ya investigando y aplicando en algunos países como lo es la preservación digital, así como lo que se ha estado implementando. También se tratará sobre las tecnologías a emplear para la construcción y desarrollo tales como el lenguaje de programación Ruby y el framework Rails. Además, las herramientas de virtualización y plataformas de Cloud Computing.

### **2.1. Preservación Digital**

Se puede entender por preservación digital la preservación de los artefactos físicos mediante su digitalización, pero también la preservación de los propios recursos digitales[3]. Preservación digital designa los procesos a que se recurre con objeto de conservar información y cualquier otro tipo de patrimonio existente en forma digital, este término no se refiere al uso de imágenes digitales o de técnicas de captura para hacer copias de elementos no digitales, inclusive si éstas se realizan con fines de preservación, sino también la realización de copias digitales (también denominada digitalización) que puede perfectamente producir materiales de patrimonio digital que necesiten ser preservados.[38].

### **2.1.1. Perspectivas de Gestión**

Según [38], se tienen las siguientes perspectivas:

#### **Comprender el patrimonio digital**

El patrimonio digital se compone de elementos informáticos de valor perdurable, dignos de ser conservados para las generaciones futuras, y que proceden de comunidades, industrias, sectores y regiones diferentes. No todos los elementos digitales poseen un valor permanente, pero, cuando es el caso, su preservación debe enfocarse con una actitud dinámica si se desea mantener la continuidad del patrimonio digital.

#### **Comprender la preservación digital**

La preservación digital consiste en los procesos destinados a garantizar la accesibilidad permanente de los objetos digitales. Para ello, es necesario encontrar las maneras de representar lo que se había presentado originalmente a los usuarios mediante un conjunto de equipos y programas informáticos que permiten procesar los datos. Para lograrlo, es necesario que la comprensión y la gestión de los objetos digitales se realice considerándolos desde cuatro puntos de vista: como fenómenos físicos, como codificaciones lógicas, como objetos conceptuales comprensibles para el ser humano y como conjuntos de elementos esenciales que deben ser preservados para ofrecer a los futuros usuarios lo esencial del objeto.

#### **Comprender los programas de preservación digital**

Los programas completos deben tomar el control de los objetos digitales apropiados y garantizar que permanezcan comprensibles y utilizables como copias auténticas. Por lo general, ello supone transferir los materiales, correctamente preparados, junto con la documentación o los metadatos asociados a ellos, a

un sistema archivístico de almacenamiento digital de algún tipo, en el que puedan ser procesados para hacer frente a las amenazas de pérdida de datos y cambios tecnológicos. También se han descrito las características o atributos de los programas en los que se puede confiar para lograr una preservación digital permanente en lo referente a responsabilidad, viabilidad, durabilidad, adecuación técnica, seguridad y precisión.

### **Aceptar responsabilidades**

La preservación digital se hará realidad únicamente si las personas físicas y morales aceptan asumir la responsabilidad que implica. Forma parte de esta aceptación tomar las disposiciones necesarias para poner en práctica las medidas de preservación definidas en las presentes directrices, así como los nuevos procesos que convengan según vayan apareciendo. Si bien es cierto que se requieren disposiciones completas y totalmente fiables, en muchos casos puede que no sea posible aplicarlas inmediatamente, en cuyo caso se puede recurrir a programas más modestos, pero seguros, que pueden aportar una valiosa ayuda.

### **Gestión de los programas de preservación digital**

Los programas de preservación requieren una gestión adecuada, que a menudo utiliza habilidades genéricas, como la adecuación de los programas a las prioridades y circunstancias del caso y el tomar las decisiones correctas en el momento oportuno. La gestión de los programas de preservación digital presenta ciertas características particulares en relación con la naturaleza evolutiva de los programas, las diferentes partes interesadas y las consecuencias que pueden acarrear a largo plazo las decisiones que se adopten.

## Trabajar en equipo

Hay buenas razones técnicas, económicas y políticas para que los programas de preservación cooperen entre sí. Las decisiones de colaboración deben basarse en la evaluación de los beneficios esperados y de los costos correspondientes. Existen varias posibilidades para buscar socios y decidir en qué se basará la relación y los marcos estructurales que pueden aplicarse. Una cooperación lograda suele ser fruto de la cuidadosa atención dedicada a estas decisiones y del esfuerzo necesario para llevar a bien la colaboración en la práctica.

Los asuntos que se deben tratar en la preservación digital incluyen[25]:

- Mantener la fiabilidad física de los archivos de imagen, los metadatos complementarios, textos y programas (por ejemplo: asegurarse de que el medio de almacenamiento es confiable, con copias de seguridad (back-ups), mantener la infraestructura de hardware y software necesaria para almacenar y proporcionar acceso a la colección;
- Asegurar el uso de la colección de imágenes digitales en forma continuada (por ejemplo: mantener una interfase de usuario actualizada, permitir a los usuarios recuperar y manipular información para poder satisfacer sus necesidades de información);
- Mantener la seguridad de la colección (por ejemplo: implementar estrategias para controlar la alteración no autorizada de la colección, desarrollar y mantener un programa de gestión de derechos para servicios con cargo).

Se puede decir que es necesario tanto mantener los medios físicos, la forma en que se almacena y todo el material digital en el tiempo, tomando en cuenta el avance de la tecnología a través de los años. También es importante que de alguna manera, estos recursos puedan ser buscados y recuperados de una manera fácil porque de nada vale matenerlos si no hay una forma de poder obtenerlos.



### 2.1.2. Política adecuada de Preservación

Una de las formas de almacenar la información es que sea en formatos ampliamente usados actualmente. Esto aumenta la probabilidad de que cuando un formato se vuelva obsoleto aún existan programas para su conversión. XML, HTML y PDF son ejemplos de estos. Otra interesante sugerencia es crear un archivo que contenga las definiciones de los formatos, estándares de metadatos, protocolos y otros elementos constructivos fundamentales de las bibliotecas digitales. Si los formatos y los esquemas de codificación son preservados, la mayoría de la información puede ser descifrada posteriormente[3].

## 2.2. Internet Archive

El Archivo de Internet fue fundado en 1996 con el intento de preservar lo que es posiblemente el archivo de crecimiento más rápido que en la expresión de la humanidad se haya creado. El estado actual de la tecnología digital y el internet hace factible para el Archivo alcanzar la misión declarada de acceso universal para el conocimiento de la humanidad[11].

Este archivo almacena contenido web, todo tipo de material digital tales como música, vídeo, software, imágenes y libros. También cuenta con un archivo de páginas web. Provee acceso no re restrictivo y además se puede subir y descargar material sin costo alguno.

Fue construido como una biblioteca de Internet. Su propósito incluye ofrecer acceso permanente para investigadores, historiadores, académicos, personas con discapacidad y al público en general a colecciones históricas que existen en formato digital. Está localizado en San Francisco, el Archivo ha estado recibiendo donaciones de datos desde Alexa Internet y otros[19].



**Figura 2.1.** Algunos de los servidores del Archivo Internet, en su ubicación original en San Francisco, hacia 2002[11].

## 2.3. The Wayback Machine

Es un archivo histórico que almacena páginas web desde hace más de una década. Rastrea y guarda sitios web completos, imágenes, documentos y todo lo que pueda ser tomado en cuenta para mantener en el tiempo. El Archivo de Internet es el actual propietario de éste.

### 2.3.1. ¿Cómo funciona?[18]

La nueva Wayback Machine funciona de una manera tan simple como la anterior: basta introducir la dirección (URL) de una página web cualquiera para ver cuál es la última copia que hay guardada en el archivo.

El diseño de la portada se ha simplificado y ahora es más claro y elegante. A partir de aquí, las cosas también cambian un poco: en la versión anterior se podía acceder a una lista a modo de calendario que mostraba los momentos clave en que la web en cuestión había sido rastreada y “guardada” en el archivo.

Ahora hay una nueva barra con más estilo en la parte superior de la pantalla, que permite ver gráficamente esos momentos a lo largo del tiempo. La longitud de las barras del gráfico indica en qué meses se hicieron más copias, lo cual está normalmente asociado a una mayor actividad en la web en cuestión: más

grandes, más actualizaciones. En la figura 2.2 se puede observar la página de la Universidad Central de Venezuela en la actualización del 5 de mayo de 2009.



Figura 2.2. Página de la UCV el día 5 de mayo de 2009

### 2.3.2. ¿Por que Wayback?

Existen muchos usos para el archivo de la Wayback Machine [28]. En un nivel básico es un gran recurso para encontrar información de páginas cuando o el host no está disponible. Cuando se encuentra con un “404 not found” o un mensaje similar en la Web, simplemente se accede al Wayback Machine para encontrar una copia de la página como usualmente se ve.

Las implicaciones históricas son inmensas. Los investigadores de historia pueden ver porciones significativas de la Web como existían en algunos tiempos desde 1996 hasta la actualidad. Las ventajas históricas van más allá que simplemente la investigación histórica. Los expertos de negocio pueden buscar los planes de compañías con fracaso del negocio. Los empresarios pueden investigar las páginas de los estudiantes que solicitan empleo.

Son muchas la ventajas que se pueden obtener de este proyecto que cada día aumenta en disponibilidad de páginas donde se puede encontrar información de cualquier tipo sin depender de los servidores donde se encuentren o que por alguna razón fue eliminada de la Web.

### **2.3.3. Formatos de Archivos**

El formulario de búsqueda avanzada también señala que la Wayback Machine provee acceso a más que simplemente páginas[28]. El límite de Tipos de Archivo incluye seis (6) formatos: Imágenes, Audio, Video, Binario, Texto y PDF. Escogiendo uno de estos archivos y colocando la ruta URL (con un nombre completo de host), los resultados van a incluir todos los tipos de archivo de ese formato desde ese host en el archivo. Cada registro de un tipo de archivo individual tiene una URL única, pero si el buscador no conoce la dirección completa, este límite ayuda a indentificarlas. Además, puede ser usado como una herramienta para contar el número de un tipo de archivo específico en un servidor específico.

## **2.4. Ruby on Rails**

Ruby on Rails es un framework de desarrollo de aplicaciones Web el cual fue construido usando el lenguaje de programación Ruby. Se dice que por sus características únicas y filosofía de diseño permite la simplificación e integración de funcionalidad así como un fácil desarrollo y mantenimiento.

### **2.4.1. Lenguaje de Programación Ruby**

Ruby es un lenguaje de secuencia de comandos, orientado a objetos y de código abierto que fue creado por Yukihiro Matsumoto en los comienzos de 1990. Ruby

hace la programación más flexible e intuitiva, y con esto, se puede escribir código entendible tanto por los humanos así como por las máquinas[23].

Se dice que Ruby está creado para mayor productividad, *de hecho Matsumoto afirma que su primer objetivo en cuanto al diseño de Ruby era crear un lenguaje que él mismo disfrutara usando, minimizando el trabajo del programador y las confusiones posibles*[15].

#### **2.4.2. Gemas de Ruby (Rubygems)**

Es una herramienta que provee de librerías y programas escritos en Ruby mediante un estándar establecido. Estos son llamados gemas. Todas las utilidades de Ruby se instalan a través de esta herramienta y también proporciona flexibilidad a la hora de utilizar código de programas ya construidos.

#### **2.4.3. Framework Rails**

Rails es un framework de aplicación compuesto de varias librerías las cuales proporcionan un marco completo para la construcción de aplicaciones Web. Se puede usar para construir cualquier aplicación, por ejemplo: blogs, wikis, aplicaciones de seguimiento de proyecto, aplicaciones de galería de fotos, aplicaciones de redes sociales y sitios de compra.

#### **2.4.4. Ventajas de Rails**

Son muchos los motivos por los cuales Rails se ha convertido en un framework popular. Algunos de ellos se listan a continuación[14]:

- Rails permite construir aplicaciones Web de forma rápida y sencilla al hacer la mayor parte del trabajo, común a la mayoría de las aplicaciones Web.
- Rails hace el desarrollo divertido y fácil para los desarrolladores.

- Rails provee un framework completo. Esto significa que provee de todas las piezas necesarias para construir una aplicación Web completa en un solo paquete.
- Se puede definir y acceder a la base de datos, usar Ruby embebido en las plantillas y código Ruby en la lógica del negocio. Esto libera a los desarrolladores en enfocarse a aprender un lenguaje y conocerlo bien.

#### 2.4.5. Filosofía de Rails: Principios de Desarrollo

Además de los motivos anteriormente expuestos, Rails apoya varios principios de software o paradigmas que lo hacen resaltar sobre otros frameworks de desarrollo web. Estos principios son[23]:

- Convención sobre configuración
- Don't Repeat Yourself (DRY)

Por todos estos principios se dice que Ruby on Rails es un framework que hace ahorrar tiempo y esfuerzo a los desarrolladores[17].

#### Convención sobre Configuración(CoC)

Muchos otros frameworks requieren pasar por un largo proceso de configuración antes de comenzar con una aplicación simple. Por ejemplo la configuración de información es usualmente almacenada en archivos XML los cuales pueden llegar a ser bastante largos y engorrosos de mantener.

Por lo tanto se propuso crear Rails de tal manera que no se necesite excesiva configuración, como las convenciones estándar que son seguidas. El resultado es que no se requiere una configuración de archivos largos. Por ejemplo, hay archivo que es usado para establecer una conexión con la base de datos (database.yml).

La configuración específica de Rails es mostrada en el siguiente bloque de código:

```
adapter: mysql
myblog_development
username: admin
password: pswd123
host: localhost
```

Otras convenciones que son preescritas por Rails incluyen el nombrar los items relacionados con la base de datos, proceso por el cual los controladores encuentran sus correspondientes modelos y vistas.

### **Don't Repeat Yourself (DRY)**

Este principio ayuda a reducir la duplicación de código, por lo tanto incrementa productividad, reduciendo errores y facilitando el mantenimiento y reconstrucción. Esto quiere decir que cuando se decide cambiar el comportamiento de una aplicación basada en este principio no se debe necesitar la modificación de código en más de una localidad.

