Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información - CISI

Implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web en Venezuela.

Trabajo Especial de Grado
presentado ante la Ilustre
Universidad Central de Venezuela
por los Bachilleres
Laura Gabriela Sánchez Castillo (C.I. 19.672.992)
Gamar Francisco Edgardo Milano Aponte (C.I. 19789566)
para optar al título de
Licenciado en Computación

Tutora: Profa. Mercy Ospina

Caracas, 14 de Mayo de 2015

ACTA

Quienes suscriben, miembros del jurado designado por el Consejo de la Escuela de Computación, para examinar el Trabajo Especial de Grado titulado **"Implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web en Venezuela"** y presentado por los Bachilleres Laura Gabriela Sánchez Castillo C.I: 19672992 y Gamar Francisco Edgardo Milano Aponte C.I: 19789566, a los fines de optar al título de Licenciado en Computación, dejamos constancia de lo siguiente:

Leído como fue dicho trabajo, por cada uno de los miembros del jurado, se fijó el día 14 de Mayo de 2015, a las 07:00 am, para que los autores lo defendieran en forma pública, lo que estos hicieron en la Sala de Postgrado de la Escuela de Computación, mediante una presentación oral de su contenido, luego de lo cual respondieron a las preguntas formuladas. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobar con la nota de 20 puntos.

En fe de lo cual se levanta la presente Acta, en Caracas el día 14 de Mayo de 2015.

Profa. Mercy Ospina Torres
(Tutora)

Profa.Concettina Di Vasta
(Jurado)

Profa. Yusneyi Carballo
(Jurado)

Agradecimientos

Nos gustaría que estas líneas sirvieran para expresar nuestro más profundo y sincero agradecimiento.

Primeramente a ti Dios por bendecirnos para llegar hasta donde hemos llegado, porque hiciste realidad este sueño anhelado.

A la ilustre Universidad Central de Venezuela por darnos la oportunidad de estudiar y ser profesionales. También nos gustaría agradecer a nuestros profesores durante toda nuestra carrera profesional, porque todos han aportado con un granito de arena a nuestra formación.

A todas aquellas personas que con su ayuda han colaborado en la realización del presente trabajo, en especial a la Profesora Mercy Ospina, tutora de esta investigación, por la orientación, el seguimiento y la supervisión continúa de la misma, pero sobre todo por el apoyo recibido.

Nuestro más profundo agradecimiento a Hilda Pérez, por ser nuestra amiga en los momentos más difíciles, tanto en nuestra vida personal como en el ámbito académico. Gracias por confiar en nosotros cuando los demás dudaban.

Un agradecimiento muy especial merece la comprensión, paciencia y el ánimo recibidos de nuestros familiares y amigos.

A todos ellos, muchas gracias.

Dedicatorias

*Dedico este Trabajo de Grado a mis padres por todo el apoyo que me han brindado durante todos mis años de estudio, por sus consejos, sus valores, los ejemplos de perseverancia y constancia que los caracterizan a ambos. A mis familiares y amigos que indudablemente son una parte importante de mi vida, por estar en los buenos y malos momentos, y por apoyarnos mutuamente en nuestra formación profesional**: Hilda Pérez, Gamar Milano, Claudio Torrez, Aldemaro Díaz, Jhonny Acevedo, Keyli Cobo, Karylin Gómez, Anakarenina Vivas, Nancy Sánchez, Judith Sánchez, Lowtijie Rivero** y a todos los que participaron directa o indirectamente en la elaboración de esta tesis ¡Gracias a ustedes!*

Laura Sánchez

A Dios. *Por haberme permitido llegar hasta este punto y haberme dado salud para lograr mis objetivos.* ***A mi madre.*** *Por haberme apoyado en todo momento, por sus consejos, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por su amor.* ***A mi abuela (QEPD),*** *por cuidarme desde el cielo, por haber sido un ejemplo de perseverancia y constancia, esto también te lo debo a ti. A mis tíos,* ***Mariela y Jorge****, que han sido apoyo absoluto y a mis primos,* ***Mathias y Eduardo Arturo****, impulso principal para seguir adelante.* ***A Annette Annia, Laura Sánchez, Hilda Pérez y Anakarenina*** *Vivas, amigas incondicionales, con quien puedo compartir los buenos y malos momentos.* **A mi Abuelo A. Aponte***. Por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por los valores que me ha infundado siempre, por el valor mostrado para salir adelante.*

Gamar Milano

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Sistemas de Información - CISI

Implementación del Módulo de Gestión y de Control de Incidencias del Prototipo de Archivo Web en Venezuela.

Autores: Laura Gabriela Sánchez Castillo
CI: 19.672.992
Correo:lau.san029@gmail.com

Gamar Francisco E. Milano Aponte
C.I: 19.789.566
Correo: gmilano19@gmail.com

Tutora: Profa. Mercy Ospina

Fecha: 14 de Mayo de 2015

RESUMEN

La preservación del patrimonio cultural en formato digital es un tema que se ha venido desarrollando de manera amplia en los últimos años, en especial aquel patrimonio en formato Web. Esto debido a la cantidad e importancia del contenido que se ha generado y su naturaleza efímera. El presente trabajo de investigación se enfocó en la implementación del Módulo Gestión y Control de Incidencias como parte de la segunda etapa de desarrollo del Prototipo de Archivo Web implementado en la Escuela de Computación para la preservación de sitios web del país. El objetivo del módulo es mejorar algunas funcionalidades de la primera etapa, entre estas implementar algunas funcionalidades administrativas, como el control de las fallas que puedan surgir en el proceso de adquisición las cuales son importantes para la recuperación del sistema, y la aprobación y supervisión de los trabajos de rastreos como una forma de validación de los contenidos a preservar, adicionalmente se implementó las división de los roles para el ingreso al aplicativo y la gestión de las diferentes solicitudes generadas por los usuarios. Para el desarrollo de este módulo se combinaron las tecnologías *Python* (lenguaje de programación), *Django* (*Framework*), *MySQL* (sistema manejador de base de datos) y la Metodología de Desarrollo *AUP*. El Módulo fue probado a través de: la adquisición de versiones de sitios web previamente seleccionados, la validación de las solicitudes de usuarios, y la simulación de fallas para verificar el reinicio de los trabajos de rastreos por inconvenientes ocasionados con el servicio eléctrico.

Palabras Clave: Archivo Web, Preservación, gestión y control, incidencias, rastreo, adquisición, Prototipo.

Índice

AGRADECIMIENTOS ..... **III**
DEDICATORIAS ..... **IV**
RESUMEN ..... **V**
ÍNDICE DE FIGURAS ..... **VIII**
ÍNDICE DE TABLAS ..... **X**
INTRODUCCIÓN ..... **XI**
CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA **13**
1.1 OBJETIVO GENERAL 15
1.2 OBJETIVOS ESPECÍFICOS 15
1.3 JUSTIFICACIÓN 16
1.4 ALCANCE 17
CAPÍTULO II MARCO CONCEPTUAL **18**
2.1 LA PRESERVACIÓN DIGITAL Y LA PRESERVACIÓN WEB 18
2.2 ARCHIVOS WEB 21
2.3 ANTECEDENTES DE MÓDULOS O FUNCIONES DE ADMINISTRACIÓN EN ARCHIVOS WEB 23
2.3.1 Internet Archive *23*
2.3.2 PANDORA (Australia's Web Archive) *23*
2.3.3 Módulos de Administración *24*
2.4 ESTADO DEL ARTE: PROTOTIPO DE ARCHIVO WEB DE VENEZUELA 25
2.4.1 Adquisición *25*
2.4.2 Almacenamiento *28*
2.4.2.1 Formato WARC *28*
2.4.3 Indexación y Motores de Búsqueda *31*
2.4.4 Herramienta de Rastreo: Heritrix *31*
2.4.4.1 Principales Componentes 33
2.4.4.2 Configuración Básica de los Rastreos 35
2.4.4.3 *Logs* de Rastreos 36
2.4.4.4 Proceso de Rastreo en el Prototipo de Archivo Web 37
2.4.5 Acceso *39*
2.4.6 Tecnologías Usadas *40*
2.4.6.1 Software 40
2.4.6.2 Hardware 40
2.4.7 Arquitectura de la aplicación actual *41*
2.4.8 Interfaces *41*
2.4.8.1 Aplicación administrativa 42
2.4.8.2 Aplicación de acceso 44
2.4.9 Lenguaje de Programación: Ruby *46*
2.4.9.1 Framework Rails 46
2.4.10 Lenguaje de Programación: Python *46*
2.4.10.1 Características: 47
2.4.10.2 Frameworks para Python 47
2.4.10.3 Django 48
2.4.10.4 Características 48
2.4.10.5 Python vs Ruby 48
CAPÍTULO III MARCO METODOLÓGICO **50**
3.1 METODOLOGÍAS DE DESARROLLO DE SOFTWARE 50

3.2 METODOLOGÍA AUP 52
3.2.1 Disciplinas o Iteraciones. 52
3.2.2 Fases 53
3.2.3 Entrega de versiones incrementales en el tiempo..... 55
3.3 ARQUITECTURA DE SOFTWARE BASADA EN COMPONENTES..... 55
CAPÍTULO IV DESARROLLO DE LA APLICACIÓN 60
4.1 OBJETIVO GENERAL DE LA APLICACIÓN..... 60
4.2 OBJETIVOS ESPECÍFICOS DE LA APLICACIÓN 60
4.3 ALCANCE DE LA APLICACIÓN 60
4.4 ADAPTACIÓN DE LA METODOLOGÍA AUP USANDO UNA ARQUITECTURA DE SOFTWARE BASADA EN COMPONENTES 61
4.4.1 Fase de Inicio..... 61
4.4.1.1 Requisitos Funcionales 61
4.4.1.2 Usuarios del Sistema.....62
4.4.1.3 Arquitectura de la Aplicación 63
4.4.1.4 Entorno de Trabajo 63
4.4.2 Fase de Elaboración 64
4.4.2.1 Prototipos de Interfaz de Usuario.....64
4.4.2.2 Diagramas de Componentes.....68
4.4.2.3 Diagramas de Actividades.....69
4.4.2.4 Casos de Uso y Diagramas de Secuencia 70
4.4.2.5 Modelo de datos 94
4.4.3 Fase de Construcción 95
4.4.3.1 Implementación de componentes 95
4.4.4 Fase de Transición 107
4.4.4.1 Puesta en producción del Prototipo de Preservación Web 107
4.4.4.2 Pruebas Funcionales.....108
CONCLUSIONES 125
TRABAJOS FUTUROS 126
REFERENCIAS BIBLIOGRÁFICAS.....127
ANEXO A INSTALACIÓN DE HERITRIX VERSIÓN 3.2.0 129
ANEXO B CONEXIÓN ENTRE LAS DIFERENTES MAQUINAS VIRTUALES EXISTENTES.....132

Índice de Figuras

Figura 1. Mapa de países con Iniciativas de Archivo Web en el mundo (IIPC, 2014) 21
Figura 2. Actividades para la Preservación Web (Ospina, Martinez, Leon, & Kabchi, 2014) . 22
Figura 3. Archivado del lado del cliente (Ospina M. , 2014) 26
Figura 4. Primera línea de un rastreo (Masanès, 2006) 27
Figura 5. Archivado con Servidor Web (Ospina M. , 2014) 30
Figura 6. Arquitectura de Heritrix (Heritrix, 2014) 33
Figura 7. Job creado exitosamente(García & Rivero, 2013) 37
Figura 8. Semilla en el archivo de configuración del job(García & Rivero, 2013) 37
Figura 9. Solicitud de rastreo almacenada en el crontab (García & Rivero, 2013) 38
Figura 10. Rastreo colocado en ejecución (García & Rivero, 2013) 38
Figura 11.Rastreos finalizados (García & Rivero, 2013) 38
Figura 12. Rastreos por indexar (García & Rivero, 2013) 39
Figura 13. Configuración de la plataforma de desarrollo (Ospina M., 2014) 40
Figura 14. Arquitectura del Prototipo de Archivo Web usando un Diagrama de componentes (García & Rivero, 2013) 41
Figura 15. Interfaz de autenticación de la Aplicación de Adquisición (García & Rivero, 2013) 42
Figura 16. Listado de rastreos del usuario autenticado (García & Rivero, 2013) 43
Figura 17. Formulario para la creación de nuevos rastreos (García & Rivero, 2013) 44
Figura 18. Página principal de la aplicación (Mantura & Martínez, 2014) 45
Figura 19. Versiones del sitio web www.usb.ve (Mantura & Martínez, 2014) 45
Figura 20. Ciclo de vida de Ágil UP, Traducida de (The Agile Unified Process, 2006) 52
Figura 21. Componente de software (Ian, 2006) 56
Figura 22. Arquitectura del Prototipo de Archivo Web de Venezuela a implementar 63
Figura 23. Prototipo de Interfaz Iniciar Sesión 65
Figura 24. Prototipo de Interfaz de Solicitud de Registro 65
Figura 25. Prototipo de Interfaz de Gestión de Registro de Usuarios 66
Figura 26. Prototipo de Interfaz Visualizar Rastreos Fallidos 66
Figura 27. Prototipo de Interfaz Opción Contáctanos 67
Figura 28. Prototipo de Interfaz Consultar Rastreos 67
Figura 29. Prototipo de Interfaz Crear Rastreos 68
Figura 30. Diagrama de Componentes del Prototipo de Archivo Web 69
Figura 31. Diagrama de actividades Reiniciar Rastreo Fallido 70
Figura 32. Diagrama de casos de uso módulo de adquisición 71
Figura 33. Diagrama de Secuencia Solicitar Registro 72
Figura 34. Diagrama de Secuencia Contactar al Administrador 73
Figura 35. Diagrama de casos de uso módulo de gestión y control de incidencias 74
Figura 36. Diagrama de Secuencia Iniciar Sesión Administrador 75
Figura 37. Diagrama de Secuencia Cambiar Contraseña Administrador 76
Figura 38. Diagrama de Secuencia Visualizar Solicitudes de Usuarios 77
Figura 39. Diagrama de Secuencia Autorizar Usuarios 78
Figura 40. Diagrama de Secuencia Denegar Acceso Usuarios 79
Figura 41. Diagrama de Secuencia Crear Solicitud de Rastreo Administrador 81
Figura 42. Diagrama de Secuencia Consultar Rastreos Administrador 82
Figura 43. Diagrama de Secuencia Editar Solicitud de Rastreo Administrador 84
Figura 44. Diagrama de Secuencia Desactivar Solicitud de Rastreo Administrador 85
Figura 45. Diagrama de Secuencia Consultar Versiones de Solicitud de Rastreo Administrador 86
Figura 46. Diagrama de Secuencia Generar Rastreo 87

Figura 47. Diagrama de Secuencia Verificar Fin de Rastreo 88
Figura 48. Diagrama de Secuencia Visualizar Rastreos Fallidos 89
Figura 49. Diagrama de Secuencia Reiniciar Rastreo 90
Figura 50. Diagrama de Secuencia Negar acceso a URL Ya Registrada 91
Figura 51. Diagrama de Secuencia Crear Colección 92
Figura 52. Diagrama de Secuencia Deshabilitar Usuarios 93
Figura 53. Modelo relacional de la base de datos utilizada en la aplicación 94
Figura 54. Formulario de Autenticación de Usuarios 98
Figura 55. Formulario de Registro de Usuarios 98
Figura 56. Opción de Solicitudes de Usuarios 99
Figura 57. Formulario para Creación de Rastreos 99
Figura 58. Formulario para la Edición de Rastreos 100
Figura 59. Archivo crawler-beans.cxml Configuración URL a Rastrear 101
Figura 60. Archivo crawler-beans.cxml Configuración Archivos Warc 101
Figura 61. Archivo crawler-beans.cxml Configuración Rastreos en Paralelo 101
Figura 62. Archivo crawler-beans.cxml Configuración Rechazo de URL con Extensiones Definidas 102
Figura 63. Agregar Solicitud del Rastreo al Crontab 102
Figura 64. Opción Desactivar Rastreos 103
Figura 65. Opción Visualizar Versiones de un Rastreo 103
Figura 66. Generar Rastreo 104
Figura 67. Invocación de script para Inserción en Base de Datos 104
Figura 68. Script insert_database.sh 104
Figura 69. Ejemplo de Archivo rastreos_activos.txt 105
Figura 70.Verificados de Rastreos Finalizados 105
Figura 71. Opción Reiniciar Rastreos Fallidos 106
Figura 72. Script reiniciar_rastreo.sh 106
Figura 73. Formulario de la Opción Contáctanos 107
Figura 74.Heritrix permisos 129
Figura 75. Comando para ejecutar Heritrix 129
Figura 76. Crear Job 130
Figura 77. Construir Job 130
Figura 78. Lanzar Job 130
Figura 79. Despausar job 131

Índice de Tablas

Tabla 1. Comparativa entre iniciativas de Archivado Web (Mantura & Martínez, 2014) 24
Tabla 2. Comparativa entre *Python y Ruby* (Chris Oliver, 2015) 49
Tabla 3. Comparación entre las Metodologías tradicionales y Ágiles 51
Tabla 4. Caso de Uso Solicitar Registro 71
Tabla 5. Caso de Uso Contactar al Administrador (García & Rivero, 2013) 72
Tabla 6. Caso de Uso Iniciar Sesión 75
Tabla 7. Caso de Uso Cambiar Contraseña 76
Tabla 8. Caso de Uso Visualizar Solicitudes de Usuarios 77
Tabla 9. Caso de Uso Autorizar Usuarios 78
Tabla 10. Caso de Uso Denegar Acceso a Usuarios 79
Tabla 11. Caso de Uso Crear Solicitud de Rastreo 80
Tabla 12. Caso de Uso Consultar Rastreos 81
Tabla 13. Caso de Uso Visualizar Lista de Rastreos 82
Tabla 14. Caso de Uso Editar Solicitud de Rastreo 83
Tabla 15. Caso de Uso Desactivar Solicitud de Rastreo 84
Tabla 16. Caso de Uso Consultar Versiones de Solicitud de Rastreo 85
Tabla 17. Caso de Uso Generar Rastreo 86
Tabla 18. Caso de Uso Verificar Fin de Rastreo 87
Tabla 19. Caso de Uso Visualizar Rastreos Fallidos 88
Tabla 20. Caso de Uso Reiniciar Rastreo 89
Tabla 21. Caso de Uso Negar Acceso a URL ya registradas 91
Tabla 22. Caso de Uso Crear Colección 92
Tabla 23. Caso de Uso Deshabilitar Usuarios 93
Tabla 24. Componente Gestionar Solicitudes de Usuarios 95
Tabla 25. Componente Generador de Rastreos 96
Tabla 26. Componente Reiniciar Rastreo 97
Tabla 27. Caso de Prueba 001 Registrar Nuevo Usuario 108
Tabla 28. Caso de Prueba Validar Datos Ingresados 109
Tabla 29. Caso de Prueba Validar Autorizar Solicitud de Registro 110
Tabla 30. Caso de Prueba Habilitar/Deshabilitar Acceso a Suscriptores 111
Tabla 31. Caso de Prueba Validar Datos de Entrada Solicitud de Rastreo 112
Tabla 32. Caso de Prueba Validar Opción Contactar al Administrador 114
Tabla 33. Caso de Prueba Validar Opción Editar Perfil de Suscriptor 116
Tabla 34. Caso de Prueba Validar Modificar Contraseña 117
Tabla 35. Caso de Prueba Creación de Solicitud de Rastreo 118
Tabla 36. Caso de Prueba Validar Creación de Colección 118
Tabla 37. Caso de Prueba Desactivar - Activar Rastreos 119
Tabla 38. Caso de Prueba Validar Opción Reiniciar Rastreo 121

Introducción

En los últimos años, a nivel mundial, se han implementado iniciativas para preservar la web, utilizando técnicas y métodos que permitan almacenar la información de manera que pueda ser accedida posteriormente a través de un navegador web. Los sistemas encargados de la Preservación Web se denominan Archivos Web, siendo este el tema de investigación a desarrollar, temática importante pero poco abordada a nivel nacional.