Un ejemplo de como Rails apoya este principio es que, no fuerza a repetir la definición del esquema de base de datos (el cual especifica como está almacenada la estructura de los datos de la aplicación) dentro de la aplicación. Rails considera la base de datos para ser la fuente autorizada de información acerca del almacenamiento de los datos, y es suficientemente inteligente para solicitar a ésta cualquier información que pueda necesitar para asegurar que el tratamiento de la data sea correcta.

Una de las cosas más relevantes de este framework es que el programador se enfoca en un sólo lenguaje y no en varios como se hace con la mayoría de los lenguajes de desarrollo Web, por ejemplo para manipular base de datos se necesita

código SQL en conjunto con el lenguaje de programación Web. En cambio cuando se desarrollan aplicaciones en Rails casi todo es construido con código Ruby, tanto la definición de la base de datos como el acceso a ésta. De esta manera sólo se centra en aprender y conocer bien el lenguaje.

#### 2.4.6. El Patrón de Diseño Modelo - Vista - Controlador (MVC)

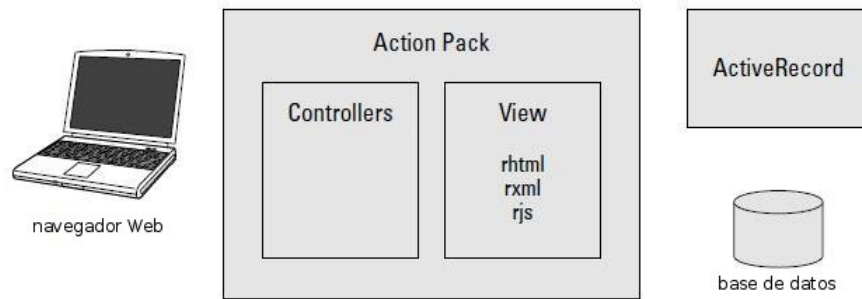
Como es conocido, una de las cosas más importante de una aplicación Web es que sea fácil de comprender y mantener. Esto se puede lograr mediante la implementación de una arquitectura de desarrollo de software que también va a influir en la calidad de la aplicación. La arquitectura de desarrollo Modelo - Vista - Controlador (MVC) describe el método de estructuración de código para lograr una “separación de interés” entre datos (el modelo), comportamiento (el controlador) y la interacción del usuario (la vista).

En términos generales el principio MVC divide el trabajo de una aplicación en tres subsistemas independientes pero estrechamente cooperativos. El Modelo, la Vista y el Controlador los cuales son descritos como sigue[33]:

- **Modelo:** es el responsable de mantener el estado de la aplicación. A veces el estado es transiente, esperando simplemente un conjunto de interacciones con el usuario; es permanente y va a ser almacenado externamente a la aplicación, frecuentemente en una base de datos.
- **Vista:** es responsable de generar una interfaz para el usuario, basada normalmente en el modelo de los datos.
- **Controlador:** orquesta la aplicación. Los controladores reciben eventos desde el mundo externo (normalmente entrada del usuario), interacciona con el modelo y despliega una vista adecuada al usuario.

Particularmente Rails implementa el concepto de que los modelos, las vistas y controladores deben mantenerse algo separados por almacenamiento de





**Figura 2.3.** Implementación en Rails del Patrón MVC

código para cada uno de los elementos como archivos separados, en directorios separados[23]. Las clases que forman la funcionalidad básica de Rails reside dentro de los módulos: ActiveRecord, Action View y ActionController. Estos dos últimos forman parte de la librería ActionPack. Cada módulo es resumido en la tabla 2.1

Fase MVC	Módulo Rails	Propósito
Modelo	ActiveRecord	Proporciona una interfaz y vinculación entre las tablas de una base de datos relacional y el código de programación Ruby que manipula los registros de base de datos. Los nombres de los métodos de Ruby se generan automáticamente de los nombres de campo de las tablas de bases de datos, y así sucesivamente.
Vista	ActionView	Un Ruby embebido (ERB) sistema basado para la definición y presentación de plantillas para exponer los datos. Cada conexión Web a una aplicación Rails arroja como resultado el despliegue de una vista.
Controlador	ActionController	Un direccionador de datos entre ActiveRecord (interfaz de la base de datos) y ActionView (máquina de presentación). Provee facilidad para manipular y organizar datos desde la base de datos o de entrada en formularios Web, los cuales se liberan a ActionView para inserción y despliegue de la plantilla.

**Tabla 2.1.** Visión general de cómo Rails implementa el framework de diseño MVC

### 2.4.7. ActiveRecord

Antes de comenzar con la definición de Active Record es importante introducir un concepto relacionado con esto. Se trata de Object Relational Mapping (ORM) es cual es una herramienta que se usa para el mapeo de tablas de una base de datos relacional a las clases orientadas a objetos. Esta función es la que cumple la librería Active Record que proporciona la capa modelo de las aplicaciones Rails.

Active Record es la capa ORM suministrada con Rails. Sigue de cerca el modelo estándar de ORM: las tablas mapean a las clases, las filas a los objetos, y las columnas a los atributos de los objetos. Se diferencia de la mayoría de las otras librerías ORM en cuanto a la manera en que es configurado. Mediante el uso de un conjunto razonable de valores predeterminados, Active Record minimiza la cantidad de configuración que los desarrolladores realizan. En la figura 2.4 se muestra como es el mapeo entre la base de datos y la estructura OO[33].

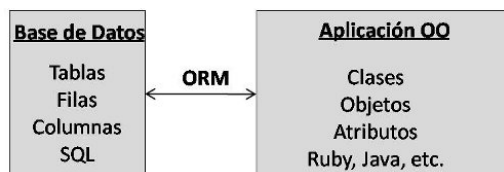


Figura 2.4. Mapeo relacional de objetos (ORM)

Hay muchas implementaciones ORM disponibles en diferentes lenguajes. Lo que hace a Active Record especial es el fácil uso y poder que tiene con pocas líneas de código. A diferencia de la mayoría de las otras implementaciones ORM, no se tiene que escribir líneas sobre líneas de código de configuración para usar esta librería. De hecho, se puede trabajar sin configuración en absoluto, si se siguen las recomendaciones de los esquemas de base de datos y clases.

## Clases y Objetos

Active Record está implementado en Rails como un conjunto de clases bases desde las cuales el modelo de objetos es extendido[14]. Cada table en la base de datos está generalmente representada por una una clase que extiende de una clase base Active Record. De esta manera, el modelo de objetos hereda una gran cantidad de funcionalidad. El modelo de objetos puede ser tan simple como:

```
class Book < ActiveRecord::Base
end
```

### 2.4.8. Action Controller

Es el componente que maneja las solicitudes del navegador y facilita la comunicación entre el modelo y la vista. Los controladores van a heredar de esta clase la cual forma parte de la librería ActionPack [23]. El controlador realiza las siguientes tareas:

- Decide como manejar una solicitud particular, es decir, si reenviar una página entera o sólo un parte de ella.
- Recuperar datos desde el modelo para pasar a la vista.
- Recuperar información de una solicitud desde el navegador para crear o actualizar datos en el modelo.

### 2.4.9. Action View

Es el componente que se encarga de la presentación de las páginas a los usuarios. También forma parte de la librería ActionPack.

Una Vista puede no contener código Ruby y ser sólo un archivo con HTML simple. Sin embargo, es más probable que las vistas contengan una combinación de HTML

y Ruby, haciendo que la página sea más dinámica. El código Ruby es incrustado en HTML utilizando la sintaxis de Ruby Embebido (ERb), siendo similar a la sintaxis de JSP y PHP al envolver el código en etiquetas especiales en el archivo HTML.

## 2.5. HTML Dinámico

El avance de la tecnología ha venido de la mano con la evolución de la Web. Cada día es mayor la exigencia por parte de los usuarios en cuanto a la interacción con una aplicación web. También es bien conocido que hay muchos lenguajes de programación Web que han sido de gran importancia a la hora de construir una aplicación tales como Hyper Text Markup Language (HTML) y Javascript por sólo nombrar algunos. Por todo esto se originó un término que engloba una serie de tecnologías Web, el cual tiene una fuerte influencia en éste ámbito.

### 2.5.1. Definición

El HTML dinámico ó DHTML como es mejor conocido, no significa en términos precisos un lenguaje de programación Web. Su origen se debe a la limitación que tenía HTML, por ejemplo, de crear efectos y animaciones para un aplicación. Se puede decir que DHTML es el arte de hacer páginas web dinámicas mediante la combinación de varias tecnologías que trabajan en conjunto para lograr este objetivo.

*Más que todo, el HTML Dinámico es un ambiente que ofrece control sobre la presentación de la información de una página Web: control sobre fuentes, colores, y todos los aspectos de lo que aparece para un usuario[35]*

### 2.5.2. jQuery

Para lograr interactividad y un ambiente dinámico en la Web se utilizan varias tecnologías que permiten desarrollar estas características. Particularmente, existe una librería en JavaScript que automatiza tareas comunes y simplifica otras complicadas. Esta librería es la muy popular jQuery, la cual proporciona un gran rango de funcionalidades que se pueden agregar a una determinada página web.

#### Funcionalidades Generales

Según [7] JQuery se caracteriza por tener las siguientes funcionalidades:

- **Acceder a partes de una página:** jQuery ofrece un mecanismo de selección robusta y eficiente para localizar partes específicas de la estructura del documento (Document Object Model DOM).
- **Modificar la apariencia de una página:** Con jQuery se tiene la posibilidad de procesar un documento de forma eficaz sin importar que navegador se utilice, algo que no ofrece el procesamiento con CSS. jQuery puede cambiar las clases o las propiedades de estilo individual aplicado a una parte del documento.
- **Alterar el contenido de la página:** Se puede cambiar una parte del documento así como en su totalidad, simplemente con pocas líneas de código, con la facilidad de utilizar el API de jQuery.
- **Añadir animación:** jQuery proporciona retroalimentación y comportamiento interactivo con el usuario.
- **Recuperar información del servidor sin actualizar la página:** Es lo que se conoce como Asynchronous JavaScript and XML (AJAX). La librería jQuery elimina la complejidad del navegador específica de este proceso, permitiendo centrarse sólo en la funcionalidad final del servidor.

- **Simplificar las tareas comunes de JavaScript:** ofrece mejoras de construcciones básicas JavaScript, tales como la iteración y la manipulación de array.

## Selectores

jQuery aprovecha las bondades del CSS y los selectores XPath para el acceso rápido y fácil a los elementos o grupos de elementos del Document Object Model(DOM). También ofrece selectores personalizados jQuery.

## Eventos

JavaScript integra varias maneras de reaccionar a la interacción del usuario y otros eventos. jQuery mejora y extiende los mecanismos de manejo de eventos básicos para darles una sintáxis más elengante y así hacerlos más poderosos.

## Efectos

Con jQuery , se puede agregar impacto fácilmente a las acciones a través de simples efectos visuales, también construir propias animaciones más sofisticadas. Esto proporciona importantes mejoras a la usabilidad que ayudan a orientar al usuario cuando hay algún cambio en la página, como las aplicaciones AJAX.

## 2.6. Virtualización

Las tecnologías de virtualización proveen una manera de separar el hardware físico y software mediante la emulación del uso del hardware y software[6].

### 2.6.1. ¿Cómo funciona?

Esencialmente, un software (llamado hypervisor) es cargado en el computador. Este software se torna para cargar archivos que definen una nueva computadora

virtual llamada Virtual Machine (VM).

Ya que una máquina virtual es en realidad un archivo de datos, no una computadora física, puede ser copiada, movida a otra computadora, etc, simplemente como otro archivo. Típicamente, las computadoras virtuales usan dos estructuras de archivo, una define el hardware, la otra el disco duro[6].

### **2.6.2. Ventajas**

La virtualización puede aportar las ventajas siguientes[30]:

#### **Consolidación para reducir costes de hardware**

- Puede utilizar la virtualización para acceder y gestionar de forma eficaz los recursos para reducir las operaciones y los costes de gestión de los sistemas a la vez que se mantiene la capacidad necesaria.
- Puede utilizar la virtualización para que un nico servidor funcione como si se tratara de varios servidores virtuales.

#### **Optimización de las cargas de trabajo**

- Puede utilizar la virtualización para responder dinámicamente a las necesidades de aplicación de los usuarios.
- Puede utilizar la virtualización para optimizar la utilización de los recursos existentes, pues permite el compartimiento dinámico de las agrupaciones de recursos.

#### **Flexibilidad y capacidad de respuesta de la tecnología de la información**

- Puede utilizar la virtualización para tener una única vista consolidada de todos los recursos de la red que están disponibles y obtener fácil acceso a éstos, con independencia de la ubicación.
- Puede utilizar la virtualización para reducir la gestión de su entorno, al ofrecerse emulación de la compatibilidad y una interoperatividad mejorada.

### 2.6.3. Tipos de Virtualización[37]

No existe una sola manera de realizar la virtualización. De hecho, existen diversas técnicas que alcanzan el mismo resultado a través de diferentes niveles de abstracción. Esta sección presenta tres de las técnicas de virtualización más comunes en Linux, identificando sus puntos fuertes y sus debilidades.

#### Emulación del Hardware

La virtualización más compleja consiste en la emulación de hardware. Con esta técnica, en el sistema anfitrión se utiliza una máquina virtual que emula el hardware, como se muestra en la figura 2.5.

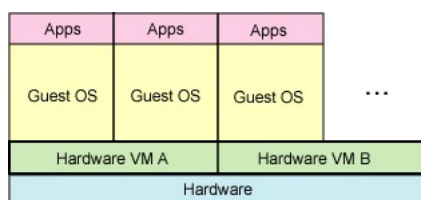


Figura 2.5. Emulación del Hardware.

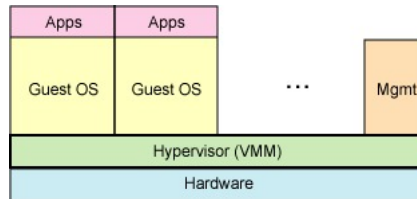
#### Virtualización Completa

La virtualización completa, también llamada virtualización nativa, es otra interesante técnica de virtualización. Este modelo utiliza una máquina virtual que media entre el sistema operativo invitado y el hardware nativo (ver figura 2.6). Algunas instrucciones protegidas deben capturarse y manejarse dentro del hipervisor ya que el hardware subyacente no es propiedad de un sistema operativo sino que es compartido a través del hipervisor.

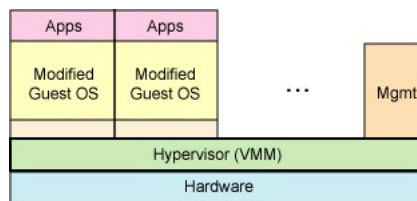
#### Paravirtualización

La paravirtualización es otra técnica popular que cuenta con algunas similitudes con la virtualización completa. Este método utiliza un hipervisor para





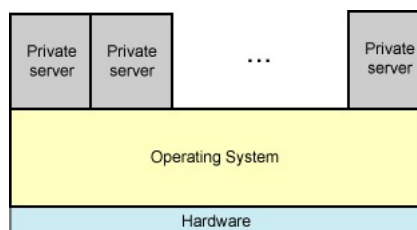
**Figura 2.6.** La virtualización completa utiliza un hipervisor para compartir el hardware subyacente. Comparte el acceso al hardware subyacente pero integra código que está al tanto de la virtualización en el propio sistema operativo (ver Figura 2.7). Esta aproximación evita la necesidad de recompilar y capturar ya que los propios sistemas operativos cooperan en el proceso de virtualización.



**Figura 2.7.** La paravirtualización comparte el proceso con el SO alojado (Guest OS)

## Virtualización en el nivel del sistema operativo

La virtualización en el nivel del sistema operativo, utiliza una técnica diferente. Esta técnica virtualiza los servidores encima del propio sistema operativo. Este método soporta un solo sistema operativo y simplemente aísla los servidores independientes (ver Figura 2.8).



**Figura 2.8.** La virtualización en el nivel del sistema operativo aísla a los servidores

La virtualización en el nivel del sistema operativo requiere cambios en el núcleo del sistema operativo, la ventaja es un rendimiento igual a la ejecución nativa.

## 2.7. Computación en la Nube

Cloud Computing[1] se refiere por una parte a la aplicaciones entregadas como servicios sobre internet y por otra parte al hardware en Datacenters que proveen estos servicios. También en [2] se menciona que Cloud Computing es la convergencia y evolución de muchos conceptos de virtualización, de aplicaciones distribuidas, de Grids que habilitan un enfoque flexible para el despliegue y del escalado de aplicaciones[20]. Cloud Computing es un término utilizado para describir una plataforma y el tipo de aplicación [5], siendo una plataforma que dinámicamente aprovisiona, configura y reconfigura servidores como sea necesario.

### 2.7.1. Evolución de Cloud Computing

El término Cloud Computing no es nuevo, ni revolucionario, sino que es un término que ha venido evolucionando con el tiempo[32]. Empezó en los años 1980's bajo conceptos de Grid Computing, aunque con ciertas diferencias y enfatizado a servidores virtuales; luego en los años 1990's se expandió el concepto de virtualización elevando el nivel de abstracción de los servidores virtuales, primero como plataforma virtual y luego como aplicaciones virtuales; más adelante se conoció el término Utility Computing, que ofrece clusters como plataformas virtuales; recientemente el término software como servicio (SaaS) elevando el nivel de virtualización a las aplicaciones, con un modelo de negocio no recargado en recursos consumidos.

Debido a esta evolución el concepto de Cloud Computing combina los términos anteriores de Grid, Utility y SaaS, siendo un modelo emergente en donde los

usuarios pueden tener acceso a las aplicaciones desde cualquier lugar a través de dispositivos conectados.

### 2.7.2. Beneficios y riesgos

En [36] se mencionan los siguientes beneficios de Cloud Computing:

Reduce el costo total de propiedad: debido a que en toda organización ese es uno de los principales objetivos, minimización de costos en hardware y software. Incrementa la escalabilidad y fiabilidad: debido que a través de diferentes infraestructuras se puede habilitar lo escalable y fiable que pueden llegar a ser los sistemas bajo conceptos de Cloud Computing. Además, existen varios beneficios que también puede actuar como riesgos dependiendo si la implementación se realizó bajo los pasos sugeridos, entre estos se pueden mencionar la disponibilidad, el rendimiento, la capacidad, etc.

Por otra parte en [34] se mencionan beneficios referentes al desarrollo y despliegue:

- Reduce el tiempo de ejecución y el tiempo de respuesta.
- Minimiza el riesgo de adquirir nueva infraestructura.
- Menor costo de inversión inicial.
- Aumenta el ritmo de la innovación.

### 2.7.3. Implementación

En [22] y [8] se mencionan tres escenarios para la formación de Cloud Computing, a saber, Private Cloud, Public Cloud e Hybrid Cloud. Estos escenarios se han convertido en medios atractivos para el intercambio computacional, de almacenamiento y de recursos de red entre desarrolladores de servicios múltiples y de aplicaciones de prestación de servicios. No hay que olvidar la capacidad de reasignar dinámicamente los recursos utilizando tecnologías de virtualización, ayudando

a mitigar la necesidad de inversiones adicionales en infraestructura en tiempos de alta demanda.

Estos tres escenarios son implementados en los Datacenters y su disposición hace que su categorización dependa del enfoque, ya sea interno, externo o combinado.

### **Private Cloud**

Private Clouds [22] son escenarios donde las compañías realizan sus operaciones fuera de línea, ejecutando aplicaciones seguras en Datacenters. De modo que, este escenario también se puede llamar Internal Cloud.

En [8] se menciona que Internal Cloud aplica los conceptos de Cloud Computing a recursos propios de la empresa que consume el servicio, proveyendo la capacidad de manejar aplicaciones web nuevas y existentes, mientras se provee de seguridad y regulación. También se menciona que Private Cloud trae consigo ciertas ventajas:

- Disponible en demanda.
- Rápido aprovisionamiento de servicios de negocio.
- Reducción del costo a través de economías a escala.
- Flexibilidad y libertad de selección.
- Basado en el uso.
- Controlado y asegurado por corporación de IT.

### **Public Cloud**

Public Clouds [22] son escenarios donde las compañías necesitan mover datos o aplicaciones desde su interior al exterior, ambos escenarios utilizan la misma

arquitectura, con la diferencia que el escenario público se conecta con otros escenarios. De modo que, este escenario también se puede llamar External Cloud.

External Cloud involucra recursos y servicios IT que son vendidos [8], tales como auto servicio, aprovisionamiento en demanda y pago por utilización, todos estos servicios accedidos a través de navegadores web o a través de API's.

Bajo el escenario de External Cloud se ejecutan diferentes tipos servicios que se detallarán en la siguiente sección, que son software, plataformas e infraestructura como servicios.

### **Hybrid Cloud**

Por último, se hace referencia a una mezcla entre los dos escenarios anteriores, llamada Hybrid Cloud, en [22] se le llama Inter Cloud, que es un tipo de escenario semi público, el cual se comporta como un Private Cloud con la particularidad que ciertas empresas pueden compartir su información con ciertos niveles de permiso, por ello el término semipúblico.

El control de Public Cloud lo hace el proveedor, mientras que el control de Private Cloud lo hace la empresa, y la finalidad es que a través de ambos enfoques satisfacer las necesidades de un sistema de aplicación.

Inter Cloud como escenario [8], brindaría la capacidad de elegir los proveedores de servicio, y los proveedores de servicio federados serían capaces de compartir las cargas de servicio, siendo una relación más flexible.

#### 2.7.4. Niveles de servicio

Los diferentes niveles de servicio que componen a Cloud Computing son IaaS, PaaS y SaaS. Sin embargo, en [9] se menciona otro nivel de servicio, aunque ciertos autores lo prefieren alojar entre los mismos tres niveles de servicio y no crear uno nuevo, es el llamado ITaaS, IT como servicio, siendo un modelo de servicio donde una organización o individuo contrata con un proveedor de servicios para obtener conectividad de red y cualquier otro servicio incluido, como backup de red, recuperación de desastres, VPN, conferencias web, etc. Aunque este nivel es muy general y puede abarcar a los tres anteriores pero de una forma unificada.

#### IaaS

Infrastructure as a Service o infraestructura como servicio [2] es entregar tanto hardware como software como un servicio. El ejemplo más común es el hosting, el cual, nos provee de hardware como un servidor y de software como un webserver, sin embargo, este concepto ha evolucionado a infraestructura como EC2 y S3.

Otra manera de ver IaaS [29] es como la manera de compartir recursos para ejecutar servicios, típicamente utilizando tecnología de virtualización, en donde múltiples usuarios utilizan dichos recursos. Los recursos pueden fácilmente escalar cuando la demanda se incrementa, y generalmente se utiliza métodos como pago por uso.

Uno de los servicios que toma importancia concerniente a la infraestructura es el Cloud Storage que se menciona a continuación:

## Cloud Storage

Es almacenamiento localizado ya sea en Datacenters públicos o privados, separados del almacenamiento primario. Pertenecce al nivel de infraestructura como servicio, la manera de implementación es a través [21] del Service Oriented Architecture (SOA), y la localización tiene diferentes variaciones, Cloud Storage puede ubicarse en: Un Datacenter público, Un Datacenter privado, o Separado del almacenamiento primario.

La manera de acceso puede ser de dos formas: Directamente como bloques o archivos; o indirectamente a través de aplicaciones que están ubicadas en el mismo lugar del almacenamiento. Hay dos tecnologías [27] que proporcionan métodos de almacenamiento: La primera es Storage Area Network (SAN) que son switches de redes de alta velocidad que permiten que múltiples computadoras tengan acceso compartido a varios dispositivos de almacenamiento; y la segunda es Network-Attached Storage (NAS) que vienen como aplicaciones NAS o Gateways NAS, son servidores de archivos virtuales que tienen soporte a protocolos como NFS, siendo un dispositivo que directamente concede a la red y que tiene capacidades de compartir archivos.

Los protocolos utilizados para Cloud Storage son SOAP o REST. También existen ciertas barreras para su adopción, tales como, el ahorro de costos no es significativo, se pone en riesgo la privacidad de los datos, cuestiones de migración, disponibilidad de datos y contratos como SLAs.

## PaaS

Platform as a Service o plataforma como servicio [2], es entregar una plataforma de desarrollo de aplicaciones como un servicio para desarrolladores en la web.

Generalmente se provee de herramientas tipo middleware, por ejemplo, Google AppEngine. Además de dicha entrega, también se ofrece un ambiente de ejecución como el servidor de aplicaciones.

## SaaS

Software as a Service o software como servicio [2] provee la administración y hosting de aplicaciones con sus propios Datacenters, se maneja el término de múltiples inquilinos, por ejemplo Oracle CRM On Demand o Salesforce.

Es interesante lo que dice [10] acerca de SaaS, lo compara con la controversia que generó la computación cliente-servidor para reemplazar la computación de mainframes, y para el usuario final SaaS es un simple concepto, el usuario solamente ingresa a una aplicación a través del navegador web sin saber en donde se aloja o como está siendo servida, solamente sabe que es rápida y segura.



## Marco Metodológico

En este capítulo se describe la metodología que se utilizó para el desarrollo del sistema. Ésta se basó en el método ágil de programación extrema (Extreme programming XP) que no se enfoca específicamente en la gran cantidad de artefactos que se deben producir (modelo y documentación detallada), como lo establecen las metodologías tradicionales, sino que está más centrado en el factor humano, el cliente y el desarrollo incremental del software. También, en esta parte se presenta la adaptación de este proceso para el desarrollo de las actividades y los formatos utilizados para el manejo de los requerimientos.

### **3.1. Programación Extrema(XP)**

Es un método ágil centrado en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico[24].

Las principales características de este método son las siguientes[16]:

- **Comunicación:** Los programadores están en constante comunicación con los clientes para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos.
- **Simplicidad:** Codificación y diseños simples y claros.
- **Realimentación (Feedback):** Mediante la realimentación se ofrece al cliente la posibilidad de conseguir un sistema apto a sus necesidades ya que se le va mostrando el proyecto a tiempo para poder ser cambiado y poder retroceder a una fase anterior para rediseñarlo a su gusto.
- **Coraje:** Se debe tener coraje o valentía para cumplir los tres puntos anteriores; Hay que tener valor para comunicarse con el cliente y enfatizar algunos puntos, a pesar de que esto pueda dar sensación de ignorancia por parte del programador, hay que tener coraje para mantener un diseño simple y no optar por el camino más fácil y por último hay que confiar en que la realimentación será efectiva

El objetivo principal que se persigue es la satisfacción del cliente, por eso tiene mucha importancia la comunicación con los usuarios o clientes. Esta comunicación se va a soportar principalmente en las historias de usuario (del término en inglés User Stories) cuando proviene desde el cliente, y de las entregas y versiones parciales del sistema cuando la comunicación es hacia el cliente[26].

### 3.2. Adaptación del Proceso de Desarrollo XP

En esta parte se describe como se adecuará el proceso XP al desarrollo en general de las actividades (ver figura 3.1). Se presentan las Historias de Usuario y la forma en que se manejarán las cuatro (4) etapas del proceso y los formatos a seguir.