Preservar la web implica entre otras cosas, seleccionar un conjunto de contenidos web (páginas web) y adquirirlos de manera periódica. Lo primero se hace debido a que por su gran tamaño y constante crecimiento es inviable preservar toda la web, y lo segundo, se debe a que los contenidos web son de naturaleza dinámica por lo que tienden a cambiar con el tiempo.

Por ser un sistema que debe funcionar de manera automática es importante realizar un monitoreo de las tareas de adquisición llevadas a cabo por él, para saber si son llevadas a cabo con éxito o si es preciso alguna recuperación.

En la actualidad, se está desarrollando un Prototipo de Archivo Web en la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela, el cual cuenta con un Módulo para la selección de contenidos, que genera trabajos de rastreo y copia de los contenidos seleccionados de manera automática, basado en una frecuencia establecida. Sin embargo, no se cuenta con un Módulo de Gestión y Control de Incidencias, que permita el registro de usuarios y seguimiento del progreso de los problemas encontrados. Una incidencia puede ser cualquier evento que no se encuentre, o pueda causar fallas en el desarrollo o uso de un sistema informático.

El presente T.E.G. realiza el desarrollo e implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web para la preservación de sitios Web de Venezuela, desarrollado como fue mencionado anteriormente en la Universidad Central de Venezuela, basado en estándares y el uso de tecnologías de software libre.

Dicho trabajo está estructurado de la siguiente manera:

Capítulo I - Planteamiento del Problema: Contiene el planteamiento del problema donde se describe la problemática general asociada a la preservación Web que esta investigación identificó para dar solución. Además se justifica y se delimita el alcance de este T.E.G.

Capítulo II - Marco Conceptual: donde se realiza la explicación de los distintos conceptos que permiten comprender la investigación.

Capítulo III- Marco Metodológico: Sección donde se describe la metodología de desarrollo utilizada para el desarrollo e implementación del Módulo propuesto.

Capítulo IV Desarrollo de la Aplicación: Aquí se detallan los componentes y las actividades ejecutadas, técnicas e instrumentos concernientes al desarrollo de la aplicación.

Seguidamente se dan las conclusiones y recomendaciones consideradas por el equipo de esa investigación a tomar en cuenta para futuras extensiones de este proyecto o futuras investigaciones.

Por último se listan las referencias bibliográficas consultadas para la investigación, desarrollo e implementación del presente T.E.G.

Capítulo I Planteamiento del Problema

Para la sociedad actual, la información digital es de gran importancia e interés, información que puede ser necesitada tanto en organizaciones gubernamentales, como en empresas privadas, en las instituciones educativas e incluso puede ser requerida por particulares no relacionados con trabajos o estudios. Es allí donde entra a jugar un papel principal la preservación que se le otorgue a los contenidos generados en formato digital. Dentro del mundo digital existen los contenidos cuyo formato permite su consulta y navegación a través de Internet, como son las páginas web y los documentos que la componen. La Preservación Web, a la par de la Preservación Digital, son conjuntos de técnicas y de metodologías que permiten que la información web pueda permanecer y seguir usándose a mediano y largo plazo.

El proceso de Preservación Web se realiza a través de Archivos Web, los cuales representan el tema de estudio de esta investigación. Los Archivos Web son sistemas de información que realizan las tareas de preservación: Selección de contenidos, adquisición constante, almacenamiento y acceso; con la finalidad de que la información pueda ser empleada para futuras investigaciones, trabajos o solicitudes del público en general. Sin embargo, para que estas tareas sean exitosas, se requiere de una tarea transversal que permita hacer un monitoreo del estado del sistema y permita el manejo de incidencias.

Actualmente se está desarrollando un Prototipo de Archivo Web para la preservación de sitios web de Venezuela, bajo la figura de un proyecto de investigación financiado parcialmente por el Consejo de Desarrollo Científico y Humanístico (CDCH) (con el código PI-03-8139-2011), del cual se han desarrollado los Módulos de Selección y Acceso, Adquisición y Almacenamiento (Ospina M. , Martinez, Kabchi, & León, 2014) bajo la Tesis de Maestría (Ospina M. , 2014) y los T.E.G. (García & Rivero, 2013) (Mantura & Martínez, 2014), realizados en la Escuela de Computación de la Facultad de Ciencias de la Universidad Central de Venezuela.

La problemática en cuestión es la carencia en el Módulo de Adquisición y Almacenamiento del Prototipo de Archivo Web para:

1. Supervisar el desempeño de las tareas de adquisición de los contenidos a preservar y reiniciar los trabajos en caso de fallas.
2. Administrar los suscriptores del aplicativo.

La problemática anteriormente presentada se debe a las funcionalidades no implementadas:

1. No se visualiza la diferencia entre los perfiles de usuario y el rol del administrador.
2. No se realiza el reinicio automático de los rastreos por fallas en el sistema (inconvenientes con el servicio eléctrico).
3. No generan mensajes de interacción con el usuario.
4. No se realiza la validación de páginas registradas con anterioridad.

Además de lo anterior el Modulo de Adquisición y Almacenamiento fue el primero en implementarse por lo que aún no se habían definido lineamientos para una interfaz institucional lo que se considera un requerimiento actual importante.

Por lo antes expuesto, nuestro equipo trabajó durante el desarrollo de este T.E.G. en la solución de estas fallas en aras de contribuir con el adecuado proceso de preservación de Archivos Web en Venezuela.

1.1 Objetivo General

Implementar el Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web en Venezuela.

1.2 Objetivos Específicos

1. Adaptar el método de desarrollo AUP (*Agile Unified Process*) en una arquitectura basada en componentes, para el desarrollo del Módulo de Gestión y Control de Incidencias.
2. Modificar la interfaz del Módulo de Adquisición y Almacenamiento, de manera que se proporcione un diseño institucional para el Prototipo de Archivo Web de Venezuela.
3. Establecer el perfil del Administrador para el Prototipo de Archivo Web de Venezuela.
4. Desarrollar el Módulo de Gestión y Control de Incidencias para el Prototipo de Archivo Web de Venezuela, el cual debe:
    a. Gestionar las solicitudes de registros de nuevos usuarios.
    b. Monitorear las actividades de los usuarios autorizados y permitir que sean deshabilitados los trabajos de rastreos.
    c. Permitir la comunicación de los usuarios registrados con el usuario administrador.
    d. Monitorear los trabajos de rastreo y realizar el reinicio de los rastreos por fallas presentadas en el sistema (inconvenientes con el servicio eléctrico).
5. Realizar pruebas funcionales del Módulo de Gestión y Control de Incidencias desarrollado.

1.3 Justificación

Como fue mencionado anteriormente, la preservación de la información a través de Archivos Web es fundamental en los tiempos actuales. Con la evolución de las tecnologías de la información, cada vez existe más información que se considera importante que sea preservada, manteniendo su integridad, autenticidad, fiabilidad, legibilidad y funcionalidades. La necesidad de preservar la información de la web, entre otras cosas, deriva del hecho de que sus contenidos son efímeros.

A pesar que en la actualidad existe un gran número de iniciativas de Preservación Web, todavía existe una gran cantidad de información web que se pierde. En el caso de Venezuela, ubicándonos en el desarrollo del Prototipo de Preservación Web planteado en el Proyecto de Investigación de la Universidad Central de Venezuela, el mismo está enfocado específicamente en la salvaguarda de sitios Web de Venezuela, de páginas que contienen aspectos históricos, culturales, sociales, entre otros.

En este sentido, con anterioridad al presente Trabajo Especial de Grado, se han desarrollado los Módulos de Adquisición y el Almacenamiento (García & Rivero, 2013) y el Módulo de Acceso los Contenidos Preservados (Mantura & Martínez, 2014).

Sin embargo, si no se corrigen las fallas o funcionalidades no implementadas en el Módulo de Adquisición y Almacenamiento, esto no permitirá una efectiva administración del Prototipo de Archivo Web de Venezuela. Consideramos que este trabajo de investigación se justifica en tanto el Módulo de Gestión y Control de Incidencias permita:

- Autorización o rechazo de nuevo registro de usuario.
- Habilite la comunicación entre los usuarios autorizados y el rol de administrador mediante el envío de correos con información que sea de interés.
- Al usuario con rol administrador la supervisión y monitoreo de rastreos (al momento de presentarse una falla en el sistema, será habilitada la opción de reinicio).
- Agregar nuevas opciones de colecciones para almacenar los rastreos.

1.4 Alcance

Este T.E.G. propone la creación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web de Venezuela, y realizar la modificación de la interfaz de los Módulos de Adquisición y Almacenamiento, los cuales fueron desarrollados con anterioridad.

Dentro del alcance de este T.E.G. se contempla proveer el reinicio de los rastreos, al momento de ocurrir una falla con el servicio eléctrico, para que no se vean afectados por la misma una vez corregido el problema presentado.

Será implementado el mensaje de alerta al momento de realizar el registro de una página web ya existente, así como también la configuración de los valores de búsqueda.

Capítulo II Marco Conceptual

En este capítulo se describen los conceptos y bases teóricas del tema que permiten entender el trabajo realizado.

Se inicia el capítulo con una breve referencia de los procesos de Preservación Digital y Preservación Web, para luego enfocarnos en los Archivos Web como herramientas de tales procesos, tanto a lo referente a su definición como antecedentes de módulos o funciones de administración en los mismos.

Seguidamente ya ubicados en el contexto venezolano se describe el estado actual de avance de Prototipo de Archivo Web de Venezuela, el cual constituye el marco referencial de esta investigación. En este sentido se presentan aspectos como los procesos de adquisición, almacenamiento, indexación y los motores de búsqueda, la herramienta de rastreo, acceso, tecnologías usadas, la arquitectura de la aplicación, las interfaces y el lenguaje de programación.

2.1 La Preservación Digital y La Preservación Web

Internet constituye una fuente de recursos de información y conocimientos compartidos, a escala mundial. Es también la vía de comunicación que permite establecer la cooperación y colaboración, entre gran número de comunidades y grupos de interés por temas específicos, distribuidos por todo el planeta.

En la búsqueda de mantener y preservar la información presente en Internet, la preservación digital juega un papel importante. La preservación digital consiste en el almacenamiento de recursos que permiten la consulta posterior de los mismos, la cual puede ser de índole académica, profesional, geopolítica, económica, social o para investigación personal.

> *"La preservación digital puede definirse como el conjunto de los procesos destinados a garantizar la continuidad de los elementos del patrimonio digital durante todo el tiempo que se consideren necesarios.*
>
> *El patrimonio digital está formado por los materiales 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 materiales digitales poseen valor perdurable, pero los que lo tienen exigen metodologías de conservación activas para mantener la continuidad del patrimonio digital."* (UNESCO, 2003).

El objetivo de la preservación digital es mantener su accesibilidad, es decir, mantener en el tiempo la colección de información de interés que pueda ser utilizada por las generaciones venideras.

El patrimonio digital se puede clasificar de la siguiente manera, sin ser exhaustivos y sin ningún orden en específico:

- Publicaciones electrónicas, algunos formatos de publicación tradicional, como monografías y fascículos, y otros en Sitios Web y revistas electrónicas.
- Documentos semi publicados; por ejemplo, artículos anticipados y tesis conservadas en formato informático, y otros archivos de uso restringido dentro de comunidades específicas como universidades y sociedades académicas.
- Los registros de actividades, transacciones, correspondencia, etc., tanto personales como de diversos organismos, grupos de discusión, blogs, etc.
- Los materiales educativos utilizados por los sistemas de enseñanza asistida por computadora.
- Las herramientas informáticas, como las bases de datos, los modelos, las simulaciones y los programas informáticos.
- Los productos de entretenimiento, de carácter comercial o no, creados por las empresas cinematográficas, de música, de juegos o de radio y televisión.
- Las obras gráficas y las fotografías documentales realizadas por medios digitales.
- Las copias digitales de imágenes, sonidos, textos y objetos tridimensionales efectuadas a partir de originales no digitales.

Muchos de estos artefactos solo existen en formato digital, por lo que es de suma relevancia e importancia mantener en el tiempo los mismos, para que sean adquiridos y consultados por las generaciones futuras.

Los problemas asociados a la preservación del patrimonio digital no son únicamente de carácter técnico, sino que también tienen dimensiones sociales y de organización, es por ello que surge el interés de la Organización de las Naciones Unidas para la Educación (UNESCO), en este proceso, al ser parte de su razón de ser fomentar y permitir la preservación y el disfrute del patrimonio cultural, científico e informativo de los pueblos del mundo. Por lo que difícilmente podía ignorar el crecimiento y la vulnerabilidad del patrimonio digital (Organización de las Naciones Unidas para la Educación, 2003).

Según la UNESCO, la preservación del patrimonio digital debería cumplir las siguientes características:

1. Ser accesible a todo público.
2. Garantizar la protección de información delicada o de carácter privado
3. Disponer de un marco jurídico y técnico que proteja su autenticidad
4. No debe estar sujeto a límites temporales, geográficos, culturales o de formato. Se debe propiciar, con el tiempo, una representación de todos los pueblos, naciones, culturas e idiomas.

A su vez, dentro del ámbito de la Preservación Digital, nos encontramos con un tipo especial de recurso que tienen un carácter dinámico en el tiempo, como son los recursos en formato web; que a diferencia de otros recursos como es el caso de libros, revistas, publicaciones, imágenes o sonidos, tienen características propias de cambiar de manera constante y tener hiperenlaces que hace que su lectura no sea secuencial. Estos recursos componen lo que se define como Preservación Web.

La Preservación Web, como parte de la Preservación Digital, tiene como objetivo conservar conjuntos seleccionados de páginas de Internet las cuales deben ser mantenidas completas, es decir, acompañadas de los formatos de archivos, imágenes, gráficas, y aspecto visual, y deben ser almacenadas en un ambiente seguro, para luego hacerlas accesibles a los usuarios de Internet (Ospina M. , 2014).

Dado que ya se conoce la importancia de la Preservación Digital y en especial de la Preservación Web, muchos países se han esforzado para llevar a cabo las iniciativas para dicha preservación, haciendo que algunas de éstas planifiquen ampliar un poco más su dominio o alcance. En la Figura 1 se evidencia cuales países del mundo han desarrollado iniciativas de preservación web.

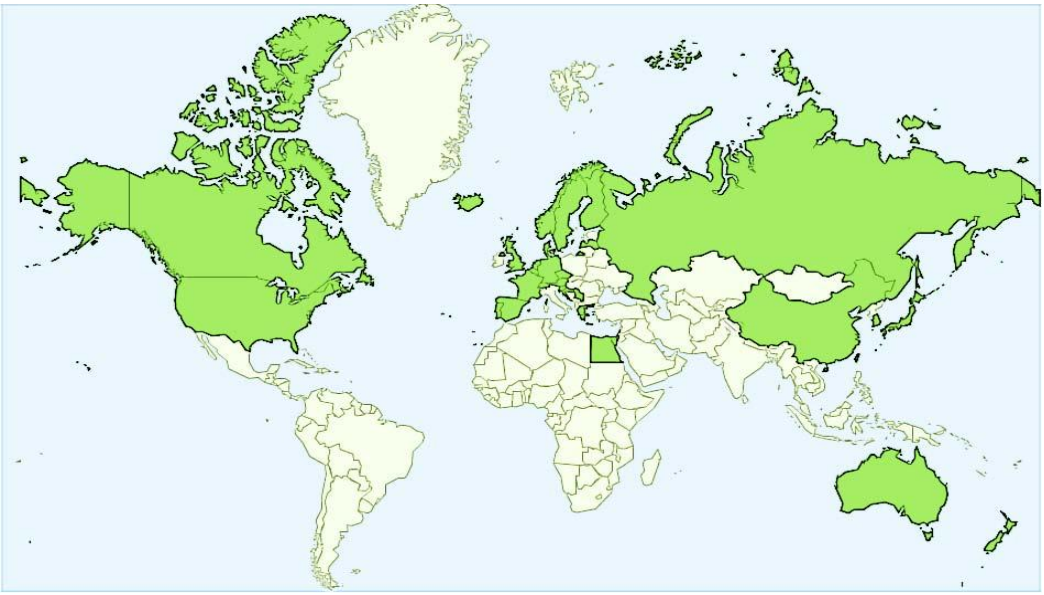


Figura 1. Mapa de países con Iniciativas de Archivo Web en el mundo (IIPC, 2014)

2.2 Archivos Web

Para llevar a cabo el proceso de Preservación Web, se han venido desarrollando sistemas de información cuyo objetivo es almacenar de manera histórica contenidos de la Web, mediante la replicación y/o migración de su formato original a otra representación en ambientes seguros, a los cuales se les denomina Archivos Web o Iniciativas de Preservación Web.