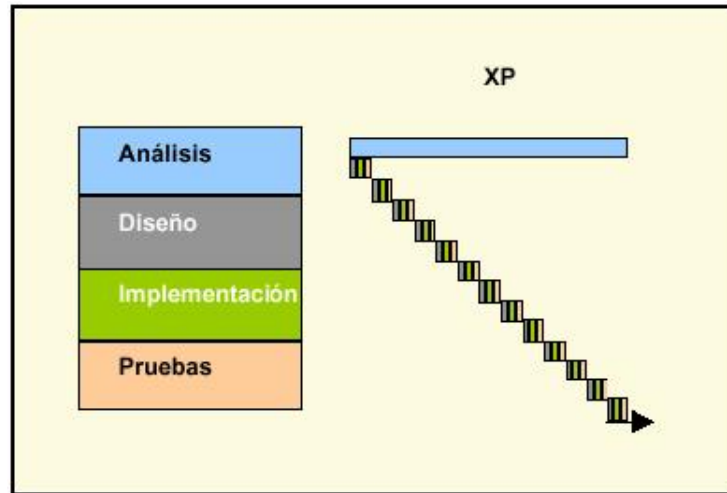


Figura 3.1. Desarrollo en XP

### 3.2.1. Historias de Usuario

Las Historias de Usuario se utilizan para especificar los requisitos del software, sean funcionales o no. El manejo es dinámico ya que pueden eliminarse, reemplazarse, modificarse o crearse nuevas historias. Para la información contenida en la plantilla de las Historia de Usuario se utilizará: un número de identificación, el nombre, la descripción, tipo (nueva, mejora, modificación), prioridad (alta, media o baja) y tiempo estimado de desarrollo expresado en días. El formato a utilizar se puede observar en la tabla 3.1.

Número: -	Nombre: -
Prioridad: -	Estimación: - Tipo: -
Descripción: -	

Tabla 3.1. Formato de Registro para Historias de Usuario

### 3.2.2. Fases de desarrollo

El método XP está compuesto por cuatro actividades fundamentales las cuales están contenidas en cada una de las iteraciones del proceso de desarrollo. A

continuación una breve descripción y cómo será la adaptación de cada una de ellas para la implementación del sistema.

## Planificación

Mediante las historias de usuario se llevará a cabo la planificación del proyecto. Allí se describen las funcionalidades, actividades y eventos que se hicieron necesarias para el cliente, a la vez que indican la cantidad o estimación del tiempo requerido para el desarrollo. Cada iteración será implementada de acuerdo a un conjunto de Historias de Usuarios y con un duración estimada de 2 a 3 semanas. Al inicio de cada iteración se utilizará un esquema (ver tabla 3.2) con el número de la iteración, una descripción, el número y nombre de las historias de usuarios a desarrollar, la fecha de inicio y la fecha de fin de la iteración.

<b>Iteración: -</b>	
<b>Descripción:</b>	
<b>F.Inicio/F.Fin</b> -	
<b>Número</b>	<b>Historia</b>
-	-

**Tabla 3.2.** Formato de planificación de cada iteración

## Diseño

El diseño en XP sigue de forma rigurosa el principio de simplicidad, prefiriendo siempre un diseño simple respecto de una presentación más compleja. Además el diseño debe ofrecer una guía de implementación para una historia de usuario determinada[13]. De acuerdo a la iteración se presentará en esta fase interfaces de usuario, modelos de datos, diagramas que permitan comprender de una mejor manera el funcionamiento del sistema.

## Codificación

La programación se realizará de forma individual a pesar de que el método sugiere que sea de dos personas. Se llevará a cabo la integración de código continua con el otro grupo de desarrollo a través del sistema de control de versiones (subversión), esto con la finalidad de evitar problemas con la integración de los componentes. Además, se hará la configuración necesaria y preparación del ambiente adecuado para el sistema. También, se mostrará código de algunas partes del sistema referente a las historias de usuario en determinada iteración.

## Pruebas

XP establece que se deben codificar y automatizar las pruebas unitarias creadas en la fase de diseño, para cada historia de usuario; así como realizar pruebas de aceptación por cada iteración terminada, las cuales son especificadas por el cliente, enfocándose en las funcionalidades del sistema que son manejadas por él [31]. En este caso, el usuario o cliente prueba el sistema y verifica si se cumple con los requerimientos. También se realizaron pruebas unitarias y por cada funcionalidad por parte del programador. El formato a utilizar para las pruebas de aceptación se muestra en la tabla 3.3.

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
-	-	-	-	-

**Tabla 3.3.** Formato de Pruebas de Aceptación

Se puede dar el caso de que en cada una de las iteraciones no se determinen las cuatro fases planteadas anteriormente ya que esto va a depender de las tareas que estén descritas en las historias de usuario pertenecientes a una determinada iteración.

### 3.2.3. Análisis General del Sistema

El análisis se realizó comenzando con los requerimientos por parte del usuario, estableciendo las tecnologías a utilizar así como la metáfora o arquitectura general del sistema. Además se determinaron las historias de usuario para el desarrollo de la aplicación y las actividades a realizar. A continuación, se muestra la lista de las historias de usuario antes mencionadas.

<b>Número:</b> 1	<b>Nombre:</b> Instalación de ambiente de desarrollo	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Instalación y configuración de las herramientas a utilizar para el desarrollo de la aplicación.		

<b>Número:</b> 2	<b>Nombre:</b> Realizar diseño del modelo de datos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Diseñar modelo de datos donde se especifican las tablas y atributos para el manejo de la aplicación.		

<b>Número:</b> 3	<b>Nombre:</b> Creación de base de datos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Creación de la base de datos con el manejador MySQL en base al modelo construído.		

<b>Número:</b> 4	<b>Nombre:</b> Desarrollo de métodos del módulo de gestión de grupos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 4 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Implementación de métodos para crear un grupo nuevo o modificarlos con las direcciones requeridas por el usuario.		

<b>Número:</b> 5	<b>Nombre:</b> Desarrollo de vistas correspondientes al módulo de grupos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 6 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Creación de vistas que muestre el formulario para crear nuevos grupos y y/o modificarlos.		

<b>Número:</b> 6	<b>Nombre:</b> Desarrollo de componente ajax para listar grupos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Decripción:</b> Implementación de un componente con tecnología ajax que permita listar los grupos creados.		

<b>Número:</b> 7	<b>Nombre:</b> Desarrollo de componente para agregar direcciones web a un grupo	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de un componente javascript que permite agregar nuevas direcciones a un grupo.		

<b>Número:</b> 8	<b>Nombre:</b> Desarrollar vistas para desplegar direcciones web	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de vistas que permitan listar las direcciones web de un grupo.		

<b>Número:</b> 9	<b>Nombre:</b> Desarrollar componente ajax para agregar dirección a grupo	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de un formulario para agregar una dirección a un grupo.		

<b>Número:</b> 10	<b>Nombre:</b> Desarrollar método para listar direcciones de un proceso	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar componente ajax para listar direcciones web contenidas en un grupo.		

<b>Número:</b> 11	<b>Nombre:</b> Desarrollar métodos para la creación y modificación de módulos	
<b>Prioridad:</b> Medio	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar métodos que permitan la gestión de módulos de la aplicación.		

<b>Número:</b> 12	<b>Nombre:</b> Desarrollar vistas de gestión de módulos	
<b>Prioridad:</b> Medio	<b>Estimación:</b> 1 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar vistas que permitan la gestión de módulos que componen la aplicación.		

<b>Número:</b> 13	<b>Nombre:</b> Desarrollar métodos para crear y modificar una configuración	
<b>Prioridad:</b> Medio	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar métodos que permitan la gestión de la configuración de información pertinente a la aplicación.		

<b>Número:</b> 14	<b>Nombre:</b> Desarrollar vistas de gestión de configuración	
<b>Prioridad:</b> Medio	<b>Estimación:</b> 1 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar vistas que permitan la gestión de la configuración de información pertinente a la aplicación.		

<b>Número:</b> 15	<b>Nombre:</b> Desarrollar menú principal de la aplicación	
<b>Prioridad:</b> Medio	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar vistas y métodos que permitan mostrar el menú principal de la aplicación.		

<b>Número:</b> 16	<b>Nombre:</b> Desarrollar métodos de gestión de usuarios	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar métodos que permitan la gestión de los usuarios de la aplicación.		

<b>Número:</b> 17	<b>Nombre:</b> Desarrollar vistas de gestión de usuarios	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de vistas que permitan el manejo de los usuarios de la aplicación.		

<b>Número:</b> 18	<b>Nombre:</b> Desarrollar método de autenticación	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar método que permitan la autenticación de usuarios registrados en la aplicación.		

<b>Número:</b> 19	<b>Nombre:</b> Desarrollar modelos de envío de correo	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollar modelos que permitan el envío de correos a usuarios registrados.		

<b>Número:</b> 20	<b>Nombre:</b> Desarrollar vistas para envío de correos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 1 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de vistas para el envío de correos a usuarios registrados.		

<b>Número:</b> 21	<b>Nombre:</b> Configuración de plugin Backgroundrb	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 4 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Instalación y configuración de plugin Backgroundrb para tareas de procesamiento en segundo plano		

<b>Número:</b> 22	<b>Nombre:</b> Creación de workers	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de workers que funcionen para colocar determinadas tareas en procesamiento de segundo plano.		

<b>Número:</b> 23	<b>Nombre:</b> Creación de procesador de imágenes	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 3 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollo de modelo para crear preview de las direcciones web		

<b>Número:</b> 24	<b>Nombre:</b> Instalación de la herramienta wkhtmltoimage	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Instalación y uso de la herramienta wkhtmltoimage para preview de la direcciones.		



<b>Número:</b> 25	<b>Nombre:</b> Desarrollar método para creación de thumbnail	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de método que permita, luego de crearse una imagen, la creación de un thumbnail de la misma.		

### 3.2.4. Metáfora del Sistema

Todo el sistema fue implementado con la finalidad de permitir a un usuario tener la posibilidad de seleccionar un conjunto de direcciones Web y obtenerlas mediante un Panel de Control el cual va a proveer de funcionalidades acordes con la solicitud que desee realizar el usuario. Mediante el Controlador Central, estas direcciones van a ser enviadas a un Spider que se va a encargar de rastrearlas mediante el internet para que luego sean guardadas en un Archivo y así permitir accederlas de manera centralizada (ver figura 3.2).

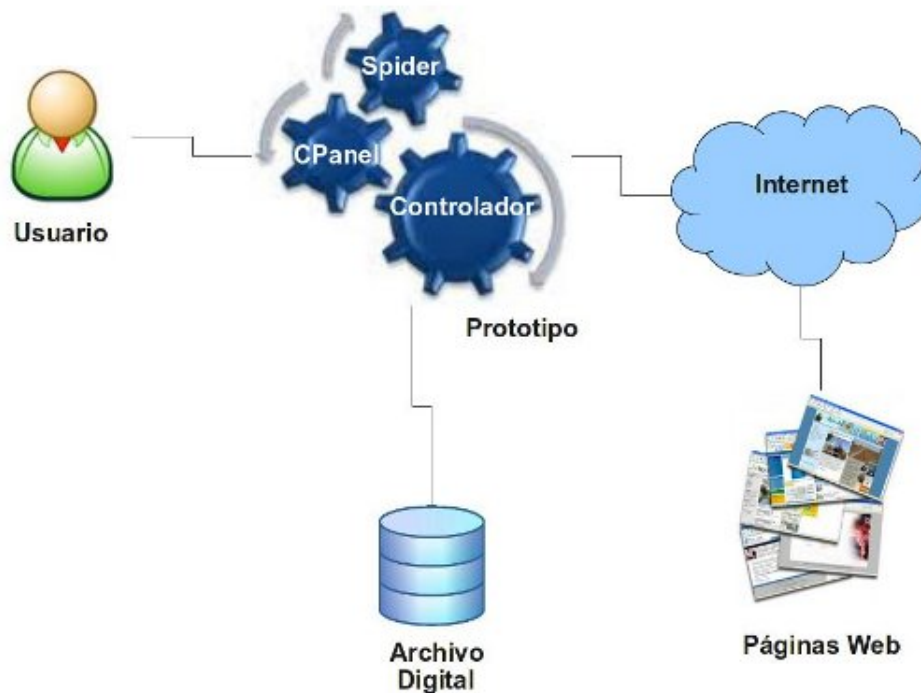


Figura 3.2. Metáfora del Sistema

## Marco Aplicativo

En esta parte se presenta todas las actividades realizadas siguiendo lo establecido por el proceso de desarrollo eXtreme Programming. Estas actividades son determinadas por una serie de iteraciones que contemplan las cuatro (4) fases de XP mencionadas en el capítulo anterior.

### 4.1. Plan de Iteración

Cada iteración estará compuesta por un conjunto de historias de usuarios de acuerdo a los requerimientos establecidos que determinarán una parte funcional del sistema. Estas tendrán una duración aproximada de 2 a 3 semanas. Se contemplarán casos de prueba para verificar el cumplimiento de los requerimientos. Todo el proceso de desarrollo junto con la documentación se realizó en cinco (5) meses aproximadamente comenzando desde el 15 de diciembre hasta el 15 de mayo de 2011.

#### 4.1.1. Iteración 0

Se comienza con la instalación de las aplicaciones para el ambiente de desarrollo. Se realiza una estructura de la base de datos y se define el modelo de datos inicial. Este va a permitir almacenar la información requerida por el usuario de la aplicación.

## Planificación

Iteración 0	
<b>Descripción:</b>	Instalación de ambiente de desarrollo, modelo de base de datos e instalación de base de datos en manejador MySQL.
<b>F.Inicio/F.Fin</b>	15-12 al 28-12-2010
<b>Número</b>	<b>Historia</b>
1	Instalación del ambiente de desarrollo.
2	Diseñar modelo de datos donde se especifican las tablas y atributos para el manejo de la aplicación.
3	Creación de la base de datos con el manejador MySQL en base al modelo construido.

## Diseño

En esta etapa se procede al diseño del modelo de datos, el resultado se muestra en la figura 4.1. Las tablas “archivos” y “carpetas” se encargarían de almacenar información referente a un administrador de carpetas que se iba a contemplar en este paso inicial del proyecto. Se tiene también la tabla de urls y procesos que se van a encargar de almacenar información acerca de los grupos que son los que van a contener una colección de direcciones web válidas provenientes de la Web. Este pool de direcciones deben pertenecer a categorías similares, quedará de parte del usuario como crear la mencionada colección que será identificada con el nombre del grupo. La tabla usuario permitirá almacenar la información referente a los usuarios que pueden utilizar la aplicación. Las tablas restantes se encargarán de almacenar información referente al manejo del contenido que se presenta en las funcionalidades de la interfaz.

Además, se realizó la instalación del ambiente general de desarrollo. De acuerdo a las tecnologías a utilizar mencionadas en el capítulo 2, se configuró lo siguiente: ruby 1.8.7 como lenguaje de programación, rails 2.3.8 como framework y MySQL 5.1 como manejador de la base de datos.

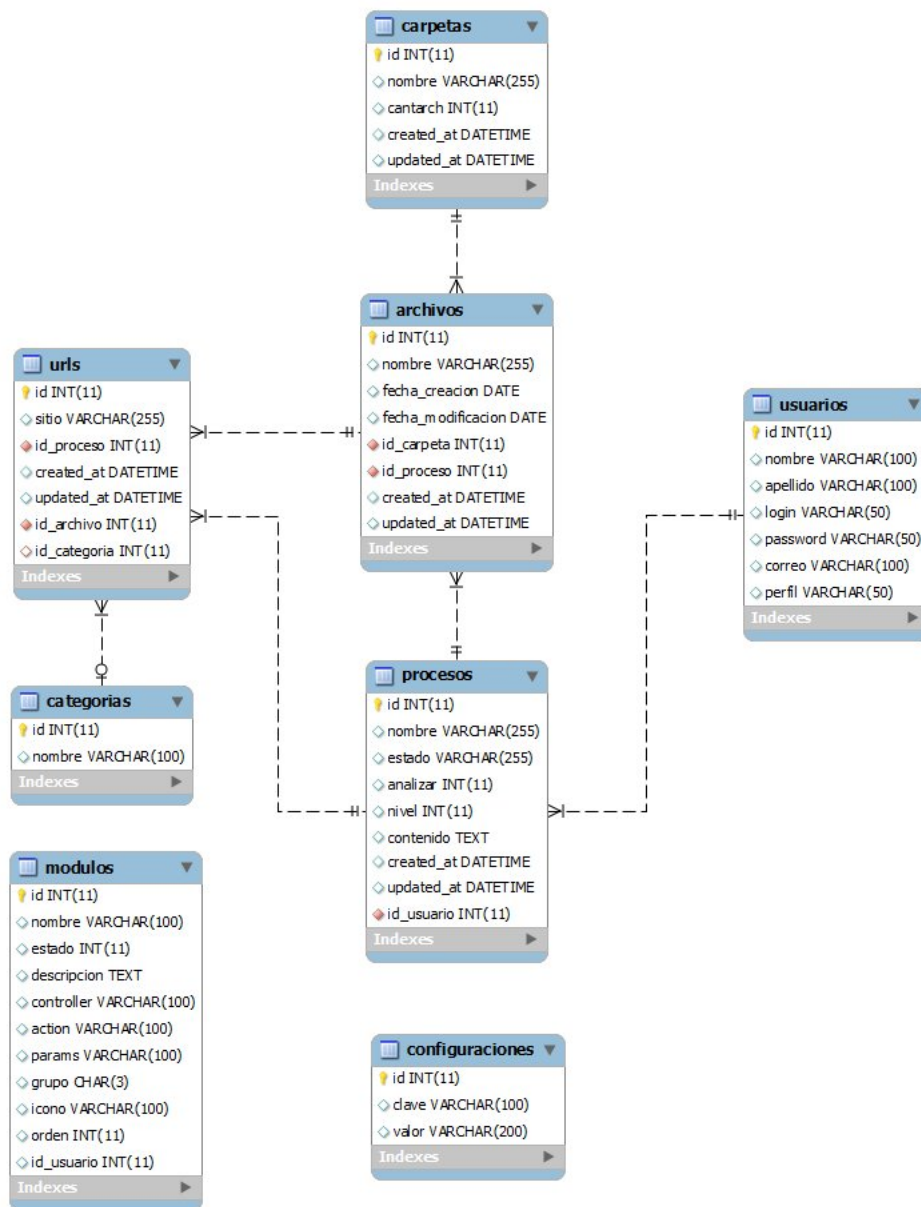


Figura 4.1. Modelo de datos inicial

## **Pruebas**

Se hicieron pruebas para verificar que la instalación de los programas fue realizada de forma correcta mediante la ejecución de ejemplos sencillos y la creación de un proyecto de prueba.

Se realizaron pruebas de conexión de base de datos con la aplicación mediante la creación de los modelos correspondientes con cada tabla de la base de datos y la configuración del archivo con los datos de la base de datos a utilizar. Se verificó a través de la consola de ruby (ruby script/console) que se obtuviesen resultados positivos de consultas sencillas con ActiveRecord.

### 4.1.2. Iteración 1

Después de la instalación del ambiente para el desarrollo, se comenzó con la implementación de las funcionalidades requeridas. Es esta iteración se desarrollaron las funcionalidades que tienen relación con el módulo de grupos que contendrán las colecciones de direcciones web.

#### Planificación

Iteración 1	
<b>Descripción:</b>	Desarrollo de métodos, modelos y vistas del módulo de grupos
<b>F.Inicio/F.Fin</b>	29-12 al 12-01-2011
<b>Número</b>	<b>Historia</b>
4	Desarrollo de método para crear, modificar y eliminar grupos de direcciones.
5	Desarrollo de vistas correspondientes al módulo de grupos.
10	Desarrollar método para listar direcciones web pertenecientes a un grupo.

#### Diseño

El desarrollo de las interfaces asociadas al módulo de grupos se encargarán de mostrar al usuario las diferentes actividades que se pueden realizar referente a los grupos que contienen las direcciones web. El diseño fue realizado de manera sencilla y consta de formularios que contienen los datos necesarios, en este caso para crear o modificar un grupo. También, se tiene la opción de agregar las direcciones asociadas al grupo de manera dinámica. En la figura 4.2 se refleja el formulario que aparece si se desea crear un nuevo grupo.

#### Codificación

Esencialmente se crearon los métodos que se encargan de procesar la información suministrada por el usuario en los formularios y de mostrar las vistas con los

## NUEVO GRUPO

---

**Volver**

Nombre

**Figura 4.2.** Formulario para crear nuevo grupo.

datos correctos. Al momento de crear un grupo nuevo se verifica que este no se encuentre en la base de datos para evitar inconsistencias. En la figura 4.3 se puede observar como es el proceso de creación de un grupo. También, se cuenta con el método cuando el usuario desea modificar el grupo de direcciones, así como en el caso de eliminar un grupo. Se implementaron mensajes necesarios para proveer feedback en cuanto a posibles errores o en caso de éxito de la solicitud realizada.

```

def create
  @proceso = Proceso.new(params[:proceso])

  id_ultimo_proc = Proceso.last
  @urls = Url.find(:all, :conditions => {:id_proceso => id_ultimo_proc.id + 1})

  respond_to do |format|
    if @proceso.save
      flash[:notice] = 'El grupo fue creado con &ecute;xito'
      format.html { redirect_to(@proceso) }
      format.xml { render :xml => @proceso, :status => :created, :location => @proceso }
    else
      format.html { render :action => "new" }
      format.xml { render :xml => @proceso.errors, :status => :unprocessable_entity }
    end
  end

  @urls.each do |d|
    d.id_proceso = @proceso.id
    d.save
  end
end
end

```

**Figura 4.3.** Código de crear nuevo proceso.



## Pruebas

Se llevaron a cabo pruebas de aceptación para verificar que el desarrollo de las funcionalidades en esta iteración cumplieran con lo requerido. Las pruebas se hicieron proporcionando los datos de los formularios, lo que dió resultados positivos. Éstas se muestran a continuación.

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
1	4, 5	Proporcionar los datos en un formulario para la crear un grupo	Se crea un grupo junto con las direcciones requeridas por el usuario	Los datos que se suministraron se guardaron correctamente y se creó o modificó el proceo con éxito
2	4, 5	Proporcionar los datos para la modificación de un grupo	Se realiza la modificación de un grupo y los datos se deben actualizar correctamente en la base de datos	Los datos se modifican correctamente en la base de datos.
3	4, 5	Permitir eliminar un proceso a través de un link	Se elimina correctamente de la base de datos un proceso seleccionado por el usuario	Se elimina correctamente de la base de datos el grupo seleccionado por el usuario

### 4.1.3. Iteración 2

La presente iteración cuenta con el desarrollo de funcionalidades asociadas al módulo de grupos que son el tener la posibilidad de agregar direcciones web a un grupo determinado de manera dinámica.

#### Planificación

Iteración 2	
<b>Descripción:</b>	Desarrollo de componentes para el manejo dinámico de los datos del módulo de grupos
<b>F.Inicio/F.Fin</b>	13-01 al 07-02-2011
<b>Número</b>	<b>Historia</b>
6	Desarrollo de componente ajax para listar grupos.
7	Desarrollo de componente javascript para agregar direcciones web a un grupo.
8	Creación de vistas que permitan listar las direcciones web de un grupo.
9	Desarrollar componente ajax para agregar dirección web a un grupo.

#### Diseño

Se procedió a crear componentes que se desarrollaron aplicando tecnologías Ajax y JavaScript con jQuery. Esto se realizó con el fin de proporcionarle al usuario más funcionalidades asociadas a los grupos y direcciones web de forma dinámica. Específicamente se desarrollo un componente para agregar las direcciones web de los grupos cuando se está creando. En la figura 4.4 se muestra el componente para agregar las direcciones a un grupo.

Entre los métodos que permiten el manejo de las direcciones están el de modificar un grupo y la opción de eliminar direcciones del mencionado grupo.

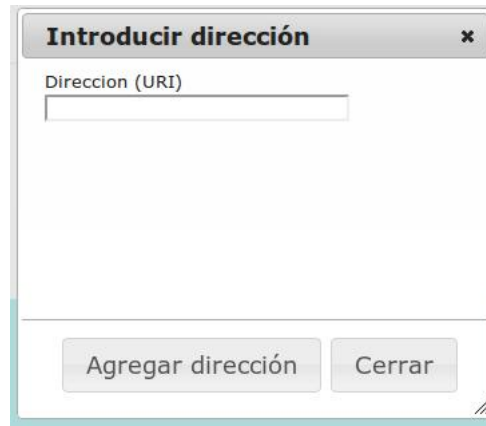


Figura 4.4. Componente javascript para agregar direcciones web a un proceso.

## Codificación

Las direcciones que se van agregando un nuevo grupo o a la modificación de un grupo se verifican de manera tal que no estén mal formadas o que se no se repitan en el mismo grupo. Se desarrolló un formulario que contiene el campo del url de la dirección que se requiere. Al agregar direcciones éstas se van colocando en la interfaz del grupo mediante llamadas ajax, lo mismo ocurre al eliminar un dirección.

```
def add_item
  contenido = params[:direccion].to_s
  id_ultimo_proc = Proceso.last
  Url.create(:sitio => contenido, :id_proceso => id_ultimo_proc.id + 1)
  @dir = Url.find(:all, :conditions => {:id_proceso => id_ultimo_proc.id + 1})
end

def add_item_edit
  proceso_id = params[:id_proc]
  contenido = params[:direccion].to_s
  Url.create(:sitio => contenido, :id_proceso => proceso_id)
  @dir = Url.find(:all, :conditions => {:id_proceso => proceso_id})
  render "add_item"
end

def delete_item
  #encontrar los urls del proceso actual
  proceso_id = params[:id_proc]
  direccion = Url.find(params[:id_url])
  direccion.delete
  @dir = Url.find(:all, :conditions => {:id_proceso => proceso_id})
  render "add_item"
end
```

Figura 4.5. Código para agregar o eliminar mediante ajax las direcciones web.

## Pruebas

Las pruebas realizadas en esta fase principalmente se enfocan en verificar el correcto funcionamiento de los componentes desarrollados mediante la introducción de datos e interacción con las opciones mostradas en la interfaz. Las pruebas de aceptación se muestran a continuación.

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
4	6, 7 y 9	Proporcionar los datos para la agregar direcciones a un grupo	Se agrega una dirección a un grupo de terminado y se guarda en la base de datos correctamente	Se agrega las direcciones y se guardan correctamente en la base de datos
5	6, 7 y 9	Seleccionar mediante un botón agregar una dirección	El sistema debe mostrar una ventana con un formulario para agregar una dirección	El sistema muestra la ventana con los datos correctos
6	6, 7 y 9	Seleccionar mediante un link eliminar una dirección	El sistema debe permitir que se elimine una dirección determinada a través de un link	El sistema elimina la dirección seleccionada por el usuario

#### 4.1.4. Iteración 3

En esta iteración se desarrollaron los módulos de configuración y módulos que componen la aplicación que le permitirá al usuario manejar la información concerniente al sistema como tal.

#### Planificación

<b>Iteración 3</b>	
<b>Descripción:</b>	Desarrollo de métodos, modelos y vistas referentes a la creación y modificación de módulos, configuración y menú principal de la aplicación.
<b>F.Inicio/F.Fin</b>	08-02 al 22-02-2011
<b>Número</b>	<b>Historia</b>
11	Desarrollar métodos para la gestión de módulos de la aplicación.
12	Desarrollar vistas para la gestión de módulos de la aplicación.
13	Desarrollar métodos para la gestión de la configuración.
14	Desarrollar vistas para la gestión de configuración.
15	Desarrollar menú principal de la aplicación.

#### Diseño

Para esta iteración, las interfaces desarrolladas iban a proveer las funcionalidades requeridas para la gestión de la configuración y los módulos que componen la aplicación. Para la creación o modificación de la información asociada a estos módulos se proporcionan formularios.