Los Archivos Web son Sistemas de Información que surgen para archivar de manera histórica documentos que están publicados en la Web, considerados parte del patrimonio digital de las naciones. Se puede definir un documento web como un documento basado en el lenguaje, de marcas HTML (*HiperText Markup* Lenguaje) también llamado página web, y los demás archivos asociados en su composición como imágenes, videos, hojas de estilo, scripts, entre otros, que pueden ser localizado a través de un URL y que normalmente forman parte de un sitio web (Ospina M. , 2014).

Los Archivos Web tienen como objetivo preservar conjuntos seleccionados de páginas o sitios web y sus documentos, mediante su replicación y/o migración de su formato original a otra representación. Los sitios replicados son mantenidos completos, es decir, acompañados de los archivos, imágenes, gráficas, y aspecto visual, y son almacenados en servidores de preservación en un ambiente seguro (Masanès, 2006).

El objetivo delArchivo Web es conservar la forma original del contenido cosechado sin modificaciones. Para lograr este objetivo, deben existir instrumentos, normas, políticas y buenas que aseguren la Gestión de Archivos Web a través del tiempo (IIPC, 2012).

Los Archivos Web deben llevar a cabo un conjunto de tareas que hagan factible la preservación de la web, las cuales se presentan en la Figura 2.

Figura 2. Actividades para la Preservación Web (Ospina, Martinez, Leon, & Kabchi, 2014)

A continuación se presenta una breve descripción de estas tareas: (Ospina, Martinez, Leon, & Kabchi, 2014)

- La selección permite limitar el ámbito del Archivo pudiendo preservar contenidos locales, o de un tipo en particular, por ejemplo contenidos de un país, o educativos solamente.
- La adquisición logra que se puedan almacenar los cambios que se generan sobre los contenidos que se preservan a través del tiempo.
- El almacenamiento requiere estrategias que permitan preservar grandes volúmenes de: información (del orden de los *Terabytes*), millones de archivos y diferentes formatos. Para este fin se han desarrollado formatos de archivos contenedores específicos, cuyo objetivo principal es superar la limitación de los sistemas de archivos propios de los sistemas operativos donde se alojan los Archivos Web, siendo los más usados el formato ARC, desarrollado por *Internet Archive*, y el formato WARC un estándar reconocido desarrollado por el IIPC para la preservación web como una mejora del formato ARC.
- El acceso o recuperación de los contenidos está estrechamente ligado a la forma en que se encuentran almacenados, pero debido a la naturaleza hipertextual y multimedia de la web, se espera que el usuario final pueda acceder a este contenido de manera similar a cuando lo hace en los servidores originales.

2.3 Antecedentes de Módulos o Funciones de Administración en Archivos Web

En el siguiente apartado se expondrán dos Archivos Web existentes a nivel mundial.

2.3.1 Internet Archive

Es una organización sin fines de lucro que fue fundada en 1996 con el propósito de construir una biblioteca de Internet. Entre sus objetivos está el proporcionar acceso permanente a las colecciones históricas que existen en formato digital dirigido a investigadores, historiadores, académicos, personas con discapacidad y al público en general. Esta organización opera en los Estados Unidos.

Desde el año 2006 ofrece un servicio de suscripción llamado *Archive-it*, que permite construir colecciones personalizadas de contenido digital.

Una de las características relevantes de este servicio, es que no es necesario poseer conocimientos técnicos ni sistemas de alojamiento, ya que cuenta con una aplicación que se encarga de hacer el trabajo. Otro punto importante, es que los suscriptores pueden cosechar, catalogar y archivar sus colecciones así como hacer búsquedas y navegar por las colecciones, una vez se haya terminado la construcción de las mismas. Las colecciones están alojadas en el centro de datos del Internet Archive y son de acceso público.

2.3.2 PANDORA (*Australia's Web Archive*)

Es una colección de publicaciones nacional en línea de Australia, la cual fue establecida inicialmente por la Biblioteca Nacional de Australia en el año 1996, y actualmente cuenta con la colaboración de nuevas bibliotecas australianas y organizaciones culturales. Las publicaciones recogidas documentan la vida cultural, social, política y actividades de la comunidad australiana, como actividades intelectuales y actividades artísticas (PANDORA, 1996).

Pandora recopila, preserva, y pone a disposición, para su visualización, los recursos Web australianos. El acceso al archivo está a disposición del público, a través del sitio Web de Pandora. Los sitios Web son seleccionados en base a su importancia cultural y el valor de la investigación.

A continuación se realiza una comparación entre los dos Archivos Web que fueron mencionados anteriormente, ilustrándose en la Tabla 1.

Tabla 1. Comparativa entre iniciativas de Archivado Web (Mantura & Martínez, 2014)

Iniciativa Características	*Internet Archive*	*PANDORA(Australia's Web Archive)*
País	Estados Unidos	Australia
Año de Creación	1996	1996
Tecnologías	Heritrix, Wayback Machine, Nuthwax	Sistema digital de archivado PANDORA (PANDAS), HTTrack, DOSS
Formato de Archivo	ARC/WARC	ARC/WARC
Contenido de Archivado(millones de archivos)	150.000	3.100
Métodos de Acceso	Búsquedas por URL, Búsquedas por metadatos avanzada, Búsquedas de texto completo	Búsquedas por metadatos avanzada, Búsquedas de texto completo

2.3.3 Módulos de Administración

Los módulos de administración del Internet Archive y de PANDORA no se encuentran para la visualización del público, pero si se lista la información de interés relevante para la presente investigación. El módulo de administración del Internet Archive se encarga de: (Internet Archive, 2003; PANDORA, 1996)

> *"Recoger y almacenar datos sobre los visitantes del sitio durante la operación, entre estos la dirección IP, la última página visitada, y el navegador utilizado, entre otras informaciones. Estos datos de registro son útiles para la seguridad, mejorar la facilidad de uso, y con fines de marketing, Para ser eficaz, una política de registro de uso debe describir:*
>
> - *Qué datos se registra*
> - *Información que será expuesta a los visitantes*
> - *Cuando será retenido o destruido"*

Se realizan las siguientes actividades a lo largo del proceso de adquisición y visualización de información mostrada en el Internet Archive (Internet Archive, 2013):

- **Adquisición de datos recopilados**: mediante el uso de las cookies, la dirección IP del usuario, almacenamiento de la dirección del correo electrónico mediante formularios.
- **Control de gestión de acceso para publicaciones de páginas**: Los controles de acceso se aplican mediante el otorgamiento de autorizaciones para la publicación de información.
- **Retención y Eliminación**: Los datos deben pasar por aprobación para que puedan ser expuestos para los usuarios que visitan la página, de no pasar la aprobación los mismos son eliminados.

2.4 Estado del Arte: Prototipo de Archivo Web de Venezuela

En la búsqueda de realizar aportes para que Venezuela implemente la Preservación Web, se han desarrollado proyectos para el desarrollo de un Prototipo de Archivo Web. En esta sección se expone la información con la que se cuenta acerca de este proyecto.

En la actualidad se cuenta con un Prototipo de Archivo Web desarrollado bajo la Tesis de Maestría (Ospina M. , Un marco de referencia para la implementación de Archivos Web. Trabajo de Grado de Maestría, 2014) y los T.E.G. realizados por Mantura Kabchi y Miguel Martínez (Mantura & Martínez, 2014), y Lorena Rivero y Jonathan García (García & Rivero, 2013). En los mismos se realizan la implementación del Módulo de Acceso, y el Módulo de Adquisición y Almacenamiento para el Prototipo de Archivo Web de Venezuela.

2.4.1 Adquisición

El término adquisición, designa los diferentes medios técnicos utilizados para obtener el contenido a ser preservado. La captura del contenido se puede hacer tanto en línea como fuera de línea (se obtienen los archivos directamente del servidor), muchas veces se necesita más de un enfoque para obtener el contenido, es por ello que existen varios métodos de adquisición.

En el Prototipo de Archivo Web desarrollado fue utilizado el archivado del lado del cliente (García & Rivero, 2013).

Este es el principal método de adquisición tanto, por su simplicidad, como por su escalabilidad y adaptación a un entorno cliente-servidor, como se muestra en la Figura 3.

Los rastreadores se adaptan a la forma habitual de acceder a la Web (peticiones http). Esto permite el archivado de cualquier sitio que sea accesible libremente ya sea en la

Web tradicional en intranet o extranet, es necesario que el rastreador obtenga la autorización correspondiente.

Este método no sólo adopta la misma posición de usuarios de la Web, sino que también imita su forma de interacción con los servidores. Los rastreadores comienzan desde las páginas semillas, las analizan, extraen los enlaces y buscan los documentos enlazados. A continuación, reiteran este proceso con los documentos traídos y sigue el procedimiento siempre que haya vínculos a explorar, entonces encuentran más documentos dentro del alcance definido. Cada página tiene por lo tanto que ser "descubierta" por extracción de enlaces desde otras páginas.

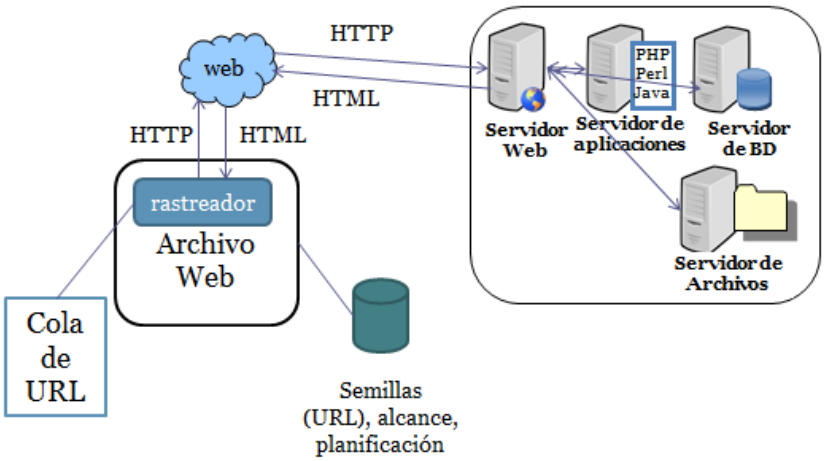


Figura 3. Archivado del lado del cliente (Ospina M. , 2014)

La tecnología de rastreo ha sido desarrollada originalmente para fines de indexación. Su aplicación para el archivado Web, a pesar de que reutilizan la mayoría de sus aspectos, implica varios cambios. El primero, es que los rastreadores para archivado trataran de buscar todos los archivos independientemente de su formato, para archivar una versión completa de los sitios; al contrario de los rastreadores de motores de búsqueda, que suelen buscar sólo los archivos que pueden indexar.

La segunda diferencia, tiene que ver con la gestión temporal de los rastreos. Para evitar la sobrecarga de los servidores Web, los rastreadores respetan normas de cortesía (se establece un retardo fijo entre dos peticiones, por lo general de varios segundos, o un retardo que depende del tiempo de respuesta del servidor). Esto implica que una captura Web puede durar varios minutos en el mejor caso, varias horas o incluso días. Este retraso

plantea un problema de la consistencia temporal de la captura del sitio, que puede sufrir cambios durante el tiempo que está siendo capturado.

Esto afecta a los rastreadores de archivado, porque el rastreo debe proporcionar el contenido y no sólo la dirección del mismo. Los documentos web deben ser rastreados en su conjunto, con o sin su coherencia interna, siendo su única condición para la navegación y la interpretación.

Esto tiene consecuencias de largo alcance en lo que respecta a la política de rastreo. Como la cortesía con los servidores siempre ha sido un cuello de botella para el rastreo, los rastreadores han estado utilizando prioridad de rastreo principalmente en amplitud *(breadth-first),* con algunas variantes principalmente destinadas a rastrear las "mejores" primeras páginas, pero esta estrategia de planificación de rastreo tiene el inconveniente de aumentar la discrepancia temporal de los rastreadores a nivel de sitio (Masanès, 2006).

Dado el retardo entre las solicitudes y los recursos disponibles para rastrear, se tiene que encontrar el número óptimo de sitios para iniciar y al mismo tiempo asegurar que la frecuencia de solicitud será establecida por las normas de cortesía, sin retardos innecesarios entre las solicitudes. La Figura 4 muestra la línea frontal" *(front-line)* de un rastreo, cuyo tamaño corresponde a la asignación óptima de los recursos de rastreo.

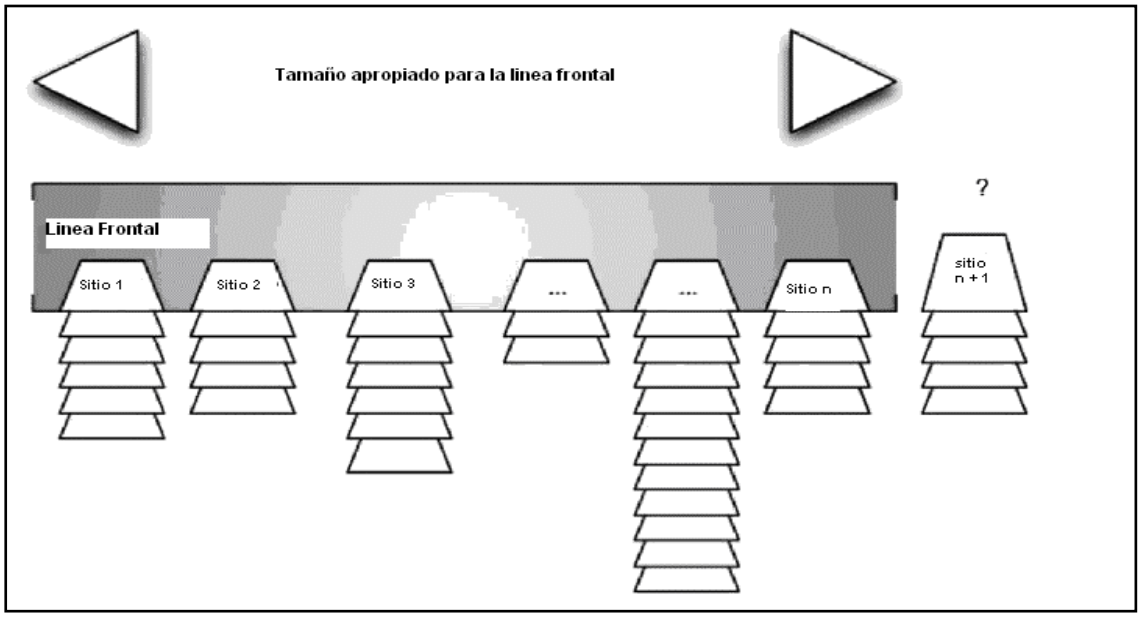


Figura 4. Primera línea de un rastreo (Masanès, 2006)

2.4.2 Almacenamiento

Realizar una copia de un sitio Web es una tarea compleja, ya que un sitio Web puede ser tan simple como un único archivo o una de las colecciones más complejas de artefactos de software que cooperan entre sí.

Lo ideal al momento de realizar la copia sería que el archivo fuera similar al original (misma estructura jerárquica, nombres de archivos, mecanismos de enlaces, formato), pero por razones prácticas, esto casi nunca sucede. Esta es la razón por la cual los responsables de los Archivos Web adoptan estrategias de transformación. Estas transformaciones pueden afectar el direccionamiento, los mecanismos de enlazado y los formatos, así como la representación del objeto en sí.

La estrategia adoptada para el Prototipo de Archivo Web de Venezuela desarrollado fue la de Archivado con Servidor Web (García & Rivero, 2013).

El Archivado con Servidor Web evita las limitaciones en cuanto al tamaño del Sistema de Archivo, que es crucial para la Preservación Web a gran escala. Se basa en el archivado de la respuesta, las respuestas del servidor original se almacenan sin cambios en contenedores de archivos WARC (*Web ARChive* – Archivo Web), que permite servirlos más tarde a los usuarios del archivo con un servidor HTTP.

2.4.2.1 Formato WARC

Este formato permite concatenar y almacenar múltiples recursos u objetos de datos que contienen cabeceras de textos y bloques de datos en un solo archivo de gran tamaño. Es una extensión del formato ARC que permitía capturar imágenes de la Web. Cada archivo contiene uno o más objetos digitales y una vez que el objeto es escrito el archivo se cierra y solo se abre para lecturas. Se espera que este formato se convierta en un estándar. Mientras que el formato ARC guarda la información primaria, el formato WARC puede almacenar información como metadatos, abreviar eventos duplicados, transformaciones de fechas y la segmentación de grandes recursos. Puede controlar la información de protocolos de la capa de aplicación como HTTP, DNS (*Domain Name System* - Sistema de Nombres de Dominio) y FTP (*File Transfer Protocol* - Protocolo de Transferencia de Archivos), comprimir datos manteniendo su integridad (ISO, 2009).

Un archivo de formato WARC es una concatenación de una o más colecciones WARC. Por lo general, la primera colección describe las que le siguen. El contenido de una colección es usualmente el resultado de la adquisición de páginas Web, imágenes, información de redireccionamiento URL, resultados de búsqueda de nombres a través del protocolo DNS,

archivos únicos o material sintetizado (metadatos, contenido transformado). Una colección WARC consiste de una cabecera seguida de un bloque de contenido (ISO, 2009).

Toda colección en WARC tiene un tipo. Actualmente hay ocho tipos propuestos existentes, que se explican a continuación (ISO, 2009):

- **warcinfo:** una colección de este tipo describe las colecciones que le siguen hasta el final del archivo o hasta la siguiente colección del tipo warcinfo. Típicamente aparece al inicio de cada archivo WARC y usualmente contiene información acerca de la cosecha que generó las siguientes colecciones.
- **response:** una colección del tipo response debería contener una esquematización completa y específica de la respuesta del servidor incluyendo la información del protocolo de red.
- **resource:** una colección calificada como un recurso puede contener, por ejemplo, un archivo adquirido directamente de un repositorio localmente accesible o el resultado de una búsqueda en la red donde el protocolo de información ha sido descartado.
- **request:** una colección del tipo *request* debería contener una esquematización completa y específica de la petición al servidor incluyendo la información del protocolo de red.
- **metadata:** es una colección creada para describir, explicar o acompañar a otra colección. Esta colección siempre debe referirse a otra sea de contenido original o versionado. Ejemplo: warc/0.9 7141 *resource* file://webserver/htdoc/images/logo.jpg
- **revisit:** este tipo describe la acción de volver a visitar el contenido previamente archivado y puede incluir un cuerpo de contenido abreviado que tiene que estar enlazado a alguna colección previa. Esto se utiliza en lugar del tipo response o del tipo *request* para indicar que el contenido visitado era un documento completo o un duplicado de un material previamente almacenado.
- **conversión:** una colección de conversión contiene una versión alternativa del contenido de otra colección que se creó como resultado del proceso de archivado. Cada transformación de una versión alternativa debe ser independiente de su versión original.
- **continuation:** los bloques de colecciones de este tipo deben estar concatenados a su correspondiente colección anterior (de otros archivos WARC) para crear la colección original de tamaño completo. Este es utilizado

cuando una colección hace que el WARC exceda el tamaño límite deseado y debe ser separado en segmentos.

Un archivo WARC registra una secuencia de documentos web cosechados, cada página está precedida por un encabezado que describe brevemente el contenido de cosecha y su longitud.

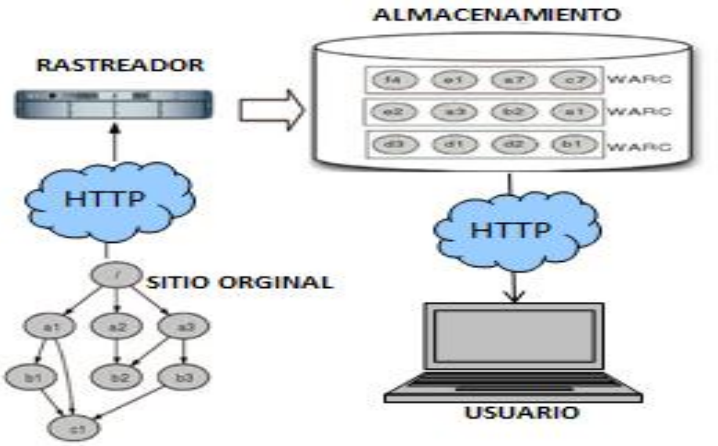


Figura 5. Archivado con Servidor Web (Ospina M. , 2014)

Como podemos ver en la Figura 5, con el Archivado con Servidor Web el sitio original se rastrea y las respuestas se almacenan sin cambios en el contenedor WARC. Lo que permite evadir el mapeo del archivo del sistema para la asignación de nombres por convenio y el cambio de la estructura del enlace. El acceso requiere que un Servidor Web obtenga lo que esté almacenado en los contenedores y los envía como una respuesta al usuario final, permitiendo que la navegación por el sitio web se pueda hacer tal cual como fue rastreado.

La principal ventaja de la utilización de contenedores WARC, es la posibilidad de superar la limitación del sistema de archivos de almacenamiento en términos de tamaño, ya que se almacenan menos archivos individuales en el sistema de archivo.

La desventaja de este enfoque se basa en que es imposible el acceso directo a los archivos almacenados. Son necesarias dos capas extra de aplicación para acceder al contenido: un sistema de índice para archivos WARC y un servidor Web. Estas dos capas no

son muy complejas, pero requieren la ejecución de un ambiente de acceso, que puede ser difícil de configurar y mantener en organizaciones pequeñas.

Este método es apropiado para la Preservación Web a mediana y gran escala, así como para los pequeños archivos que se ocupan de la preservación de la autenticidad del contenido. Como estos métodos almacenan las respuestas del servidor original, como llega del cliente, sin ningún tipo de transformación, en realidad ofrece más fidelidad que los otros métodos.

2.4.3 Indexación y Motores de Búsqueda

Para garantizar un orden y facilitar el acceso a estos contenedores se hace uso de índices. Estos disminuyen los tiempos de búsqueda y facilitan el proceso dentro del repositorio. Una indexación basada en un Archivo Web será acumulativa y dependerá de la política de recolección que se tenga, y se tendrá una referencia de todos los documentos que se han cosechado y almacenado en el archivo.

La mayoría de las herramientas de acceso a la Web están basadas en indexación automática, que no es más que la indexación que se realiza a través de procedimientos algorítmicos. Existen varios tipos de indexación automática y cuando la Web fue ganando popularidad se utilizaron tres tipos diferentes: el primero, fue un simple estilo de índice parte-de-atrás-del-libro, como en muchos libros escolares donde las palabras claves están vinculadas a una página o un capítulo. Cabe destacar que para la Web las palabras claves están vinculadas a una URL. El segundo tipo, sería indexando las palabras claves en conjunto con árboles objetos de los sitios revisados y el tercero, se realizaría con los motores de búsqueda más la indexación.

La indexación en el Archivo Web desarrollado, está basada en palabras clave con la ayuda de una plataforma de búsqueda, siendo la URL del sitio cosechado la clave de indexación. Al momento de la consulta, se coloca la URL deseada y se obtendrá como resultado todas las versiones correspondientes a la misma (García & Rivero, 2013).

2.4.4 Herramienta de Rastreo: Heritrix

Para realizar la adquisición, almacenamiento y resguardo de la información, se hace uso de las herramientas de rastreo que serán implementados para el proceso de Preservación Web. En el siguiente apartado se expone la herramienta de rastreo utilizada en el desarrollo del Prototipo de Archivo Web.

Un rastreador es un software que busca en la Web de forma metódica, y cuyo objetivo es recolectar información actualizada y descargarla en algún repositorio de datos. Son importantes, por ejemplo, para minería de datos o aplicaciones que indexan archivos.

Un rastreador mantiene una lista inicial de URLs que va a revisar. En esta lista los URLs son priorizados y mantenidos. El rastreador toma uno de los URL de la lista, descarga la página, extrae cualquier otro enlace que esta contenga y coloca los nuevos URLs en la lista. Este proceso se repite hasta que el rastreador decide cuando detenerse. La herramienta de rastreo utilizada en el desarrollo en el Prototipo de Archivo Web es el rastreador Heritrix.

Heritrix es un software que permite adentrarse en el entramado Web, es decir, recorrer las páginas Web en profundidad. Heritrix es una herramienta de software libre, extensible, escalable y de calidad de archivado del "*Internet Archive*"(Jack & Binns, 2012).

El *Internet Archive* (Internet Archive, 2013) comenzó el desarrollo de Heritrix a principios de 2003, con la intención de implementar un rastreador con el propósito específico del archivado de sitios Web. Al ser código *open-source* se fomenta la colaboración y el desarrollo con entes similares que necesiten servicio de rastreo.

Heritrix fue diseñado como un *Framework* de rastreo genérico, donde diversos componentes intercambiables pueden ser conectados. Los rastreos son configurables, para ello se eligen y configuran un conjunto de componentes específicos y se ponen en funcionamiento, la ejecución de un rastreo repite el siguiente proceso recursivamente, este proceso es común en todos los rastreadores Web:

1. Elegir un URI (*Uniform Resource Identifier* - Identificador de Recursos Uniforme) de entre todas las programadas.
2. Buscar la URI.
3. Analizar o archivar los resultados.
4. Seleccionar las URI descubiertos que sean de interés, y sumarlos a los ya programados.
5. Se termina el procesamiento de la URI actual y repetir el proceso.

Los tres componentes principales de Heritrix son:

1. **Alcance:** Determina si cierta URI esta fuera o dentro de las reglas de rastreo, el alcance incluye las semillas: URI que se usan para iniciar el rastreo, el alcance también interviene en la selección de URIs mencionadas en el paso 4 del proceso de rastreo.

2. **Frontera:** Es el responsable de seleccionar el siguiente URI a ser procesado, además de llevar un registro de las URI cosechadas y otro de las URI que ya han sido procesadas.
3. **Cadena de procesamiento**: Incluye procesadores modulares que realizan tareas específicas en cada URI, esto incluye; búsqueda de la URI, análisis de los resultados devueltos y pase de URIs descubiertas a la frontera**.**

2.4.4.1 Principales Componentes

A continuación se detallarán los componentes principales de la arquitectura de Heritrix, iniciando con la figura 6 donde se detallan todos los componentes (Heritrix, 2014).

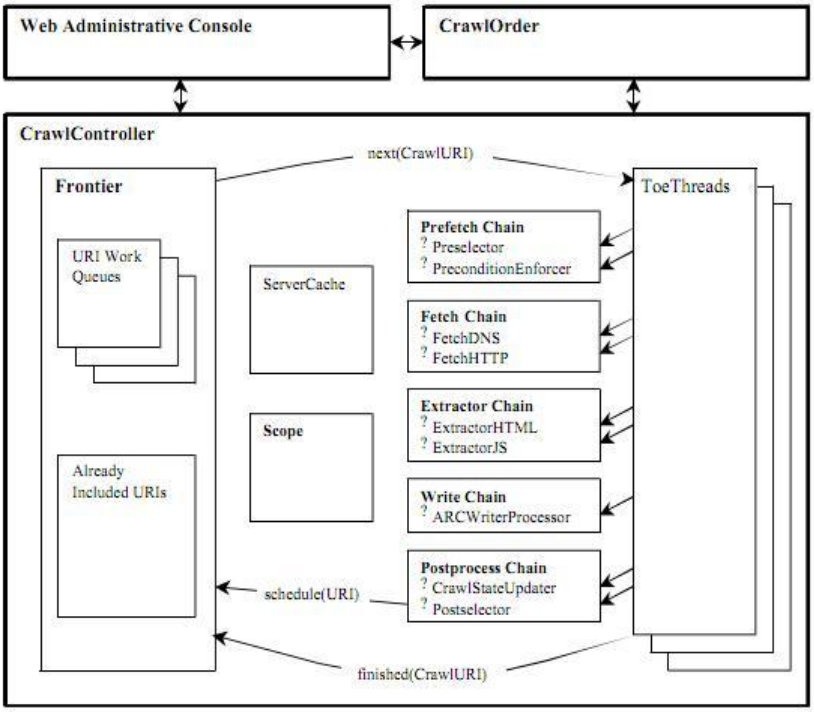


Figura 6. Arquitectura de Heritrix (Heritrix, 2014)

La ***Web Administrative Console*** puede ser vista como una aplicación independiente (*standalone*), su funcionalidad es la de permitir elegir los componentes de un rastreo y especificar los parámetros de un ***CrawlOrder*****.** Con la **Web Administrative Console** es posible ver el estado del rastreo actual, *logs* y generar reportes.

Un rastreo es iniciado cuando el ***CrawlController*** aprueba la **CrawlOrder,** el ***CrawlController*** crea una instancia de todos los módulos necesarios para el rastreo, la ***Web Administrative Console*** controla el rastreo a través del ***CrawlController*****.**

El ***CrawlOrder*** contiene suficiente información para crear el ámbito de aplicación. Datos tales como: el alcance, la semilla con la cual el ***Frontier*** iniciará el rastreo y la información de qué hacer con las URIs que sean descubiertas posteriormente.

El ***Frontier*** es el responsable de ordenar los URIs a ser visitados, así como de asegurarse que las URIs no sean revisadas innecesariamente y moderar las visitas que el rastreador hace a cualquier sitio remoto. Se consiguen estos objetivos mediante el mantenimiento de una serie de colas internas de URIs para ser visitados y una lista de todos los URIs que ya han sido visitados o encolados, por defecto el ***Frontier*** tiene una implementación de búsqueda en anchura, esto determina las políticas para seleccionar las URIs a procesar, con una opción de preferiblemente terminar el rastreo de los sitios en progreso antes de iniciar con nuevos sitios. Otras implementaciones de ***Frontier*** son posibles.

El rastreador Heritrix es multiproceso, por lo que puede llevar muchos rastreos en paralelo. Cada subproceso de trabajo se denomina ***ToeThread*** y mientras se mantiene un rastreo activo cada ***ToeThread*** recorre los pasos que corresponden al proceso genérico de rastreo descrito anteriormente.

El número de ***ToeThread*** ejecutándose en un rastreo se puede ajustar para lograr el máximo rendimiento con los recursos locales. El número de ***ToeThread*** por lo general oscila en el rango de los cientos.

Cada URI es representada por una instancia de ***CrawlURI***, se empaqueta la URI con información adicional recogida durante el procesos de rastreo, los componentes del sistema comunican su progreso y la salida a través de ***CrawlURI*,** que lleva los resultados de procesamientos anteriores, para ser revisados posteriormente y finalmente el ***CrawlURI*** regresa a la frontera para influir en futuros reintentos o programación de rastreos

El ***ServerCache*** contiene datos persistentes acerca de los servidores a través de ***CrawlURI*.** Contiene cualquier número de entidades ***CrawlServer*** recogiendo información como direcciones IP, política de exclusión de robots, capacidad de respuesta histórica y estadísticas de rastreos del host.

La funcionalidad global de un rastreador con respecto a una URI programada, es en gran parte especificada por la serie de procesadores configurados para ejecutarse. Cada procesador a su vez realiza sus tareas marcando el estado del ***CrawlURI*** y retornando. Las tareas realizadas varían en función del tipo de URI, historia o el contenido recuperado.

Los procesadores se agrupan en cinco cadenas:

- Procesadores en el ***PrefetchChain*** reciben el ***CrawlURI*** antes de resolver cualquier actividad de red o de ir a buscar la URI, Por lo general cada procesador demora, reordena o veta la tramitación posterior de una ***CrawlURI***, por ejemplo para asegurar que las políticas de exclusión de robots se captan y son consideradas antes de que la URI sea procesada.
- Procesadores en el ***FetchChain***, la actividad de red intenta adquirir el recurso que se refiere a un ***CrawlURI*,** es el caso típico de una transacción HTTP. Un procesador de búsqueda llenará la demanda y respuesta en el *buffers* de la ***CrawlURI***o indicará cualquier condición de error que impida que el buffer sea llenado.
- Procesadores en el ***ExtractChain*,** realizan seguimiento al procesamiento de un ***CrawlURI***que busca las URIs ya finalizadas, extrayendo característica de interés, por lo general estos son nuevos URIs que también pueden ser elegibles para ser visitados. En este paso las URIs solo se descubren, no se evalúan.
- Procesadores en el ***WriteChain*,** almacenan el resultado del rastreo – retornando contenido o características importantes – para almacenamiento permanente. El rastreador estándar simplemente escribe los datos en el formato de archivos de Internet, conocidos como archivos ARC. Sin embargo, terceros han creado procesadores que escriben los datos en otros formatos o indexan los resultados del rastreo.
- Por último, procesadores en el ***PostProcessChain*,** realiza el mantenimiento final de las acciones de rastreo en el ***CrawlURI*** como las pruebas descubiertas.

2.4.4.2 Configuración Básica de los Rastreos

La configuración del rastreo se realiza editando el archivo *crawler-beans.cxml*. Se puede modificar las siguientes características:

- **Límites de rastreo:** Además de los límites impuestos en el ámbito de aplicación del rastreo, es posible aplicar límites arbitrarios sobre la duración y el alcance del rastreo con los siguientes valores:

- ***maxBytesDownload***: Detener el rastreo después de un número fijo de bytes que se han descargado. Cero significa ilimitado.
- ***maxDocumentDownload***: Detener el rastreo después de la descarga de un número fijo de documentos. Cero significa ilimitado.
- ***maxTimeSeconds***: Detener el rastreo después de haber transcurrido un determinado número de segundos. Cero significa ilimitado.

- ***MaxToeThreads*:** El número máximo de subprocesos ejecutándose en paralelo. Si se ejecuta un rastreo de dominio más pequeño de 100 hosts, un valor aproximadamente el doble de la cantidad de hosts debería ser suficiente.
- ***metadata.operatorContactUrl*:** El URI del iniciador de rastreo. Esta configuración proporciona al administrador de un host rastreado un URI para referirse a en caso de problemas.

2.4.4.3 *Logs* de Rastreos

Cada rastreo tiene su propio conjunto de archivos de registro. Los registros se encuentran en el directorio "*logs*". Los *Logs* de Rastreo o Registros de Rastreos se encargan de almacenar la información pertinente de un rastreo realizado (Heritrix, 2014).

Existen diversos tipos de archivos de rastreos:

- ***alerts.log***: Este registro contiene alertas que indican problemas en el rastreo realizado.
- ***crawl.log*:** Cada URI que Heritrix intenta rastrear causará una línea de registro que deben escribirse en el archivo `crawl.log`.
- ***progreso-statistics.log*:** En intervalos configurables, una línea se ingresa en el registro detallando el progreso del rastreo.
- ***runtime-errors.log*:** Este registro captura excepciones inesperadas y los errores que se producen durante el rastreo. Algunos pueden ser debido a las limitaciones de hardware (sin memoria, a pesar de que el error se puede producir sin que se escriban en este registro), pero la mayoría son probablemente debido a los errores de software, ya sea en el núcleo de Heritrix pero más probablemente en una de sus clases.
- ***uri-errors.log*:** Este registro almacena los errores resultantes de intento de recuperación de URI. Por lo general, la causa es de URIs inexistentes. Este

registro suele ser sólo de interés para los usuarios avanzados que tratan de explicar el comportamiento de un rastreo inesperado.

2.4.4.4 Proceso de Rastreo en el Prototipo de Archivo Web

En el siguiente apartado se describe el proceso de rastreo utilizado por el aplicativo.

Los usuarios utilizan el módulo administrativo para realizar la creación o edición de un rastreo (Ver Figura 15 Apartado 2.4.8.1 referente al Módulo de Adquisición).

Cuando se quiere crear una solicitud de rastreo, es necesario que los datos tengan el formato adecuado, si esto no ocurre, se le informa al usuario que existen errores en los datos introducidos.

Cuando se hace la creación de una solicitud de rastreo, se selecciona por turno rotatorio en qué máquina debe ser creado el *job*, seguidamente se crea el *job* en la máquina seleccionada, tal y como se ilustra en la Figura 7.

Si un rastreo es creado de forma exitosa se hace una inspección sobre el archivo de configuración del *job* asociado al rastreo, de manera tal que se valide que los parámetros introducidos por el usuario se encuentren en el archivo de configuración, esto se puede apreciar en la Figura 8.

Se crea un trabajo de rastreo por cada entrada de la lista de sitios web a cosechar, y se va verificando si el archivo de configuración del *job* posee los parámetros especificados.