En cuanto al módulo de configuración, se tiene la posibilidad de manejar información, como por ejemplo, agregar las rutas del directorio donde se puedan encontrar las imágenes, estos datos se identifican mediante una clave que va a contener el nombre de la configuración y un valor (por ejemplo el tamaño de las imágenes sean 20x20).

En el caso de los módulos, estos son los que se presentan en el menú principal de la aplicación. La información que se maneja es la del nombre, el controlador asociado, el orden en que aparece, entre otros.

La vista del menú principal se diseño de manera tal que las funcionalidades se diviesen en tres (3) grupos en general. Cada funcionalidad apunta a un función específica de la aplicación. En la figura 4.9 se observa el menú principal.



**Figura 4.6.** Vista menú principal

## Codificación

En esta fase se desarrollaron los métodos que permitieran la creación, modificación, actualización y eliminación de algún módulo o configuración como tal. También se generaron las interfaces correspondientes que permitieran el manejo de estas funcionalidades mediante formularios donde se muestran los datos necesarios de acuerdo a la información requerida en el módulo.

Para poder agregar un nuevo módulo o ítem, se tiene que tomar en cuenta la imagen que será mostrada en el menú principal. Esta es solicitada y debe ser alojada en el directorio correspondiente a las imágenes, el cual se maneja en el módulo de configuración. Mediante una selección en el formulario el usuario asocia la imagen que quiere ser mostrada.

El menú muestra las funcionalidades divididas de acuerdo a lo que realicen cada una. Principalmente se hace una división de tres grupos. Los grupos se identifican como: Procesamiento, Gestión de Datos y Administración. Estas se muestran de acuerdo a la gestión que se realice en módulo “módulos”, que cuenta con un atributo donde se tiene la opción de permitir que aparezca o no en el menú principal.

### **Pruebas**

Para las pruebas en este caso se corroboró que se pudiesen realizar las tareas esenciales de cada módulo tales como crear, modificar y eliminar un ítem determinado. También, se verificó si los componentes del menú principal estuviesen apuntando a la funcionalidad correspondiente. Las pruebas de aceptación se muestran a continuación.

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
7	11, 12	Proporcionar los datos para agregar, modificar o eliminar datos pertinentes a los módulos de información del sistema	Se agrega, modifica o elimina correctamente de la base de datos un módulo	Se agregó, se modificó y eliminó correctamente un módulo en la base de datos.
8	13, 14	Proporcionar los datos para agregar, modificar o eliminar datos pertinentes a la configuración de información del sistema	Se agrega, modifica o elimina correctamente de la base de datos una configuración	Se agregó, se modificó y eliminó correctamente una configuración en la base de datos.
9	15	Seleccionar un ítem del menú principal e ir a la funcionalidad correspondiente	La selección de un ítem debe llevar a la funcionalidad identificada por éste	La selección de un ítem lleva a la funcionalidad correspondiente



#### 4.1.5. Iteración 4

La presente iteración muestra la creación de un módulo que se encarga de la gestión de los usuarios. La aplicación va a permitir el manejo de usuarios, el cual va a poseer un determinado perfil. Se cuenta con una serie de funcionalidades como la configuración de la cuenta.

#### Planificación

Iteración 4	
<b>Descripción:</b>	Desarrollo de métodos, modelos y vistas y componentes para la creación y modificación de usuarios de la aplicación.
<b>F.Inicio/F.Fin</b>	23-02 al 09-03-2011
<b>Número</b>	<b>Historia</b>
16	Desarrollar métodos de creación y modificación de usuarios.
17	Desarrollar vistas para la gestión de usuarios.
18	Desarrollar métodos de autenticación.
19	Desarrollar modelos de envío de correo.
20	Desarrollar vistas para envío de correos.

#### Diseño

En esta etapa de la presente iteración se creó un módulo para el manejo de los usuarios. Se puede agregar, modificar o eliminar un usuario determinado, esto con el fin de manejar perfiles ("usuario" y "superusuario") y permitir que se pueda acceder a la aplicación mediante la autenticación.

Los datos solicitados en este módulo son los de identificación de un usuario tales como: nombre, apellido, correo y un login y password para permitirle el ingreso. Cabe destacar que sólo un usuario con perfil "superusuario" tendrá la potestad de agregar un usuario nuevo, donde se le enviará un correo con el login y clave que se proporciona.

El usuario tiene la opción de cambiar la clave al ingresar a la aplicación y en caso de no recordar la contraseña se le es enviado un correo a la dirección asociada con

una nueva clave.

También se desarrollaron las interfaces de manera que se pueda gestionar las funcionalidades correspondientes al manejo de los usuarios, además de las vistas que muestran el inicio de sesión y las asociadas a envío de correo.

## Codificación

Para esta parte se realizaron los métodos requeridos para la gestión de usuarios. Para la autenticación se verifica tanto el login como la clave para poder ingresar a la aplicación (ver figura 4.7).

Se hacen validaciones de correo en caso de envío de una nueva clave, que

```
def autenticar
  login = params[:inicio][:usuario]
  clave = params[:inicio][:clave]

  if login.empty? || clave.empty?
    flash[:mensaje_error] = "Debe introducir todos los datos"
    redirect_to :controller=>"inicio"
    return
  else
    @clave = Digest::MD5.hexdigest(clave)
    u = Usuario.find(:first, :conditions=>["login=? AND password=?", login, @clave])
    if u
      session[:current_user_id] = u.id
      redirect_to :controller=>"principal"
      return
    else
      flash[:mensaje_error] = "Datos incorrectos"
      redirect_to :controller=>"inicio"
      return
    end
  end
end
end
```

Figura 4.7. Código de autenticación

corresponda con el usuario que la solicita.

También, el método cambiar contraseña procesa los datos enviados desde un formulario que son la clave anterior junto con la nueva clave y su confirmación. En caso de no coincidir, se muestra un mensaje al usuario indicando lo sucedido

y es redireccionado de nuevo al formulario. En la figura 4.8 se muestra el método que realiza lo descrito.

```

def cambiar_contrasena
  clave_anterior = params[:formulario][:clave]
  nueva_clave = params[:formulario][:password]
  confirmacion = params[:formulario][:password_confirmation]
  user_id = session[:current_user_id]

  if clave_anterior.empty? or nueva_clave.empty? or confirmacion.empty? then
    flash[:mensaje_error] = "Los campos no pueden estar vacíos;"
    redirect_to :action => "cambiar"
  else
    clave_anterior = Digest::MD5.hexdigest(clave_anterior)
    @u = Usuario.find(:first, :conditions => ["id = ?", user_id])
    if @u.id == user_id then
      if @u.password == vieja_clave and nueva_clave == confirmacion then
        @u.password = Digest::MD5.hexdigest(nueva_clave)
        @u.save
        flash[:mensaje_success] = "Su clave ha sido cambiada"
        redirect_to :action => "personalizar"
        return
      else
        flash[:mensaje_error] = "Datos incorrectos"
        redirect_to :action => :cambiar
        return
      end
    else
      flash[:mensaje_error] = "No puedes realizar esta acción;"
      redirect_to :action => :personalizar
      return
    end
  end
end
end
end

```

Figura 4.8. Código de cambiar contraseña

También se creó un modelo que contiene dos (2) métodos que permiten que al crear un usuario, le será enviado un correo con su nueva cuenta junto con una clave aleatoria. En el caso de que olvide contraseña, el usuario tendrá la opción de recibir una nueva clave al correo registrado en sus datos.

## Pruebas

En esta caso las pruebas realizadas se enfocaron en como los datos que fuesen proporcionados por el usuario se procesaban correctamente y las operaciones con la base de datos fuesen correctas. También se hicieron pruebas de envío de correo utilizando el servicio de correo de gmail.

```

class Correo < ActionMailer::Base

  def forgot_password(usuario, clave)
    @subject = 'Nueva Contraseña'
    @recipients = usuario.correo
    @from = 'ccpdlabucv@gmail.com'
    @body['usuario'] = usuario
    @body['clave'] = clave
    @content_type = "text/html"
  end

  def cuenta_admin(usuario,clave)
    @subject = 'Cuenta de Usuario'
    @recipients = usuario.correo
    @from = 'ccpdlabucv@gmail.com'
    @body['usuario'] = usuario
    @body['clave']= clave
    @content_type = "text/html"
  end
end

```

Figura 4.9. Modelo correo

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
10	16, 17	Proporcionar los datos para agregar, modificar o eliminar los usuarios encargados del sistema	Se agrega, modifica o elimina correctamente un usuario de la base de datos	Se agregó, se modificó y eliminó correctamente un usuario de la base de datos.
11	18	Proporcionar un login y un password para acceder a la aplicación	Validar que el login y password concuerden con los de la base de datos	Se valida correctamente el login y password en la base de datos.
12	19, 20	Crear un usuario para que pueda ingresar a la aplicación	Al crear un usuario se le envía un correo con los datos para ingresar a la aplicación	Al crear un usuario se le envía un correo al usuario para que pueda ingresar a la aplicación
13	19, 20	El usuario olvidó la contraseña	El usuario coloca su correo el cual se valida y se le envía un correo con nueva clave	El usuario al colocar el correo válido se le envía la nueva clave al mencionado correo

#### 4.1.6. Iteración 5

En esta iteración trata de como una herramienta resulta muy útil a la hora de procesar tareas que pueden tardar mucho tiempo en procesarse. Esto con el fin de manejar el procesamiento de tareas en segundo plano y que sea separado del ciclo request/response.

#### Planificación

<b>Iteración 5</b>	
<b>Descripción:</b>	Instalación de plugin para background, procesamiento de imagen y creación de thumbnail
<b>F.Inicio/F.Fin</b>	10-03 al 29-03-2011
<b>Número</b>	<b>Historia</b>
21	Configuración de plugin BackgroundRb.
22	Creación de workers.
23	Creación de procesador de imágenes.
24	Instalación de herramienta wkhtmltoimage.
25	Desarrollar método para creación de thumbnail.

#### Diseño

En esta iteración se trabajo con una herramienta llamada “wkhtmltoimage” que permite crear las imágenes preview de las direcciones web que sean agregadas a un grupo. Esto con el fin de mostrar las imágenes en las interfaces asociadas a la gestión de grupos y direcciones.

Dado que era requerido el procesamiento de tareas en background se utilizó el plugin BackgroundDRB (ver figura 4.10). Mediante esta, se tiene la posibilidad de descargar la imagen preview antes mencionada de la dirección sin afectar el ciclo request/response y así no mantener la aplicación ocupada por mucho tiempo.

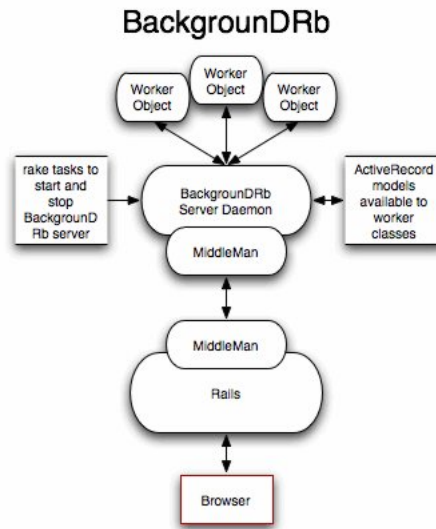


Figura 4.10. Arquitectura de BackgroundDRb[39]

## Codificación

Para poder utilizar la herramienta que obtiene la imagen preview se procedió a la descarga de ésta. Mediante una consola en linux, el comando se ejecuta pasando como parámetros la dirección del sitio y el archivo de salida con el nombre correspondiente.

Otro actividad desarrollada fue un modelo que se encargara de crear una imagen que tuviese un tamaño adecuado para que pudiese ser mostrado en la interfaz gráfica del módulo de grupos.(ver figura 4.11).

La descarga y configuración de BackgroundDRb fue realizado de la siguiente manera:

1. Ya que el plugin dependía de algunas gemas se hizo primero la instalación de ellas

```
sudo gem install chronic packet
```

```

class Imagen < ActiveRecord::Base
  require 'rubygems'
  require 'RMagick'
  class << self
    def crear_thumbnail(aux)
      imagen = aux[0]
      ruta = aux[1]
      info_url = aux[2]
      img_orig = Magick::Image.read("#{ruta}/#{info_url.id_proceso}_#{info_url.id}.jpg").
      thumb = img_orig.resize to fit(200, 200)
      thumb.write("#{ruta}/#{info_url.id_proceso}_#{info_url.id}_thumb.jpg".gsub("//", '
    end
  end
end

```

Figura 4.11. Modelo donde se crea la imagen thumbnail

2. Se descargó el plugin desde github

```
git clone git://github.com/gnufied/backgroundrb.git
```

3. Desde la aplicación en rails se procedió a la configuración mediante el comando `rake backgroundrb:setup`. Éste genera una serie de archivos indispensables para el funcionamiento del plugin.
4. para crear un worker se ejecuta `./script/generate worker billing`

Se implemento una acción en un worker creado con el nombre Imagen (ver figura 4.12) que se encarga de hacer la llamada al modelo Imagen donde se desarrolló un método que se encarga de realizar el proceso que de descarga de la imagen con la herramienta antes mencionada. Agregado a esto, el worker Imagen contenía otra acción que se iba a encargar del proceso de creación del thumbnail de la imagen. Esto último se logró llevar a cabo mediante la librería de procesamiento de imágenes ImageMagick en conjunto con RMagick.

## Pruebas

Las pruebas que se realizaron en esta iteración fueron, entre otras, verificar que la herramienta `wkhtmltoimage` estuviese creando las imágenes correctamente. Hubo algunos casos en donde no se lograba descargar las imágenes. También se hicieron varias pruebas de como funcionaba y que procesos ejecutaba el plugin

```
class ImagenWorker < BackgroundDRb::MetaWorker
  set_worker_name :imagen_worker
  def create(args = nil)
    # this method is called, when worker is loaded for the first time
  end

  def crear_imagen(aux)
    logger.info 'creando imagen url de la pagina'
    Url.crear_imagen(aux)
  end

  def crear_thumbnail(aux)
    logger.info 'creando thumbnail'
    Imagen.crear_thumbnail(aux)
  end
end
```

Figura 4.12. Workers creado con BackgroundDRb

*BackgroundDRb*. Luego de conocer su funcionamiento se procedió a la creación de los workers que efectuaban la tarea establecida de forma exitosa.