Nombre de archivo	Tamaño de archivo	Tipo de archivo	Última modifica...	Permisos	Propietario...
..					
ciens-2013-10-06-01-21-17		Carpeta de archi...	06/10/2013 01:...	drwxr-xr-x	root root

Figura 7. Job creado exitosamente(García & Rivero, 2013)

```
32  <!-- overrides from declared <prop> elements, more easily allowing
33      multiline values or even declared beans -->
34  <bean id="longerOverrides" class="org.springframework.beans.factory.config.PropertyOverrideConfigurer">
35   <property name="properties">
36    <props>
37     <prop key="seeds.textSource.value">http://www.ciens.ucv.ve/ciens/</prop>
38    </props>
39   </property>
40  </bean>
```

Figura 8. Semilla en el archivo de configuración del job(García & Rivero, 2013)

Luego de que las solicitudes de rastreo almacenadas en el *crontab* del sistema son ejecutados, se verifican las solicitudes de rastreos con las almacenadas en el *crontab* tal

como se puede apreciar en la Figura 9, para luego verificar que cada *job* sea colocado en ejecución de forma exitosa. Cuando esto sucede se crea una carpeta como la que se puede ver en la Figura 10.

```
1  23 1 */1 */1 * sh /home/ejecutar_rastreo.sh ciens-2013-10-06-01-21-17
```

Figura 9. Solicitud de rastreo almacenada en el crontab (García & Rivero, 2013)

Nombre de archivo	Tamaño de archivo	Tipo de archivo	Última modifica...	Permisos	Propietario.
..					
actions-done		Carpeta de archi...	06/10/2013 01:...	drwxr-xr-x	root root
logs		Carpeta de archi...	07/10/2013 01:...	drwxr-xr-x	root root
reports		Carpeta de archi...	07/10/2013 01:...	drwxr-xr-x	root root
warcs		Carpeta de archi...	07/10/2013 01:...	drwxr-xr-x	root root
crawler-beans.cxml	29.730	Archivo CXML	06/10/2013 01:...	-rw-r--r--	root root
job.log	386	Documento de te...	07/10/2013 01:...	-rw-r--r--	root root

Figura 10. Rastreo colocado en ejecución (García & Rivero, 2013)

Cuando una solicitud de rastreo almacenada en el *crontab* es ejecutada, se quiere verificar que el proceso fue exitoso. Entonces se toma como entrada la lista de rastreos activos y se ejecuta el script encargado de verificar rastreos finalizados tal y como se muestra en la figura 11. Seguidamente se revisa la lista de rastreos por indexar para verificar que efectivamente todos los rastreos que estaban en la lista de activos y hayan finalizado se encuentren en esta lista, la misma se visualiza en la figura 12.

```
38  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20130930070004
39  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20130930070103
40  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
41  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131001070005
42  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
43  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131001070104
44  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131002070007
45  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131002070103
46  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131002070203
47  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131003070006
48  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131002070203
49  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131003070103
50  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131004070005
51  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131004070103
52  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131004070202
53  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131005070006
54  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
55  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131005070103
56  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131006043928
57  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131006070003
58  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131006043928
```

Figura 11.Rastreos finalizados (García & Rivero, 2013)

```
1  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20130930070004
2  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20130930070103
3  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
4  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131001070005
5  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
6  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131001070104
7  /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131002070007
8  /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131002070103
9  /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131002070203
10 /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131003070006
11 /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131002070203
12 /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131003070103
13 /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131004070005
14 /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131004070103
15 /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20131004070202
16 /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131005070006
17 /data/heritrix-3.1.1/jobs/uc-2013-09-30-02-30-35/20130930070203
18 /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131005070103
19 /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131006043928
20 /data/heritrix-3.1.1/jobs/ula-2013-09-30-02-28-50/20131006070003
21 /data/heritrix-3.1.1/jobs/udo-2013-09-30-02-29-41/20131006043928
```

Figura 12. Rastreos por indexar (García & Rivero, 2013)

2.4.5 Acceso

El acceder a un Archivo Web, se refiere a la acción de hacer una consulta por parte de los usuarios.

Existe una gran necesidad de proveer métodos para acceder a un Archivo Web. Puede ser un acceso proveniente de una dirección URL, un motor de búsqueda o de Minería de Datos. Sin importar cuál sea el método utilizado, se debe crear una interfaz gráfica para el usuario.

Para almacenar los archivos, se utiliza un registro de encabezado con una identificación única que se basa generalmente en la dirección del archivo y una marca de tiempo. Esta marca de tiempo se utiliza para señalar las distintas versiones que pueda tener el archivo hasta el momento que se haga la búsqueda por parte del usuario. La función que realiza la búsqueda devuelve una lista de referencias que enlazan con el archivo y se utilizan como punto de entrada para el usuario. En la actualidad, la mayoría de los archivos utilizan este método para proporcionar acceso a los documentos, ya que es relativamente

más sencillo indexar los registros de las cabeceras de la manera antes mencionada(García & Rivero, 2013).

2.4.6 Tecnologías Usadas

2.4.6.1 Software

- ReST como arquitectura de software para la transferencia de datos.
- HTTP como protocolo de comunicación y para la integración con el Prototipo.
- Ruby como lenguaje de programación con su *framework* Rails.
- Estándar de formato de archivos WARC (ISO, 2009).

2.4.6.2 Hardware

Se utiliza un Servidor HP ProLiant ML110 G7 E3-1220 1P con un procesador Intel QuadCore Xeon de 3.1 GHz y 10 GB de RAM y 6 TB de disco duro Este servidor fue configurado con 4 máquinas virtuales, cuatro de las cuales tienen instalado Heritrix para rastreos y una tiene instalado Solr para indexación (ver Figura 13).

De estas cuatro máquinas virtuales solamente 2 tienen acceso público y son aquellas donde se alojarán las aplicaciones resultantes de los subsistemas a desarrollar, éstas son:

- IP: 190.169.69.154, Dominio: solr.ciens.ucv.ve
- IP: 190.169.69.155, Dominio: heritrix1.ciens.ucv.ve

El repositorio de archivos WARC se encuentra distribuido entre las 4 máquinas que tienen instalado Heritrix y su ubicación es manejada por los índices de Solr.

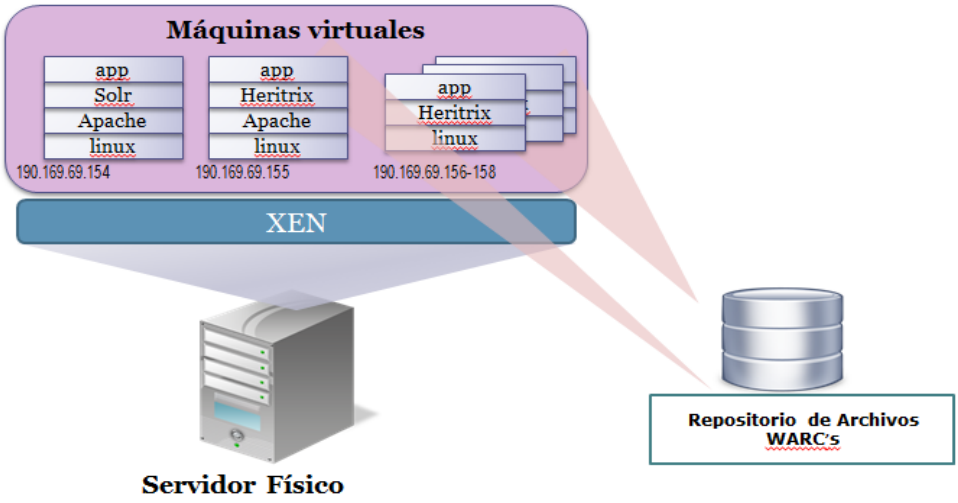


Figura 13. Configuración de la plataforma de desarrollo (Ospina M., 2014)

2.4.7 Arquitectura de la aplicación actual

La aplicación actual del Prototipo de Archivo Web de Venezuela cuenta con tres módulos los cuales se pueden observar en la figura 14, donde se describe la arquitectura del Prototipo usando diagramas de componentes.

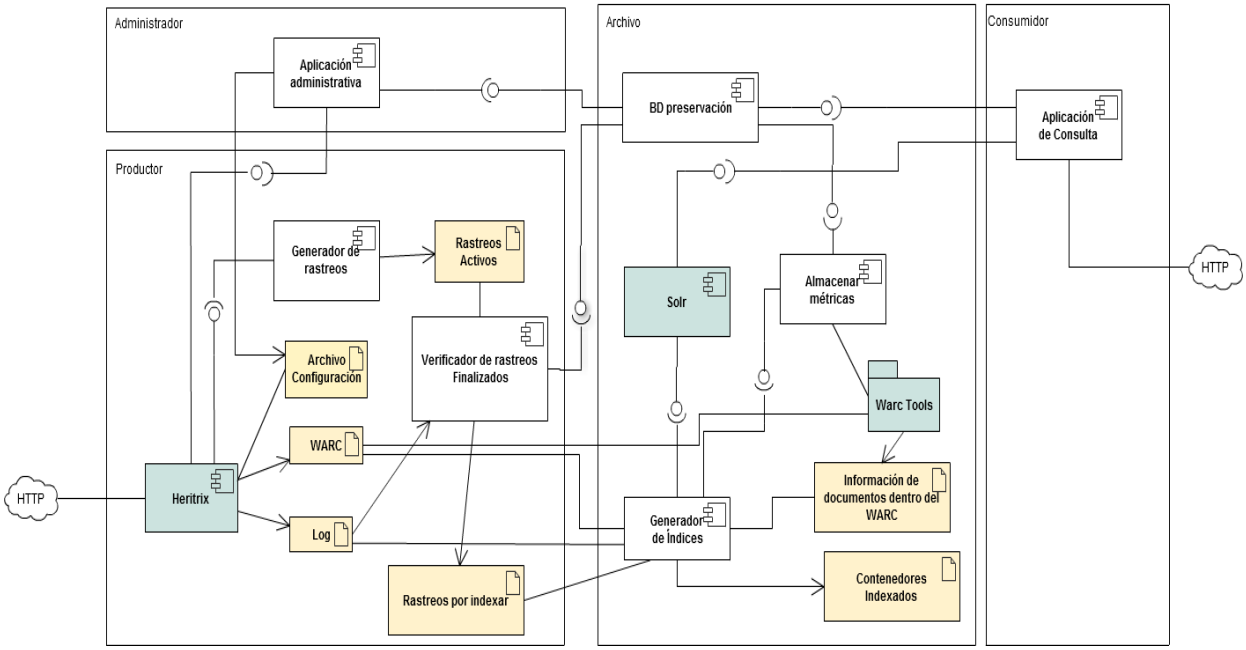


Figura 14. Arquitectura del Prototipo de Archivo Web usando un Diagrama de componentes (García & Rivero, 2013)

El Módulo de Adquisición, permite la suscripción de sitios web y su adquisición periódica. El Módulo de Indexación realiza la gestión del almacenamiento llevando índices para el repositorio de archivos WARC.

El Módulo de Acceso y Despliegue de archivos WARC, permite consultar los sitios suscritos, sus versiones y ver los contenidos preservados de manera similar a como se ven en el servidor original.

Es importante destacar, que aunque existe una aplicación administrativa en el Módulo de Adquisición, esta solo cumple con la función de realizar suscripciones de sitios web.

2.4.8 Interfaces

El Prototipo de Archivo Web posee las siguientes interfaces:

2.4.8.1 Aplicación administrativa

En la Figura 15 se muestra la interfaz de autenticación de la aplicación para los suscriptores y en la Figura 16 se muestra el listado de los rastreos realizados o en proceso, sobre los sitios suscritos por el usuario.

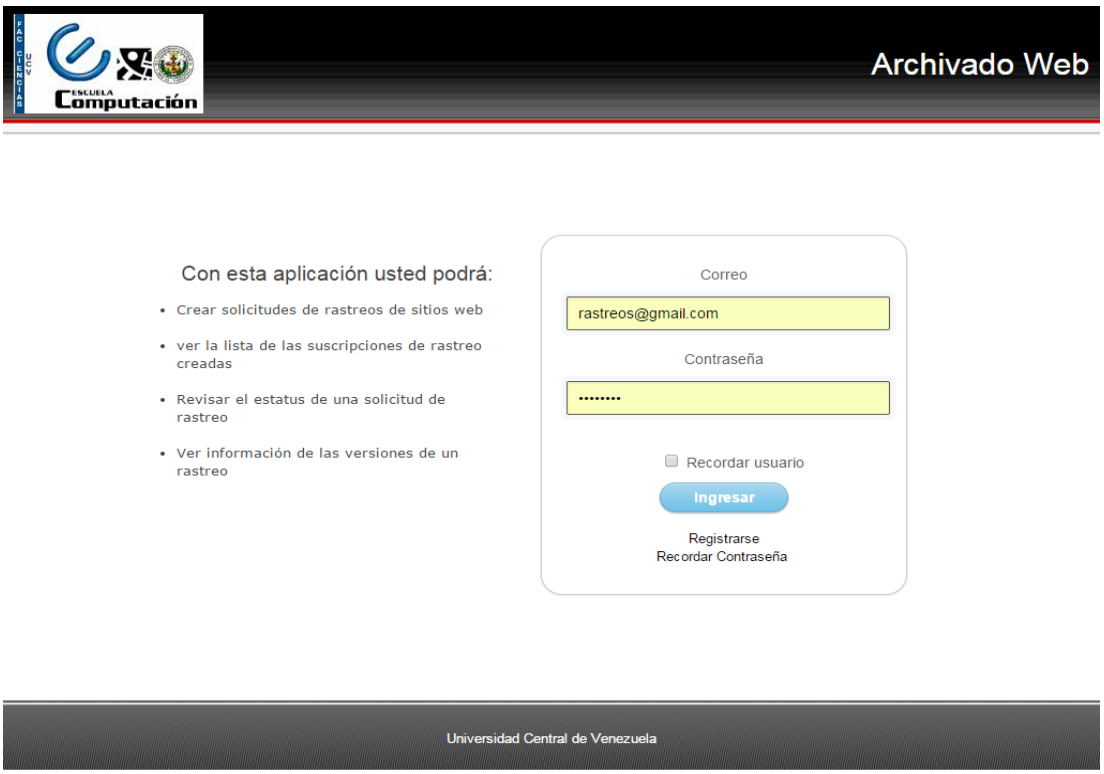


Figura 15. Interfaz de autenticación de la Aplicación de Adquisición (García & Rivero, 2013)

Figura 16. Listado de rastreos del usuario autenticado (García & Rivero, 2013)

En la figura 17 se ilustra el formulario que debe realizar los suscriptores al momento de realizar la creación de un nuevo rastreo.

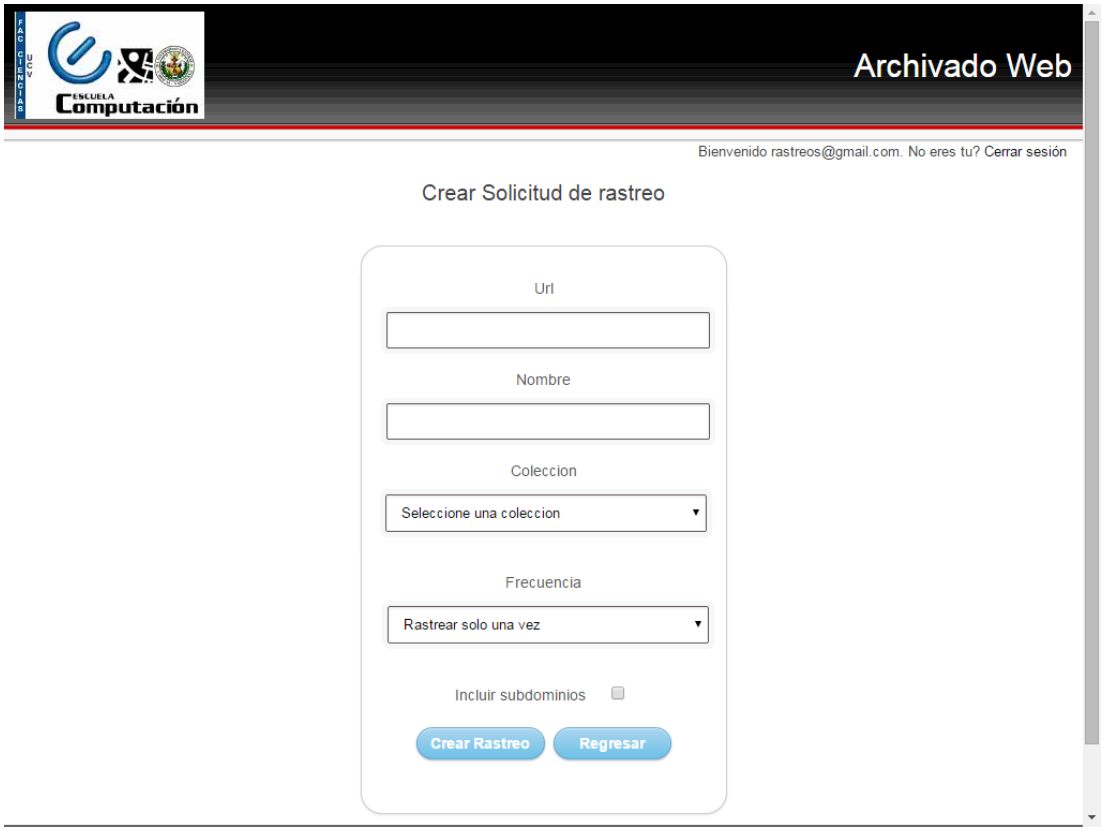


Figura 17. Formulario para la creación de nuevos rastreos (García & Rivero, 2013)

2.4.8.2 Aplicación de acceso

En la figura 18 se observa la página principal de la aplicación y en la figura 19la interfaz de las versiones de un sitio web.

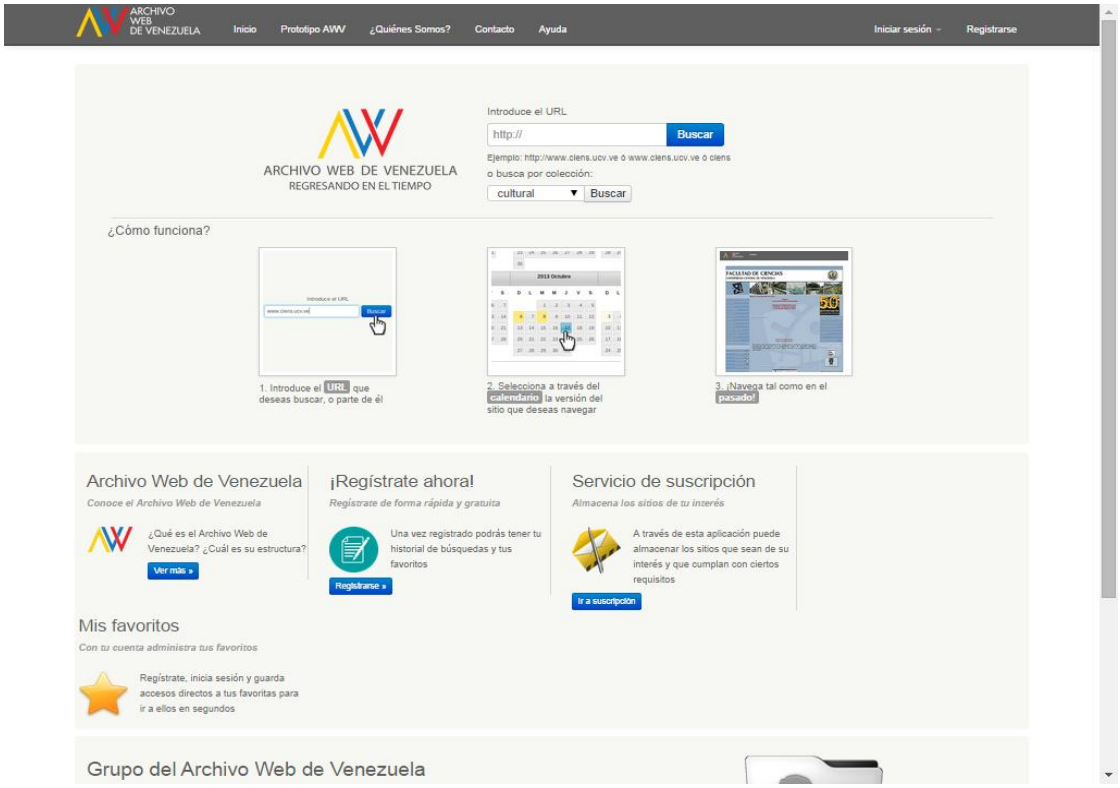


Figura 18. Página principal de la aplicación (Mantura & Martínez, 2014)

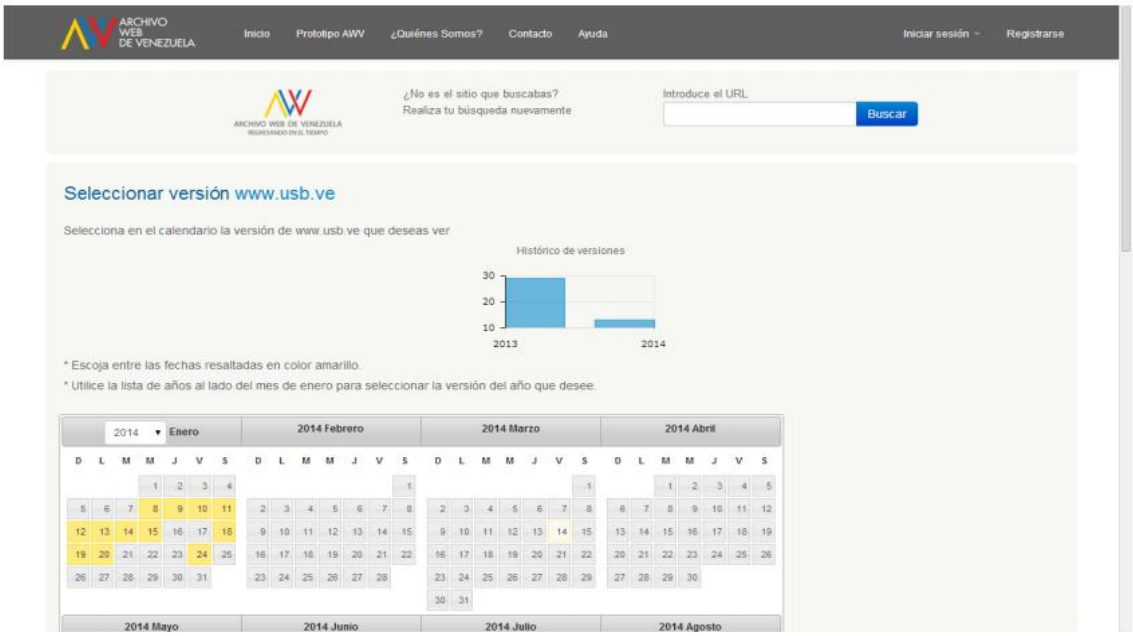


Figura 19. Versiones del sitio web www.usb.ve (Mantura & Martínez, 2014)

2.4.9 Lenguaje de Programación: Ruby

Los Módulos de Adquisición y Almacenamiento y el Módulo de Acceso del Prototipo de Archivo Web de Venezuela fueron desarrollados e implementados en lenguajes Ruby bajo su *framework* Rails.

Ruby es un lenguaje de programación dinámico y de código abierto enfocado en la simplicidad y productividad. Su elegante sintaxis se siente natural al leerla y fácil al escribirla. Fue creado por el programador japonés Yukihiro "Matz" Matsumoto, quien comenzó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995.

Ruby es orientado a objetos, todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas. Ruby ha sido descrito como un lenguaje de programación multiparadigma, permite programación procedural (definiendo funciones y variables fuera de las clases haciéndolas parte del objeto raíz *Object*), con orientación a objetos, (todo es un objeto) o funcionalmente (tiene funciones anónimas, clausuras o *closures*, y *continuations*; todas las sentencias tiene valores, y las funciones de vuelven la última evaluación) (Ruby).

2.4.9.1 Framework Rails

Rails es un *framework* de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros *frameworks* y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de *RubyGems*, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby (RubyonRails, 2003).

2.4.10 Lenguaje de Programación: Python

En el siguiente apartado se describirá el lenguaje de programación a utilizar en la implementación y desarrollo del Módulo de Gestión y Control de Incidencias; como ya fue mencionado en los puntos anteriores que el Prototipo de Archivo Web está desarrollado en Ruby bajo su *framework* Rails.

La razón principal de porque escoger Python es porque es un lenguaje de programación poderoso que se ha venido trabajando a lo largo de la carrera. Tiene una sintaxis muy limpia lo que la hace muy cercana al lenguaje pseudo-formal, facilitando la

curva de aprendizaje del mismo. Además, nos encontramos familiarizados con el entorno de programación y el ambiente de ejecución para realizar las configuraciones necesarias. Aunque no presenta experiencias con Preservación Web en su haber y por las razones antes expuesta es que se elige a Python como lenguaje de programación candidato para la propuesta del desarrollo de módulo de administración de un Archivo Web.

Python es un lenguaje de programación dinámico y orientado a objetos que puede ser usado de muchas maneras en el desarrollo de software. Ofrece gran soporte e integración con otros lenguajes y herramientas, viene con una extensiva cantidad de librerías. (Lambert, 2012)

Python se puede ejecutar en Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, y teléfonos celulares Nokia. Python también ha sido portado para las máquinas virtuales de Java y .NET. Python es distribuido bajo la licencia *open source OSI* que lo hace libre para ser usado inclusive en el desarrollo de productos comerciales.

2.4.10.1 Características:

- Lenguaje Interpretado y orientado a objetos
- Multiplataforma
- De sintaxis sencilla.
- Permite dividir su programa en módulos reutilizables desde otros programas en Python.
- Incorpora una gran colección de módulos estándar que puedes utilizar como base de los programas (o como ejemplos para empezar a aprender Python).
- Hay módulos incluidos que proporcionan E/S de archivos llamadas al sistema, *"sockets''* e interfaces gráficas con el usuario.
- Permite escribir programas muy compactos y legibles. Con compactos se quiere expresar que con muy pocas líneas de código se puede lograr diversas funcionalidades. Los programas escritos en Python son normalmente mucho más cortos que sus equivalentes en C o C++.

2.4.10.2 Frameworks para Python

Un *framework* web es una colección de paquetes o módulos que permiten a los desarrolladores escribir aplicaciones o servicios sin tener que manejar la búsqueda de detalles de bajo nivel como los protocolos, desarrollo de plugin, conexiones y creaciones de base de datos y desarrollo de manejador de contenidos entre otros.

Actualmente la mayoría de los frameworks web trabajan con tecnología del lado del servidor y del cliente. (Python Software Foundation, 2001). Se describe a continuación Django que será el Framework utilizado en el desarrollo del trabajo.

2.4.10.3 Django

Es un *framework* de desarrollo web de código abierto, escrito en Python, que respeta el paradigma conocido como Modelo-*Template*-Vista. Fue desarrollado en origen para gestionar varias páginas orientadas a noticias de la *World Company* de Lawrence, Kansas, y fue liberada al público bajo una licencia BSD en julio de 2005. El framework fue nombrado en alusión al guitarrista de jazz gitano Django Reinhardt (Django , 2005).

2.4.10.4 Características

- Es un *framework* de desarrollo web
- Código abierto
- Permite construir aplicaciones web más rápido
- Utilizando menos código
- Principio DRY (Don't Repeat Yourself).
- Legible.
- Django, es un *framework* que permite construir aplicaciones web más rápido y con menos código. Fue inicialmente desarrollado para gestionar aplicaciones web de páginas orientadas a noticias de *World Online*, más tarde se liberó bajo licencia BSD.
- Django se centra en automatizar todo lo posible y se adhiere al principio DRY (No te repitas a ti mismo).
- La meta fundamental de Django es facilitar la creación de sitios web complejo poniendo énfasis en el re-uso, la conectividad y extensibilidad de componentes y el desarrollo rápido. Python es usado en todas las partes del framework, incluso en configuraciones, archivos, y en los modelos de datos.

2.4.10.5 Python vs Ruby

A pesar de los numerosos argumentos entre ambos lenguajes, es imposible decir si uno es "mejor que el otro". Está claro que cada uno tiene algunas áreas que se adapta mejor, debido a sus características y el apoyo de otros usuarios en la misma zona. Para Rubí esto es el desarrollo web a través del framework Rails, y para Python es la programación científica y académica, pero sin dejar a un lado el desarrollo web bajo el

framework Django. Y cada uno tiene algunas características o capacidades que el otro no tiene.

Cualquier cosa que usted puede hacer en Ruby on Rails también puede hacer en Python y Django. Qué lenguaje es mejor no es realmente una cuestión de capacidad, en realidad es una cuestión de preferencia para el desarrollador. La tabla 2 muestra aspectos relevantes del Lenguaje Ruby contrastándolas con los aspectos relevantes del lenguaje Phyton.

Tabla 2. Comparativa entre *Python y Ruby* (Chris Oliver, 2015)

	Ruby	Python
Origen	Creado en 1995 por Yukihiro Matsumoto	Creado en 1991 por Guido Van Rossum
PROS	• Toneladas de características fuera de la caja para el desarrollo web • Rápida para abrazar nuevas cosas	• Muy fácil de aprender • Comunidad diversa con grandes vínculos con Linux y el mundo académico
CONTRAS	En ocasiones puede ser muy difícil de depurar	A menudo muy explícito y poco elegante para leer
Frameworks Web	Ruby on Rails – Iniciado en 2005 por David Heinemeier Hansson	Django - Iniciado en 2003 por Adrian Holovaty y Simon Willison
Comunidad	Innova más rápido pero causa más cosas para arreglar	• Muy estable y diversa pero innova más lento. • Se utiliza ampliamente en el mundo académico y Linux.
Usos	• Apple • Twitter • Github	• Google • Pinterest • Instagram • Mozilla Firefox • National Geographic • The Washington Post

Fuente: Traducido de http://learn.onemonth.com/ruby-vs-python

Capítulo III Marco Metodológico

En esta sección se explicará que es una metodología de desarrollo de software, además se explicará la metodología seleccionada para la implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web de Venezuela.

3.1 Metodologías de Desarrollo de Software

Todo proceso de desarrollo de software involucra fases para la conceptualización, elaboración y diseño del mismo, este proceso no es más que *"aquel en el que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo"* (Jacobson, 1998).

Poner en práctica el desarrollo de un software no resulta una tarea sencilla, es por ello que se debe integrar una metodología que incluya las fases del ciclo de vida de un software de manera específica y concreta para obtener el resultado esperado, un software de calidad. La metodología se trata de un conjunto de métodos, principios y reglas que permiten enfrentar de manera sistemática el desarrollo de software.

Para el desarrollo de software, se requiere de diversos elementos que agrupados hacen que el desarrollo de software sea o no exitoso.

Para esto existen las metodologías tradicionales que se modificaron para poder aplicarlas al desarrollo de software, aunque durante mucho tiempo fueron la única solución al desarrollo, hizo estas metodologías poco flexibles y estrictas. Estas consistían en una serie fundamentos y conceptos aplicados al desarrollo de software, documentación, planificación y procesos.

Ante las dificultades de las metodologías tradicionales referentes al tiempo y flexibilidad, surgen las metodologías ágiles que emergen como posible alternativa para solventar los diferentes inconvenientes que antes se presentaban al aplicar un esquema tradicional en el proceso de desarrollo del software. Esto no implica que las metodologías ágiles sean propicias para todo tipo de proyectos, puesto que algunos requieren una mayor cantidad de artefactos asociados que provean cierto nivel de detalle y especificación, pero permite lograr una simplificación al momento de ejecutar las actividades previstas. La tabla 3 ilustra las diferencias entres las metodologías tradicionales y las metodologías agiles.

Tabla 3. Comparación entre las Metodologías tradicionales y Ágiles

Metodologías Tradicionales	Metodologías Ágiles
– Rigidez ante los cambios, de manera lentos o moderada – Los clientes interactúan con el equipo de desarrollo mediante reuniones – Grupos de gran tamaño y varias veces distribuidos en diferentes sitios – Dependencia de la arquitectura de software mediante modelos – Poco Feedback lo que extiende el tiempo de entrega – Mínimos roles – Basadas en normas de estándares de desarrollo – Procesos muy controlados por políticas y normas – Seguimiento estricto del plan inicial de desarrollo	- Flexibilidad ante los cambios del proyecto de forma moderada a rápida - Los clientes hacen parte del equipo de desarrollo - Grupos pequeños (promedio 10 participantes in situ) en el mismo lugar. - Menor dependencia de la arquitectura de software - Continuo Feedback acortando el tiempo de entrega - Diversidad de roles - Basadas en heurísticas a partir de prácticas de producción de código - Procesos menos controlados, pocas políticas y normas - Capacidad de respuesta ante los cambios

Fuente: Tomado de https://www.google.co.ve/url?sa=t&rct=j&q=&esrc=s&source=web&cd=6&cad=rja&uact=8&sqi=2&ved=0CD0QFjAF&url=http%3A%2F%2Fdialnet.unirioja.es%2Fdescarga%2Farticulo%2F4752083.pdf&ei=m9BTVcCfNYq0sASW6IC4CQ&usg=AFQjCNEmq9BimRXMHE333cJPRHBEwoHkaA&sig2=xJRA-JPw_LGMSZuNAyTHvg

Existen diversas metodologías ágiles, como:

- Programación Extrema (XP, por sus siglas en inglés)
- SCRUM
- Proceso Unificado Ágil (AUP, por sus siglas en inglés)

3.2 Metodología AUP

AUP es una Metodología de Desarrollo basada en la simplificación del *Rational Unified Process* o Proceso Racional Unificado (RUP) de IBM. Cuenta con un enfoque disciplinado hacia las prácticas de pruebas industriales para el diseño de software y sistemas dentro de una organización de desarrollo.

Este enfoque aplica técnicas ágiles e incluyen desarrollo basado en pruebas (TDD), Modelado basado en desarrollo ágil (AMDD), gestión de cambios ágil, y refactorización de base de datos para mejorar su productividad.

El ciclo de vida de AUP es serial en lo grande e iterativo en lo pequeño, liberando entregables incrementales en el tiempo. El ciclo de vida de AUP se muestra en la Figura 20.

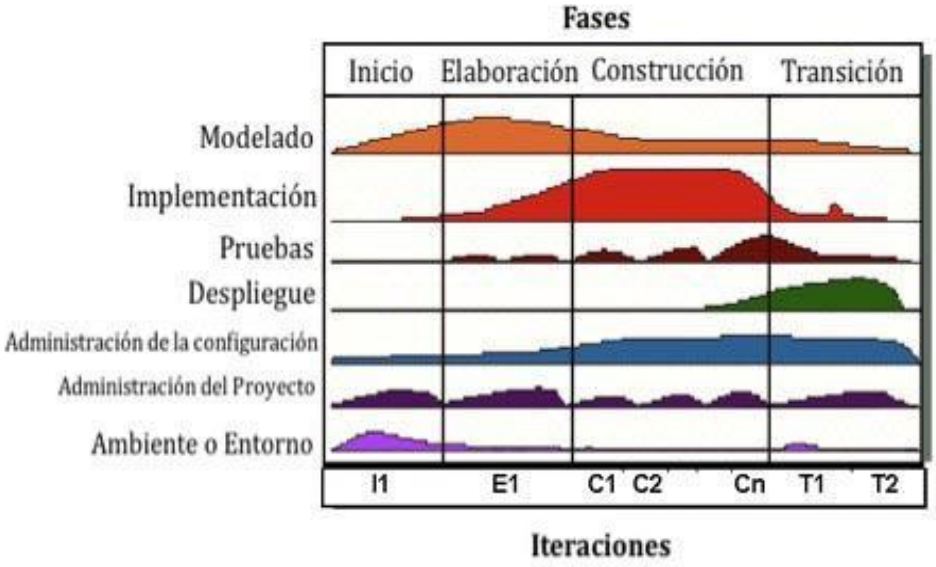


Figura 20. Ciclo de vida de Ágil UP, Traducida de (The Agile Unified Process, 2006)

3.2.1 Disciplinas o Iteraciones.

a) **Modelado:** La meta de ésta disciplina es entender el negocio de la organización, el dominio del problema que el proyecto aborda e identificar una solución viable para abordar el dominio del problema.