#### 4.1.7. Iteración 6

Dado que en cada uno de los subsistemas se situaba en una base de datos local y habían datos en común, se decidió que hubiese sólo una base de datos centralizada para tener un mejor control. En esta iteración se realizó la integración de las bases de datos en el Controlador Central, esto para evitar inconsistencia en los datos. Se efectuaron cambios tales como la migración de los modelos ActiveRecord a ActiveResource, así como la adaptación de los métodos para que funcionarían tal cual lo hacían antes de la migración. Además, se hizo un cambio del plugin utilizado para tareas en segundo plano.

<b>Número:</b> 26	<b>Nombre:</b> Configuración de Workling & Starling	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 4 días	<b>Tipo:</b> Modificación
<b>Descripción:</b> Configuración de plugin Workling e instalación de gema Starling para realizar tareas en background.		
<b>Número:</b> 27	<b>Nombre:</b> Adaptación de workers	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 4 días	<b>Tipo:</b> Modificación
<b>Descripción:</b> Adaptación de workers desarrollados con el plugin Backgroundrb a Workling.		
<b>Número:</b> 28	<b>Nombre:</b> Migración de modelos a ActiveResource	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 1 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Colocar modelos en ActiveRecord a ActiveResource para el manejo de recursos.		
<b>Número:</b> 29	<b>Nombre:</b> Adaptación de métodos a ActiveResource	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 6 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Adaptación de los métodos desarrollados para ActiveRecord a ActiveResource.		
<b>Número:</b> 30	<b>Nombre:</b> Integración de base de datos	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Modificación
<b>Descripción:</b> Integración de base de datos de los distintos subsistemas que componen el sistema en general.		

## Planificación

Iteración 6	
<b>Descripción:</b>	Migración desde Active Record a Active Resource, integración de base de datos y plugin para background.
<b>F.Inicio/F.Fin</b>	30-03 al 21-04-2011
<b>Número</b>	<b>Historia</b>
26	Configuración de working y starling.
27	Adaptación de workers.
28	Migración de modelos a ActiveRecord.
29	Adaptación de métodos a ActiveRecord.
30	Integración de base de datos.

## Diseño

Debido a que se tenía la necesidad de migrar la base de datos, surgió un nuevo modelo de datos ya que, a pesar de tener tablas en común, se manejaban datos particulares en cada lugar. En la figura 4.13 se pueden observar los cambios que fueron hechos sobre el diseño de la base de datos.

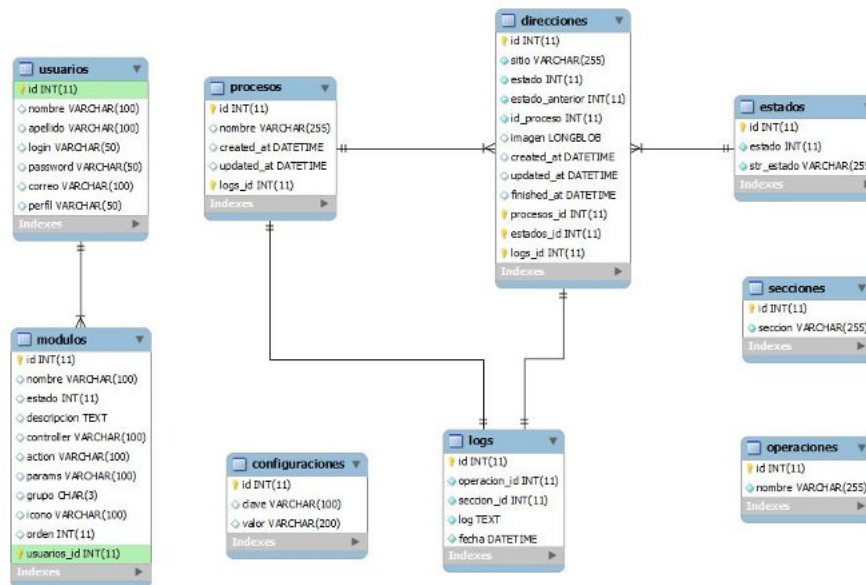


Figura 4.13. Modelo de Datos del Controlador Central

Otro de los cambios realizados fue el de sustituir el plugin de procesamiento de tareas en en segundo plano “backgroundDRb” ya que resultaba ineficiente. El nuevo plugin a utilizar era “Workling & Starling”. Este último actúa como un servidor de colas y fue desarrollado por “Twitter” para el procesamiento de tareas en background.

### **Codificación**

Para el desarrollo, primordialmente el enfoque fue el de migrar los métodos de los módulos desde ActiveRecord a ActiveResource. Para esto se crearon una serie de parches que facilitarían el funcionamiento de los métodos CRUD principalmente. La integración con la base de datos se llevo a cabo de manera satisfactoria migrando tabla por tabla y eliminando las tablas que ya no se utilizarían en esta fase.

La instalación y configuración de workling y starling se efectuó de manera mucho más sencilla que backgroundDRb, a pesar de que no tenían muchas diferencias, de esta manera, solo se hizo la adaptación de los workers al nuevo plugin.

### **Pruebas**

En la presente iteración, las pruebas efectuadas fueron principalmente las que ya se realizaron en las iteraciones 1, 2, 3 y 4. Estas se basaron en que los métodos de crear, modificar y eliminar, funcionaran de forma correcta en cada uno de los módulos desarrollados con Active Resource. En el siguiente cuadro se muestra de forma general las pruebas que se ejecutaron.

No.	H.U. Involucradas	Descripción del caso de prueba	Resultado esperado	Resultado obtenido
14	28, 29 y 30	Proporcionar los datos para agregar, modificar o eliminar información en cada funcionalidad desarrollada	Se agregan, modifican o eliminan correctamente elementos de la base de datos al agregar, modificar o eliminar un determinado item	Se agregan, modifican o eliminan correctamente elementos de la base de datos al agregar, modificar o eliminar un determinado item

#### 4.1.8. Iteración 7

Con el fin de brindarle al usuario una manera de asociar o clasificar las direcciones que se agregan a un grupo, se desarrolló un método que permitiera, mediante un clasificador, asignar a cada dirección una o varias categorías, de acuerdo a un entrenamiento previo de texto de páginas web. Esto se convertiría como en una especie de etiquetas que identifiquen que temas puede manejar una página.

<b>Número:</b> 31	<b>Nombre:</b> Desarrollar método de clasificación de direcciones	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 5 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Desarrollo de método de clasificación de direcciones mediante filtro bayesiano.		

<b>Número:</b> 32	<b>Nombre:</b> Desarrollar vistas de búsqueda	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Creación de vistas de la búsqueda de direcciones mediante filtro bayesiano.		

#### Planificación

Iteración 7	
<b>Descripción:</b>	Implementación de algoritmo de clasificación para búsqueda.
<b>F.Inicio/F.Fin</b>	22-04 al 03-05-2011
<b>Número</b>	<b>Historia</b>
31	Desarrollar método de clasificación de direcciones.
32	Desarrollar vistas de búsqueda.

#### Diseño

En esta iteración, se utilizó particularmente el clasificador Bayesiano Ingenuo. Este utiliza el teorema matemático de Bayes, que habla de las probabilidades condicionales de eventos y es ingenuo porque supone que cada evento es totalmente independiente entre sí. Los pasos a seguir para obtener los resultados esperados son los siguientes:

1. Indicarle al clasificador cuántas categorías habrá.

2. Entrenar al clasificador mediante una serie de documentos indicándole a que categoría pertenecen.
3. Pasar un documento por el clasificador y este debe señalar a qué categoría pertenece.

#### 4.1.9. Iteración 8

Durante esta iteración se realizó el análisis y selección de las posibles máquinas virtuales que se pueden instalar en el servidor destinado para el alojamiento de las aplicaciones desarrolladas para la plataforma de la “nube”, esta elección se realizó basándose en las características dadas tanto del esquema de las máquinas virtuales como de las características del servidor. Después de seleccionar la máquina virtual se configura e instala y se adapta a la configuración de cloud computing. Al término se realiza la implementación de los trabajos (jobs) que van a ejecutarse en cada uno de los nodos generados en la “nube”.

<b>Número:</b> 33	<b>Nombre:</b> Instalación de KVM	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Instalación de KVM para virtualización.		

<b>Número:</b> 34	<b>Nombre:</b> Configuración de máquinas virtuales	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Instalación y configuración de máquinas virtuales para Cloud Computing.		

<b>Número:</b> 35	<b>Nombre:</b> Configuración de Cloud Computing	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nuevo
<b>Descripción:</b> Configuración de Cloud Computing para procesamiento de tareas por lotes.		

<b>Número:</b> 36	<b>Nombre:</b> Colocar jobs en la nube	
<b>Prioridad:</b> Alto	<b>Estimación:</b> 2 días	<b>Tipo:</b> Nueva
<b>Descripción:</b> Configuración de jobs para procesamiento en la nube.		

#### Planificación

Iteración 8	
<b>Descripción:</b>	Instalación y configuración de máquinas virtuales y cloud computing.
<b>F.Inicio/F.Fin</b>	04-05 23-05-2011
<b>Número</b>	<b>Historia</b>
33	Instalación de KVM.
34	Configuración de máquinas virtuales.
35	Configuración de cloud computing.
36	Colocar jobs en la nube

## Diseño

Se desea implementar una estructura virtual conformada múltiples máquinas virtuales alojadas en un servidor físico, debido a la gran demanda de procesamiento que esto implica, se necesita un servidor de gran escala (en nuestro caso se utiliza un procesador de ocho (8) núcleos) cuyo sistema operativo es Ubuntu Server 10.10. Para realizar la virtualización se elige KVM (kernel virtual machine) como base para la generación de los nodos de la “nube”, esto debido al alto acoplamiento que tiene con Ubuntu Server, para la configuración se realizaron las tareas de:

1. Comprobación de compatibilidad con el tipo de procesador.
2. Descargar e instalar el paquete KVM.
3. Incluir usuarios en el grupo de usuarios KVM.

Durante la fase de configuración de cloud computing se realizan todas las asignaciones de los nodos creados con cada una de las asignaciones y tareas de cada uno con respecto a las aplicaciones desarrolladas para el proyecto.



## 4.2. Resultados



Figura 4.14. Menú principal

En la figura 4.14 se puede observar la forma como se divide el menú principal: Procesamiento, Gestión de Datos y Administración.



Figura 4.15. Lista de grupos creados

En la figura 4.15 se aprecia la lista de los grupos de direcciones web que han sido creados por un usuario determinado.



**Figura 4.16.** Creación de grupo nuevo

Aquí (figura 4.16) se muestra la forma como se puede crear un nuevo grupo de direcciones web. Se coloca un nombre propuesto por el usuario y se agregan las direcciones una por una al grupo.



**Figura 4.17.** Dirección agregada a un grupo nuevo

Luego de agregar una dirección a un grupo se muestra que fue agregada de forma exitosa colocándola dentro del grupo(ver figura4.17).



**Figura 4.18.** Lista de direcciones que contiene un grupo

En la figura 4.18 se puede observar la lista de direcciones web que contiene un grupo, también se aprecia que luego de agregar una dirección se genera una imagen preview de ésta.



**Figura 4.19.** Opción de modificar grupo de direcciones

Además de la opción de agregar un nuevo grupo, existe la posibilidad de modificar un grupo como se puede ver en la figura 4.19



Figura 4.20. Lista de configuraciones

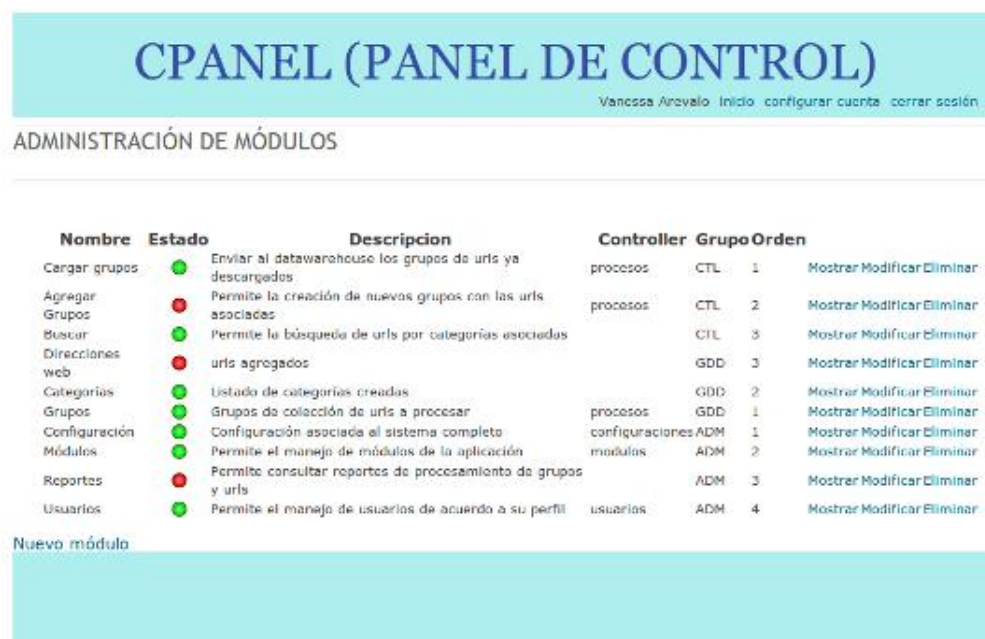


Figura 4.21. Lista de módulos de la aplicación

En las figuras 4.20 y 4.21 se observan los componentes pertenecientes al conjunto Administración, que es donde el usuario tiene la potestad de poder manejar como se presenta la información en la aplicación.

The screenshot shows a web application interface. At the top, there is a light blue header with the text 'CPANEL (PANEL DE CONTROL)' in a large, bold, blue font. Below this, in a smaller font, are the links 'Vanessa Arevalo Inicio configurar cuenta cerrar sesión'. Underneath the header, the text 'MODIFICAR MÓDULO' is displayed. The main content area contains a form with the following fields: 'Nombre' (text input), 'Cargar grupos' (text input), 'Estado' (text input), and '1' (text input). Below these is a 'Descripción' section with a text area containing the text 'Ejecutar el sistema de los grupos de usuarios ya descargados'. A mouse cursor is visible over the text area.

**Figura 4.22.** Modificar módulo seleccionado

El usuario tiene la posibilidad de modificar un módulo contenido en la aplicación. En la figura 4.22 se observan algunos de los campos solicitados.

Para que un usuario pueda ingresar y utilizar la aplicación, este debe estar registrado en ésta (ver figura 4.23).

En la figura 4.24 se muestra el formulario con los datos que solicitan para agregar un nuevo usuario.

Además, luego de registrarse un usuario, éste va a poder ingresar a la aplicación y cambiar su clave a la deseada (ver figure 4.25).



Figura 4.23. Lista de usuarios registrados



Figura 4.24. Agregar nuevo usuario



The image shows a web interface for a control panel. At the top, there is a light blue header with the text "CPANEL (PANEL DE CONTROL)" in a large, dark blue font. Below this, in a smaller font, are the links "Vanessa Arevalo", "inicio", "configurar cuenta", and "cerrar sesión". The main content area is titled "CAMBIAR CLAVE" in a bold, dark blue font. Below the title, there are three input fields: "Clave anterior:", "Nueva Clave:", and "Confirmación de clave:". At the bottom of the form, there are two buttons: "Actualizar" and "Volver". The entire form is set against a light blue background.

CPANEL (PANEL DE CONTROL)

Vanessa Arevalo inicio configurar cuenta cerrar sesión

CAMBIAR CLAVE

Clave anterior:

Nueva Clave:

Confirmación de clave:

Figura 4.25. Cambiar clave de usuario





---

## Conclusiones y Recomendaciones

La investigación realizada anteriormente permitió la construcción y manejo de herramientas para llevar a cabo los objetivos planteados. También para que el desarrollo fuese logrado con éxito el utilizar el método de programación eXtreme Programming (XP) fué de gran apoyo y ayuda para la culminación del presente Trabajo Especial de Grado.

Se logró la construcción de un aplicación que permita la gestión de grupos de direcciones web que sean de interés para el repositorio así como también de otras funcionalidades aplicando el patrón de diseño MVC. Esta aplicación forma parte de un prototipo para un archivo de documentos de la Web. Mediante ésta se van a poder solicitar al controlador central las direcciones web que el usuario desee descargar o actualizar.

El haber utilizado las tecnologías de Ruby on Rails, jQuery y AJAX fue de mucho apoyo ya que muchas de las funcionalidades se lograron mediante varias de las funciones que contienen los frameworks.

El procesamiento de tareas en segundo plano fue necesario debido a las actividades que requerían de un tiempo de procesamiento largo. Esto permitía a la aplicación seguir en funcionamiento y no esperar por que terminara la mencionada tarea.

La idea de implementar un sistema de virtualización fue de gran importancia ya que esto permite que muchas tareas se ejecuten de manera eficiente y el tiempo de

respuesta sea menor, aprovechando así todos los beneficios que nos brindan este tipo de tecnologías junto con la Computación en la Nube.

Cabe destacar que además de haberse cumplido con los objetivos, se desarrollaron cualidades y se obtuvo experiencia tanto en la investigación como en el desarrollo. En cuanto al aporte que se logrará obtener con este sistema es el de ofrecer un prototipo como paso inicial para la construcción de un archivo histórico donde se pueda almacenar las páginas Web en forma de versión.

Esto sería un pequeño avance ya que este tema es bastante complejo, como se indica en las directrices de la UNESCO en cuanto a la preservación digital.

Actualmente en Venezuela se ha estado creando mucha información que no es integrada y no es recopilada de forma eficiente. Son muchos los beneficios que se obtienen con la construcción de un sistema tal envergadura. Esto va a ser de mucha utilidad tanto para las generaciones actuales como las futuras ayudando a la presevación y difusión de conocimientos.

## **Limitaciones**

- Algunas páginas no tienen información en la página principal cuando son rastreadas.
- La herramienta de descarga de imagen preview de las páginas web `wkhtmltoimage` no toma la imagen de páginas en flash.
- La adaptación de los métodos a `ActiveResource` fue bastante engorroso.
- El plugin para el procesamiento de tareas en segundo plano no permite el control de lo que se procesa, por ejemplo, no se podía detener el proceso de descarga.

### **Recomendaciones**

- Establecer criterios de selección de acuerdo al contenido de las direcciones en el dominio de Venezuela
- Colocar el enfoque en el dominio venezolano
- Establecer niveles de profundidad de búsqueda
- No utilizar ARes y Estudiar la posibilidad de usar Sinatra como framework de comunicaciones
- Estudiar la posibilidad de usar Delayed Job para manejo de tareas en segundo plano

### **Trabajos Futuros**

- Lograr el funcionamiento de las máquinas virtuales
- Integrar la aplicación con los subsistemas por desarrollar
- Mejorar la interfaz en cuanto a la interacción con el usuario



## Anexos

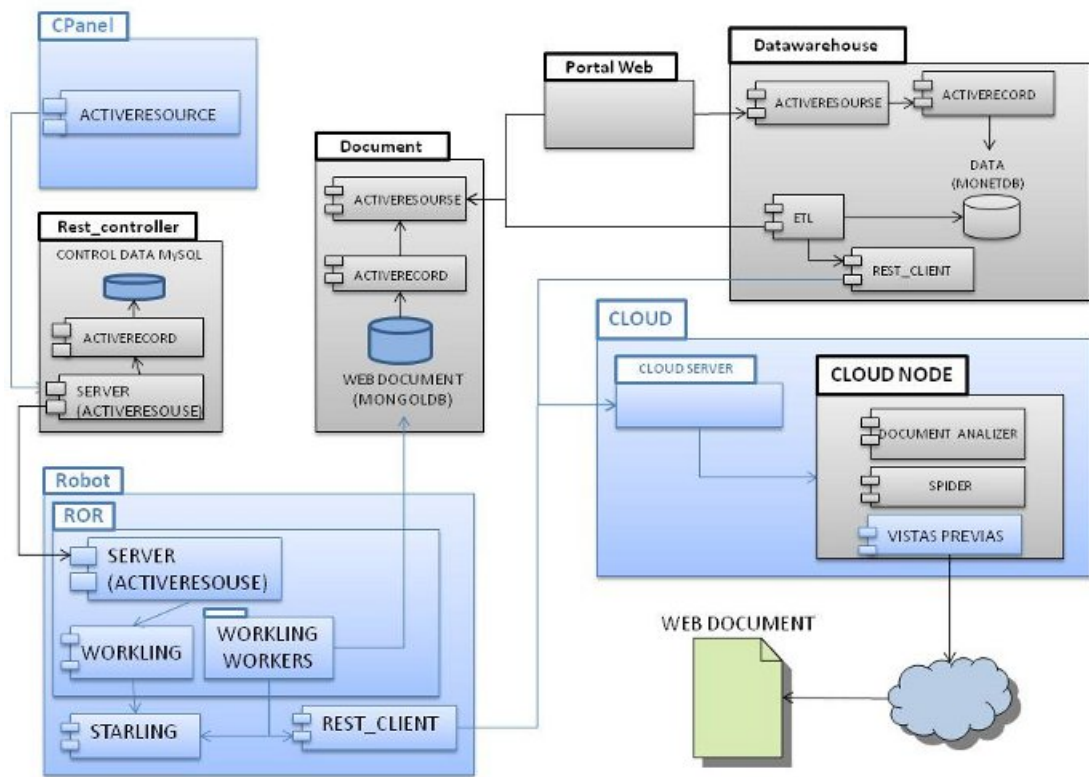


Figura 4.26. Arquitectura detalla del proyecto



---

## Referencias

1. Ambrust, M. et al. *Above the Clouds: A Berkeley View of Cloud Computing*. Electrical Engineering and Computer Sciences, University of California at Berkeley, California, Technical Report UCB/EECS-2009-28, 2009.
2. Bennett, S., Bhuller, M., and Covington, R.(2009) *Architectural Strategies for Cloud Computing* Oracle Corporation.
3. Bia, A. y Sánchez, M.*Desarrollo de una política de preservación digital: tecnología, planificación y perseverancia*. Biblioteca Virtual Miguel de Cervantes, Universidad de Alicante.
4. Black, D.(2006) *Ruby for Rails* Greenwich: Manning.
5. Boss, G., Malladi, P., Quan, D., Legregni, L., y Hall, H.(2007)*Cloud Computing* IBM Corporation.
6. Burfor, D.(2008)*Virtualization. It is right for you?*. Lad Enterprizes, INC.
7. Chaffer, J. y Swedberg, K.(2007) *Learning jQuery. Better Interaction Design and Web Development with Simple JavaScript Techniques*. Birmingham: Packt Publishing.
8. Cisco (2009) *Private Cloud Computing for Enterprises*. Cisco White Paper.
9. Cisco Systems, Inc.(2009) *The Cisco Powered Network Cloud: An Exciting Managed Services Opportunity*. Cisco Systems, Inc.
10. Dver, A.(2008) *Enemy of SaaS?* Software Magazine, vol. 27, no. 6, p. 24.
11. Doug, R.(2002) *Inside The Internet Archive*. Extraído el 02 de mayo de 2011 desde <http://www.mindjack.com/feature/archive.html>
12. El desarrollo Web que no molesta. Extraído el 25 de diciembre de 2009 desde <http://www.rubyonrails.org.es/>
13. Escalante, S. y Ovalles, G.(2010) *Desarrollo de los Módulos de Constancias y de Comprobantes para el Sistema CONEST Postgrado*. UCV.
14. Fisher, T.(2008) *Ruby on Rails Bible (1a Ed.)*. Indiana, USA: Wiley Publishing, Inc.
15. García, R.(Noviembre 2008)*Desarrollo e Implementación de un Sistema de Colaboracin Web con Metodología Ágil, P.F.C*. Madrid.

16. Germán, M.(2005)*Extreme Programming*. Extraído el 08 de mayo de 2011 desde [http://curso\\_sin2.blogia.com/2005/070905-extreme-programming-x.p.-por-german-marcano.php](http://curso_sin2.blogia.com/2005/070905-extreme-programming-x.p.-por-german-marcano.php)
17. González, Z.(2007) *Estudio de Métricas de Software. Caso de estudio: Ruby on Rails y Frameworks en Java* UCV.
18. Ibáñez, A.(2011) *Wayback Machine: la máquina del tiempo de las páginas web se renueva*. Extraído el 02 de mayo de 2011 desde <http://www.rtve.es/noticias/20110129/waybackmachine-maquina-del-tiempo-paginas-web-se-renueva/399043.shtml>
19. *Internet Archive*. Extraído el 02 de mayo de 2011 desde <http://www.archive.org/about/about.php>
20. Kajeepeta, S. Cloud Computing: From Metaphor to Mainstream. *Software Magazine*, vol. 27, no. 6, pp. 1013, Nov. 2008.
21. Kamaraju, A. y Nicolas, P.(2009).*Cloud Storage*. Storage Networking Industry Association.
22. Lasica, J. D.(2009)*Identity in the Age of Cloud Computing*. United States of America: The Aspen Institute.
23. Lenz, P.(2007). *Build Your Own Ruby On Rails Web Applications*. (1a Ed.). USA: SitePoint.
24. Letelier, P. y Penadé, M. *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia.
25. *Llevando la Teoría a la Práctica. Tutorial de Digitalización de Imágenes*. Extraído el 01 de mayo de 2011 desde <http://www.library.cornell.edu/preservation/tutorial-spanish/preservation/preservation05.html>
26. Márquez, M. y Fernández, W.(2008) *Tópicos para el desarrollo de un Módulo de generación de reportes del sistema CONEST*. Universidad Central de Venezuela.
27. Mendoza, A. (2007). *Utility Computing Technologies, Standards and Strategies*. United States of America: Artech House, Inc.
28. Notess, G.(2002).*The Wayback Machine: The Web's Archive*. ONLINE. Vol 26. No.2.
29. Ohlman, B., Eriksson, A. y Rembarz, R.(2009)*What Networking of Information Can Do for Cloud Computing* 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises. pp. 7883.
30. Power System (2009).*Introducción a la Virtualización*. IBM.
31. Pressman, R. (2007) *Ingeniería del Software. Un enfoque práctico*. Sexta Edición. Editorial Mc Graw Hill.
32. Quan, D.(2008) *From Cloud Computing to the New Enterprise Data Center*. IBM Corporation.
33. Ruby, S., Thomas, D. y Heinemeier D.(2009, marzo) *Agile Web Development with Rails* (3ra Ed.). USA: The Pragmatic Bookshelf.
34. Sun Microsystems (2009) *Introduction to Cloud Computing Architecture* Sun Microsystems White Paper.
35. Taylor, D.(2002) *Dynamic HTML Weekend Crash Course*. New York: Hungry Minds.



36. Thomas, D.(2009) *Cloud Computing Benefits and Challenges!* Journal of Object Technology, vol. 8, no. 3, pp. 3741.
37. Tim Jones, M.(2009) *Virtual Linux. An overview of virtualization methods, architectures, and implementations.* IBM Developerworks.
38. UNESCO(2003)*Directrices para la Preservación del Patrimonio Digital (Resumen)*
39. Zygmuntowicz, E.,(2007) *Introduction to BackgroundDRb.* Extraído el 30 de marzo de 2011 desde <http://www.infoq.com/articles/BackgroundDRb>