   No es necesario crear todos los modelos que existan para trabajar en un sistema pero es importante que se tenga en el proyecto aquellos que sean adecuados para la situación Su objetivo es crear modelos que son solamente lo suficientemente buenos para su situación a la mano, siempre puede ir atrás

y mejorar su objetivo después cuando necesite más detalles o la situación de cambios.

a) **Implementación:** La meta de ésta disciplina es transformar los modelos en un código ejecutable y realizar una prueba de nivel básico en una unidad particular de prueba.

b) **Pruebas:** La meta de ésta disciplina es ejecutar una evaluación de los objetivos para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funcione como fue diseñado y verificar que los requerimientos están completos.

c) **Despliegue:** La meta de ésta disciplina es planificar la entrega del sistema y ejecutar el plan para que el sistema esté disponible para los usuarios finales.

d) **Administración de la Configuración:** La meta de ésta disciplina es administrar el acceso a los entregables o productos del proyecto. Esto incluye no sólo el rastreo de versiones del producto en el tiempo, sino que también incluye controlar y administrar los cambios que ocurran.

e) **Administración del Proyecto:** La meta de ésta disciplina es dirigir las actividades que se llevan a cabo en el proyecto. Esto incluye administración del riesgo, la dirección de personas (asignar tareas, seguimiento de los procesos, etc.), y coordinar con los sistemas y personas fuera del alcance del proyecto para que el este termine a tiempo y dentro del presupuesto.

f) **Ambiente o entorno:** La meta de ésta disciplina es apoyar el resto de los esfuerzos para garantizar que, el proceso adecuado, la orientación (normas y directrices) y herramientas (hardware, software, etc.) estén disponibles para el equipo según sea necesario.

3.2.2 Fases

Las fases son implementadas de una forma serial a lo largo de un proyecto siguiendo la metodología AUP. Estas fases son:

1) **Inicio:** El objetivo principal de la fase de inicio es definir el alcance del proyecto, donde se establecen los límites desde dónde el equipo operará. También se realiza un potencial de arquitectura de la aplicación para así obtener financiamiento y la aceptación de las partes interesadas.

Principales actividades que se deben realizar para esta fase:

- Modelado de diagramas de casos de uso y diagrama de clases
- Diseño de Prototipos de interfaz.
- Realizar plan de pruebas iníciales.
- Iniciar plan de despliegue.
- Estimar riesgos y factibilidad del proyecto.

2) Elaboración: El principal objetivo de la fase de elaboración es probar la arquitectura del sistema que se va a desarrollar. El punto es asegurar que el equipo puede desarrollar un sistema que pueda satisfacer los requisitos planteados.

En esta fase, el equipo también se prepara para la Fase de Construcción. Se comienza con la creación del ambiente propicio para la Construcción mediante la configuración de hardware, software y herramientas.
Principales actividades que se deben realizar para esta fase:

- Modelar la arquitectura.
- Implementar la arquitectura.
- Validar la arquitectura.
- Actualizar plan del proyecto.

3) Construcción: El objetivo de la fase de Construcción consiste en construir software de trabajo, en una base regular e incremental que cumpla las necesidades prioritarias de los interesados en el proyecto.

Principales actividades que se deben realizar para esta fase:

- Documentación.
- Primeras pruebas.
- Desarrollo de interfaces de activos legados.
- Evolucionar el entorno de trabajo.

4) Transición: Se enfoca en llevar el sistema a producción. Se debe realizar pruebas extensivas a lo largo de esta fase, incluyendo las pruebas beta. Una

buena afinación del proyecto tiene lugar en esta fase, incluyendo el trabajo dirigido a los defectos no tan relevantes. El tiempo y esfuerzo necesarios en la Transición varía según el desarrollo del proyecto completo.

Principales actividades que se deben realizar para esta fase:

- Finalizar documentación.
- Corregir defectos.
- Validar el sistema.
- Finalizar el paquete de entrega o liberación.

3.2.3 Entrega de versiones incrementales en el tiempo

Los equipos de desarrollo AUP suelen emitir revisiones al final de cada iteración en escenarios de reproducción, cosa que no sucede en el enfoque "*bigbang*", donde se cumple con la entrega de todos los programas a la vez. Una versión de desarrollo de una aplicación es algo que podría ser liberado en producción si pasa a través de un módulo de aseguramiento de calidad (también llamado ambiente *QA Quality Assurance* o Aseguramiento de la Calidad), las pruebas y procesos de despliegue, en pre-producción.

3.3 Arquitectura de software basada en componentes

La arquitectura basada en componentes consiste en una rama de la Ingeniería de software que se enfoca principalmente en la descomposición del software en componentes funcionales (Ingeniería de software basada en componentes, 2013). Esta descomposición permite convertir componentes pre-existentes en piezas más grandes de software.

Un Componente es un elemento de un sistema software que ofrece un conjunto de servicios, o funcionalidades, a través de interfaces definidas (Ingeniería de software basada en componentes, 2013).

La Figura 21 muestra la notación UML utilizada para representar un componente, las funcionalidades que el componente provee a través de una interfaz son representadas con un círculo al final de una línea desde el componente (1).

Figura 21. Componente de software (Ian, 2006)

Para especificar los servicios que necesita unos componentes se utiliza una interfaz representada con un semicírculo al final de una línea desde el componente (2).

El proceso de construcción de una pieza de software con componentes ya existentes, da origen al principio de reutilización del software, mediante el cual se promueve que los componentes sean implementados de una forma que permita su utilización funcional sobre diferentes sistemas en el futuro.

Un componente de software es similar a una caja negra, en donde para el exterior se tiene una especificación general, la cual es independiente de la especificación interna(Ian, 2006). Existen tres conceptos ligados a la definición de un componente:

- **Interior del componente:** es una pieza de software que cumple con un conjunto de propiedades y que se encuentra conformada como un artefacto del cual se espera que sea reutilizable.
- **Exterior del componente:** es una interface que cumple con un conjunto de propiedades y provee un servicio a usuarios humanos u otros artefactos de software.
- **Relación interior-exterior:** proceso de relación entre el interior y exterior del componente, aquí son claves conceptos como especificación, implementación y encapsulación.

Existen 4 principios que definen a un componente de software como elemento de la arquitectura (Ian, 2006):

- **Múltiple uso**: Un componente debe ser significativo es decir, debe estar dentro de un contexto que permita que su funcionalidad sea útil en la creación de distintas piezas de software.

- **Contexto no específico**: La especificación de un componente debe estar planteada forma general que permita su adaptación en distintos sistemas, sin que el contexto tenga prioridad.
- **Encapsulación**: Especificación interna o implementación del componente, no es investigable a través de la interfaz. El resto de componentes o piezas de software dentro de un sistema no se ven afectados por cambios en el diseño de uno de los componentes.
- **Una unidad independiente de desarrollo con su propio control de versiones**: Permite que un componente pueda ser desarrollado de manera independiente, cambiando el diseño o agregando nuevas funcionalidades, sin afectar significativamente el resto del sistema.

Existen 3 aspectos fundamentales respecto a la estructura de la arquitectura basada en componentes:

1. **El nombre de los componentes:** Debe identificar las funcionalidades de un componente así como su uso que tiene como software, generalmente se siguen convenciones, estándares para facilitar la identificación de los componentes.
2. **La interface de los componentes:** La interface puede ser vista como el área de intercambio entre el interior y el exterior de un componente de software, a través de la interfaz es que se puede acceder a las funcionalidades que brinda el componente de software más sin embargo no se puede acceder a sus especificación interna, además de la interfaz se debe proporcionar documentación relacionada a como se debe utilizar el componente.
3. **Cuerpo y código de Implementación:** Es el área del componente donde se encuentra el código o implementación de los servicios y funcionalidades brindados por el componente, debe cumplir con el principio de encapsulación.

En la estructura de la arquitectura basada en componentes existen 2 procesos fundamentales para el desarrollo (Ian, 2006):

1. **Ensamblajes de sistema a partir de componentes de software**: Este proceso está compuesto por 4 actividades:

- **Análisis de los componentes**: En esta fase de determinar qué tan adecuado es un componte para la construcción del sistema: Esta evaluación se realiza de acuerdo a un conjunto de métricas y criterios establecidos por los analistas y diseñadores de la arquitectura.
- **Adaptación de los componentes**: Es posible que el componente esté orientado hacia cierto contexto, atado a funcionalidades específicas, es por ello que en necesario adaptar el componente: Existen 3 maneras de adaptar un componente.
    a) ***White-Box*:** cuando el componente debe ser reescrito para operar en conjunto con el resto de componentes del sistema.
    b) ***Grey-Box***: cuando el componente incorpora su propio API (*Programming Interface*).
    c) ***Black-Box***: cuando el componente no posee un API. Una interfaz completamente independiente es construida para acceder a los servicios del componente.
- **Ensamblaje de los componentes:** Se integran los componentes a través de la estructura mediante la cual fueron definidos.
- **Mantenimiento:** Una vez el funcionamiento ya está en funcionamiento puede que sea necesario hacer cambios en los componentes ya sea por nuevos requerimientos o inconvenientes con los mismos, estos cambios pueden consistir en la reescritura o sustitución del componente.

2. **Reusabilidad:** La capacidad de poder utilizar una pieza de software dentro de otro sistema es una de las características más importantes en una arquitectura basada en componentes, para lograr la reutilización es necesario un esfuerzo extra de los desarrolladores basado en lo siguiente:
    - Una documentación completa de cada atributo y funcionalidad del componente
    - Una etapa de pruebas organizada y certera que certifique el correcto funcionamiento del componente.
    - Una definición de comprobaciones precisa para el chequeo de cada parámetro de entrada (*input*) del componente.
    - Un manejo de notificaciones de errores preciso, que advierta de la existencia de estos de una forma apropiada.

- Desarrollar teniendo en cuenta que el componente puede ser requerido para trabajar en muchos contextos muy diferentes unos de otros (tomar en cuenta la eficiencia, uso de memoria y recursos).

Capítulo IV Desarrollo de la Aplicación

En este capítulo se describirán los objetivos generales y específicos de la aplicación así como la adaptación de la metodología de desarrollo de software AUP dentro del paradigma de desarrollo Basado en Componentes, durante el desarrollo del Módulo de Gestión y Control de Incidencias.

4.1 Objetivo General de la Aplicación

Desarrollar el Módulo Gestión y Control de Incidencias para extender y mejorar las funcionalidades del Módulo de Adquisición y Almacenamiento del Prototipo de Archivo Web de Venezuela, de manera que además de realizar suscripciones de rastreos y almacenar e indexar las versiones cosechadas, brinde la supervisión de los nuevos registros de usuarios, para su autorización como suscriptores, y supervise los trabajos de rastreos para permitir la recuperación en caso de encontrar incidencias en los mismos.

4.2 Objetivos Específicos de la Aplicación

- Proporcionar una interfaz institucional para el Prototipo de Archivo Web de Venezuela.
- Crear el Módulo de Gestión y Control de Incidencias.
- Realizar el reinicio de los rastreos, al momento de presentarse una falla en el aplicativo (inconvenientes con el servicio eléctrico).
- Permitir la comunicación directa entre los suscriptores y el administrador, mediante el envío de correos con temas de interés.
- Hacer uso de tecnologías existentes para el desarrollo del módulo correspondiente.

4.3 Alcance de la Aplicación

Este desarrollo representa la tercera versión del Prototipo de Archivo Web en la cual se implementa la modificación de la interfaz de las funcionalidades de los Módulos de Adquisición y Almacenamiento, para proporcionar una interfaz institucional en el Prototipo de Archivo Web de Venezuela.

Se logra la separación de las funcionalidades realizadas en la aplicación con la creación del rol de administrador, el cual se encarga de la supervisión de las solicitudes de

registro de usuarios, la autorización o rechazo de dichas solicitudes, la supervisión y monitoreo de los trabajos de rastreos realizados por los usuarios y el reinicio de los trabajos de rastreos por fallas en el aplicativo (inconveniente con el servicio eléctrico).

Se logra el envío de mensajes al administrador por parte de los suscriptores, con información o solicitudes que sean de relevancia o sugerencias que deseen realizar.

4.4 Adaptación de la metodología AUP usando una arquitectura de software basada en componentes

Para el desarrollo del Prototipo de Archivado Web se implementó las fases que sigue la metodología AUP, como se mencionó anteriormente esta metodología sigue los principios de la Modelación Ágil y nos da la oportunidad de crear de manera rápida y sencilla la documentación del proyecto. También se hizo uso de los artefactos o entregables UML para explicar cómo se adapta la metodología AUP en una arquitectura de Software Basada en Componentes.

A continuación se describe la adaptación de las fases de la metodología AUP en el presente T.E.G.

4.4.1 Fase de Inicio

En esta fase, como se ha descrito anteriormente se indican los requerimientos funcionales a desarrollar.

4.4.1.1 Requisitos Funcionales

Módulo de Adquisición

- Solicitar Registro Nuevo Suscriptor (Modificación de la funcionalidad en el Módulo existente (García & Rivero, 2013)).
- Contactar al administrador (Modificación de la funcionalidad en el Módulo existente (García & Rivero, 2013)).

Módulo de Control y Gestión de Incidencias

- Iniciar sesión.
- Administración de Suscriptores.
- Denegar Acceso a URLs ya registradas.
- Crear solicitud de rastreo.

- Editar solicitud de rastreo.
- Desactivar solicitud de rastreo.
- Visualizar información de las solicitudes creadas por suscriptores.
- Configuración de parámetros de rastreo.
- Reiniciar los rastreos en caso de fallas por inconvenientes con el servicio eléctrico.
- Contactar Suscriptores.

Estos requisitos serán explicados en detalle en la sección 4.4.2.4 donde se encuentran los casos de uso y los diagramas de secuencias.

4.4.1.2 Usuarios del Sistema

Este Prototipo provee funcionalidades que permitan el rastreo de sitios web, supervisión de los trabajos de rastreo, reiniciar los rastreos en caso de falla y la creación de nuevas colecciones. Por tal motivo, el Módulo de Gestión y Control de Incidencias está diseñado para dos tipos de usuarios:

- **Usuario Público:** corresponde a todo aquel usuario que accede a la aplicación a través de Internet. Este grupo se divide en dos tipos:
  - **No registrado:** este tipo de usuario cuenta con la opción de solicitar registro para acceder a la aplicación, adicionalmente se le permite la comunicación con el administrador con la opción de Contáctanos.
  - **Registrado (Suscriptor):** Estos usuarios cuenta con las opciones de gestión (creación, consulta, edición y desactivar) los trabajos de rastreos, adicionalmente se le permite la comunicación con el administrador con la opción de Contáctanos.
- **Usuario Administrador:** Representa aquel usuario encargado de monitorear los trabajos de rastreos de la aplicación, gestionar las solicitudes de registro, reiniciar los rastreos en caso de fallas por inconvenientes con el servicio eléctrico. Adicional puede realizar trabajos de rastreo, editar, consultar y desactivar los rastreos de los suscriptores, realizar la creación de nuevas colecciones donde se podrá catalogar a los rastreos.

4.4.1.3 Arquitectura de la Aplicación

Como se ha indicado anteriormente el Prototipo realizado se basa en una arquitectura basada en componentes, la cual se visualiza en la Figura 22.

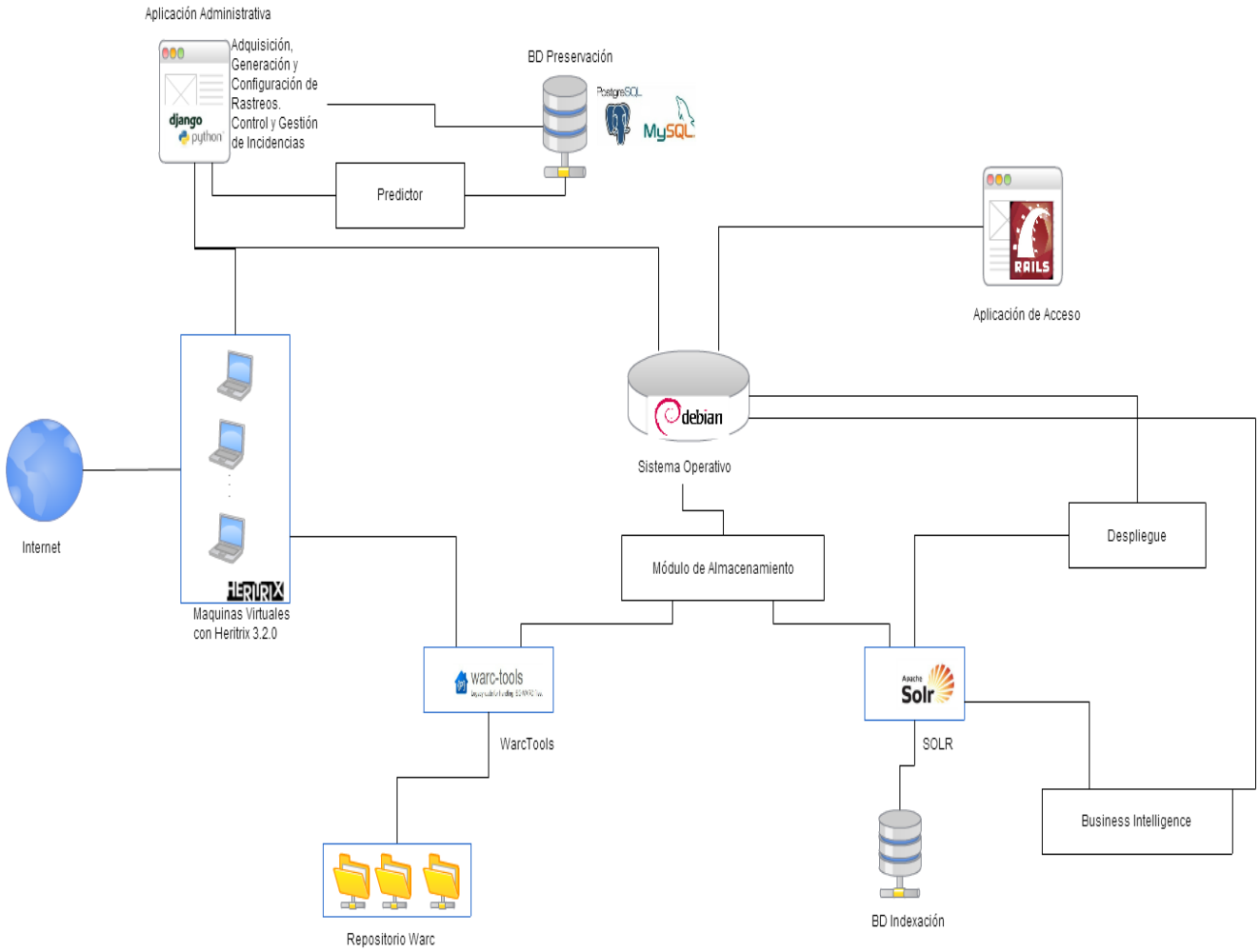


Figura 22. Arquitectura del Prototipo de Archivo Web de Venezuela a implementar

4.4.1.4 Entorno de Trabajo

Luego de realizar las gestiones pertinentes se obtuvieron los recursos necesarios para adquirir el hardware donde funcionará el Prototipo de Preservación Web, se cuenta con un servidor dividido en cuatro máquinas virtuales, adicionalmente fue adquirido un disco duro de 1 Terabyte dividido y asignado a cuatro de las máquinas virtuales, este espacio de almacenamiento es necesario para almacenar los archivos WARC cosechados, desde este

mismo momento se toma en cuenta la escalabilidad del sistema desde el punto de vista del espacio de almacenamiento, si más espacio es requerido es posible integrar discos duros adicionales.

A cada una de las máquinas virtuales se le asignó un nombre y una dirección IP:

- Solr - 190.169.69.154
- Heritrix1 - 190.169.69.155
- Heritrix2 - 190.169.69.156
- Heritrix3 - 190.169.69.157

El software requerido y/o seleccionado es el siguiente:

- Sistema Operativo Linux- Debian.
- Lenguaje de programación Python y su Framework Django.
- Rastreador Heritrix.
- Manejador de base de datos MySQL.

4.4.2 Fase de Elaboración

En esta fase se diseñó y modeló la arquitectura del sistema la cual da soporte para a todos los requerimientos funcionales diseñados.

4.4.2.1 Prototipos de Interfaz de Usuario

A continuación se muestran los Prototipos de interfaces de usuario diseñados para el sistema.

Iniciar Sesión

Como se puede observar en la Figura 23 se cuenta con un formulario que tiene 2 campos que el usuario debe completar para iniciar sesión en el sistema.

Figura 23. Prototipo de Interfaz Iniciar Sesión

Solicitud de Registro

Como se puede observar en la Figura 24 se cuenta con un formulario que el usuario debe completar para poder registrarse en el sistema, así como también el mensaje generado al ingresar la información de registro.

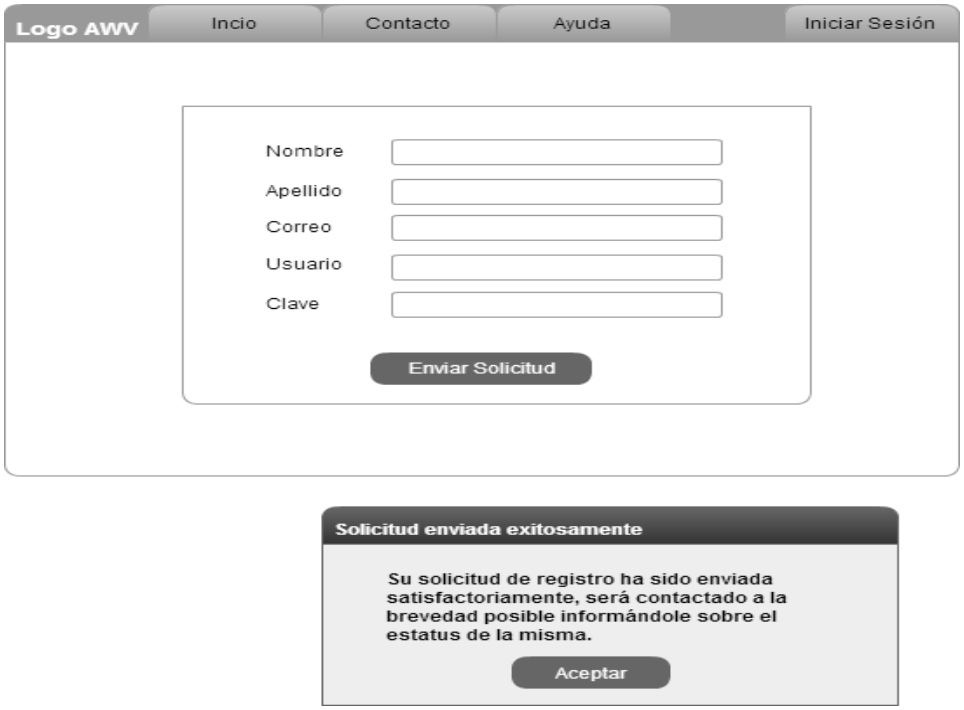


Figura 24. Prototipo de Interfaz de Solicitud de Registro

Lista de Gestión de Registro - Administrador

En esta interfaz se puede visualizar de forma tabular los datos de las solicitudes de registro de usuarios, esto se puede apreciar en la Figura 25.

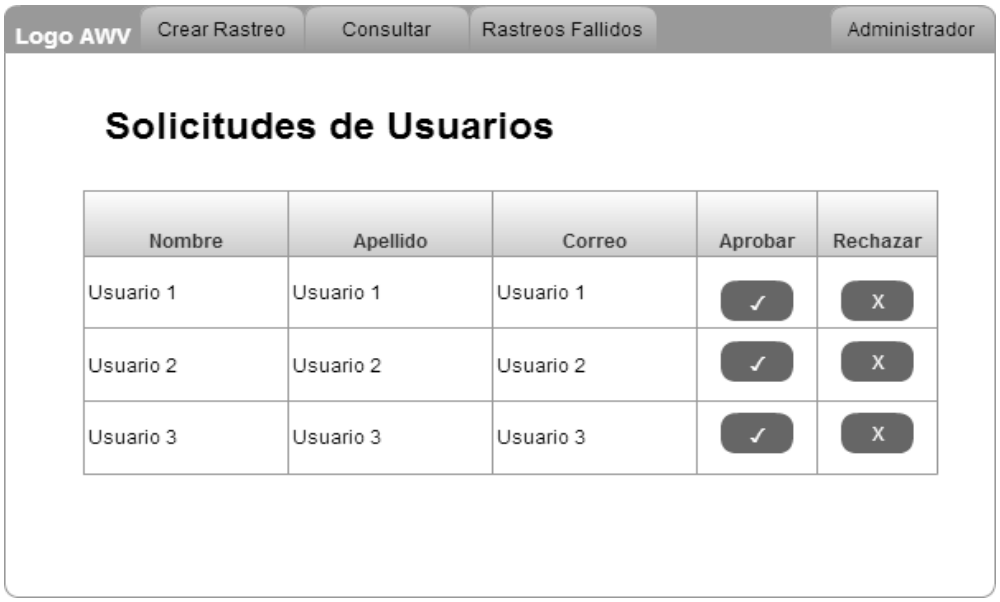


Figura 25. Prototipo de Interfaz de Gestión de Registro de Usuarios

Visualizar Rastreos Fallidos

Como se puede ver en la Figura 26 los datos de los rastreos fallidos son mostrados de forma tabular.

Figura 26. Prototipo de Interfaz Visualizar Rastreos Fallidos

Opción Contáctanos

En la Figura 27 se aprecia el formulario que debe completar los usuarios para enviar información del interés al administrador.

Figura 27. Prototipo de Interfaz Opción Contáctanos

Lista de Solicitudes de Rastreo

En esta interfaz se puede visualizar de forma tabular los datos de las solicitudes de rastreo que el usuario ha creado, esto se puede apreciar en la Figura 28.

Figura 28. Prototipo de Interfaz Consultar Rastreos

Crear Solicitud de Rastreo

Como se puede ver en la Figura 29, se cuenta con un formulario que permite crear solicitudes de rastreo.

Figura 29. Prototipo de Interfaz Crear Rastreos

4.4.2.2 Diagramas de Componentes

Como se ha comentado anteriormente se decidió utilizar una arquitectura basada en componentes, en la Figura 30 se puede apreciar el diagrama de componentes del Módulo de Gestión y Control de Incidencias, estos componentes serán descritos en la sección 4.4.4.3.1 Implementación de Componentes.

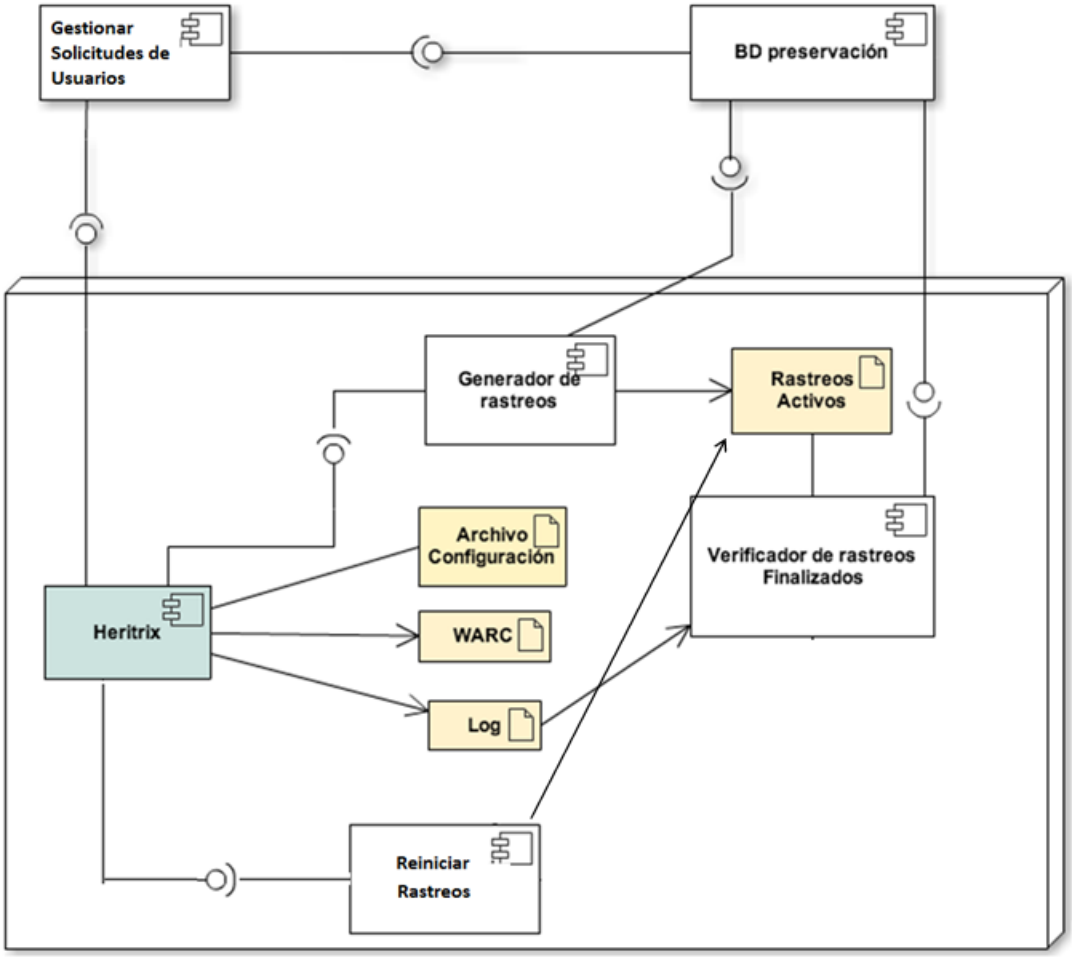


Figura 30. Diagrama de Componentes del Prototipo de Archivo Web

4.4.2.3 Diagramas de Actividades

En esta sección se describe la actividad incluida al Prototipo de Archivo Web de Venezuela.

Reiniciar Rastreo Fallido

Para reiniciar un rastreo fallido se siguen la serie de pasos mostrados en la Figura 31.

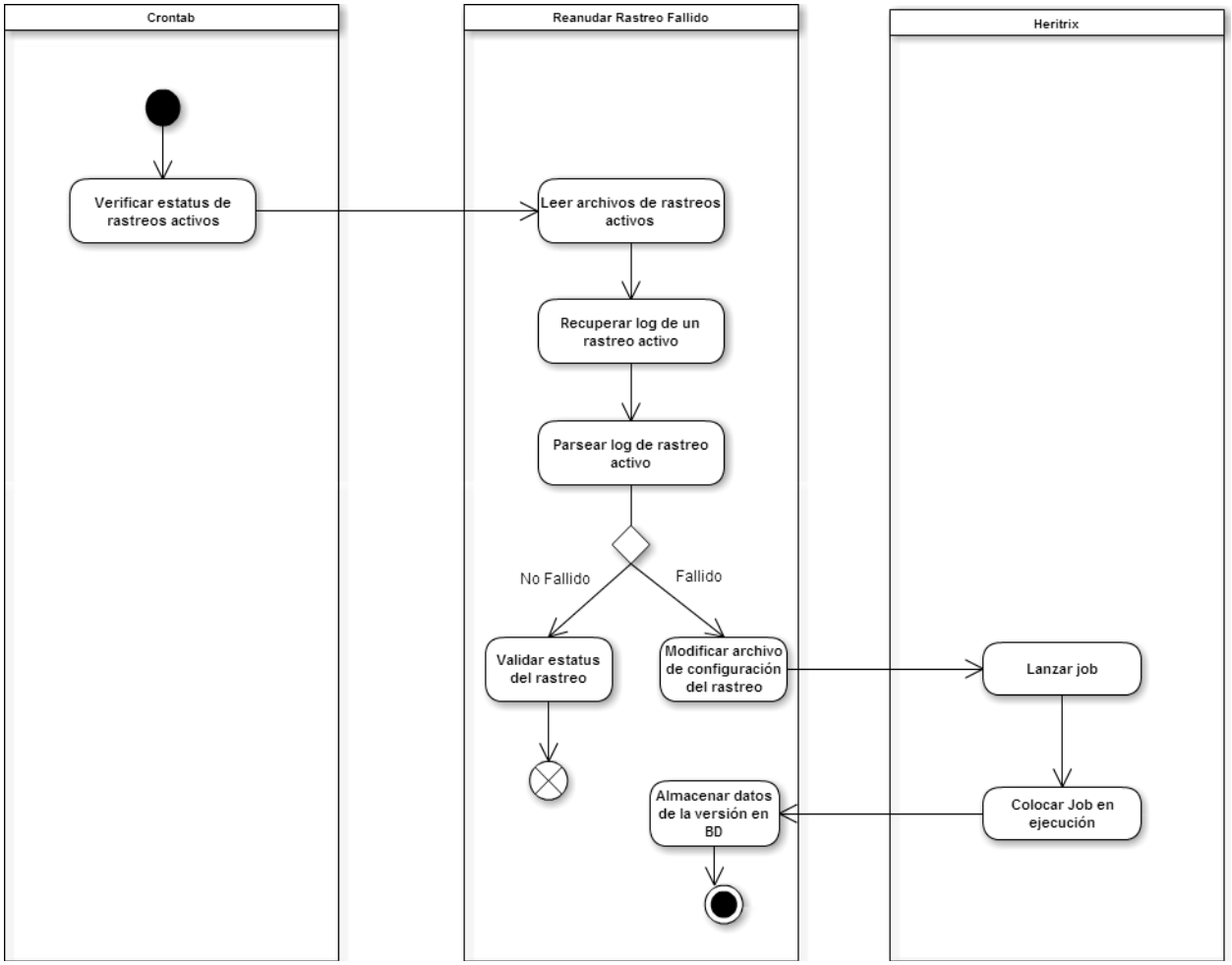


Figura 31. Diagrama de actividades Reiniciar Rastreo Fallido

4.4.2.4 Casos de Uso y Diagramas de Secuencia

Por cuestiones de legibilidad y mayor entendimiento se decidió separar los casos de uso por módulo, es decir, casos de usos con la modificación realizada a las funcionalidades del módulo de adquisición y casos de uso para el módulo de gestión y control de incidencias.

Casos de uso y Diagramas de Secuencia del Módulo de Adquisición

En la Figura 35 se puede apreciar el diagrama de casos de uso de las funcionalidades modificadas realizadas en el Módulo de Adquisición (García & Rivero, 2013).

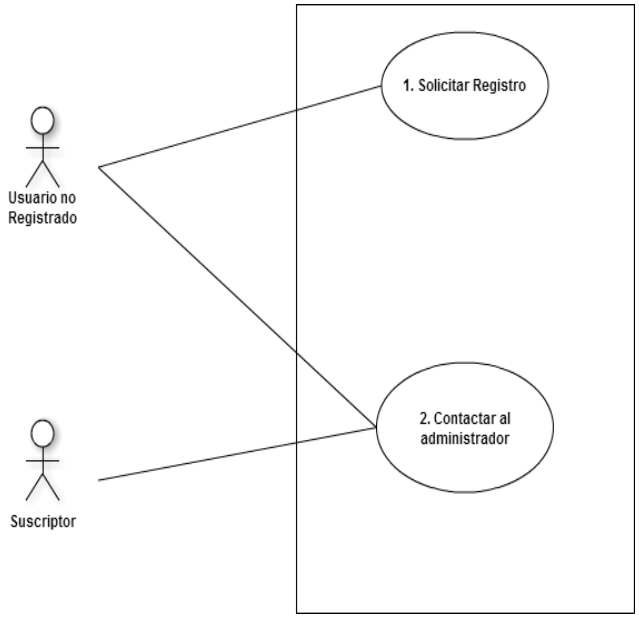


Figura 32. Diagrama de casos de uso módulo de adquisición

A continuación se encuentra la especificación de los casos de uso y los diagramas de secuencia de las modificaciones realizadas en las funcionalidades del Módulo de Adquisición:

Tabla 4. Caso de Uso Solicitar Registro

Nombre:	Solicitar Registro
Autor:	Gamar Milano
Número de CU	CU-1
Descripción: Le permite al usuario no registrado registrarse en la aplicación **Actores:** Usuario No Registrado **Pre-condiciones:** **Flujo Normal:** 1. El usuario no registrado ingresa a la aplicación 2. El usuario no registrado selecciona la opción de registrarse 3. El usuario no registrado introduce los datos solicitados 4. El sistema valida los datos 5. El sistema le notifica al usuario no registrado que su registro fue enviado al administrador **Flujo Alternativo:** 1. Si hay un error en los datos se le notifica al usuario no registrado que debe corregirlos	

Post-condiciones: Solicitud de registro enviada

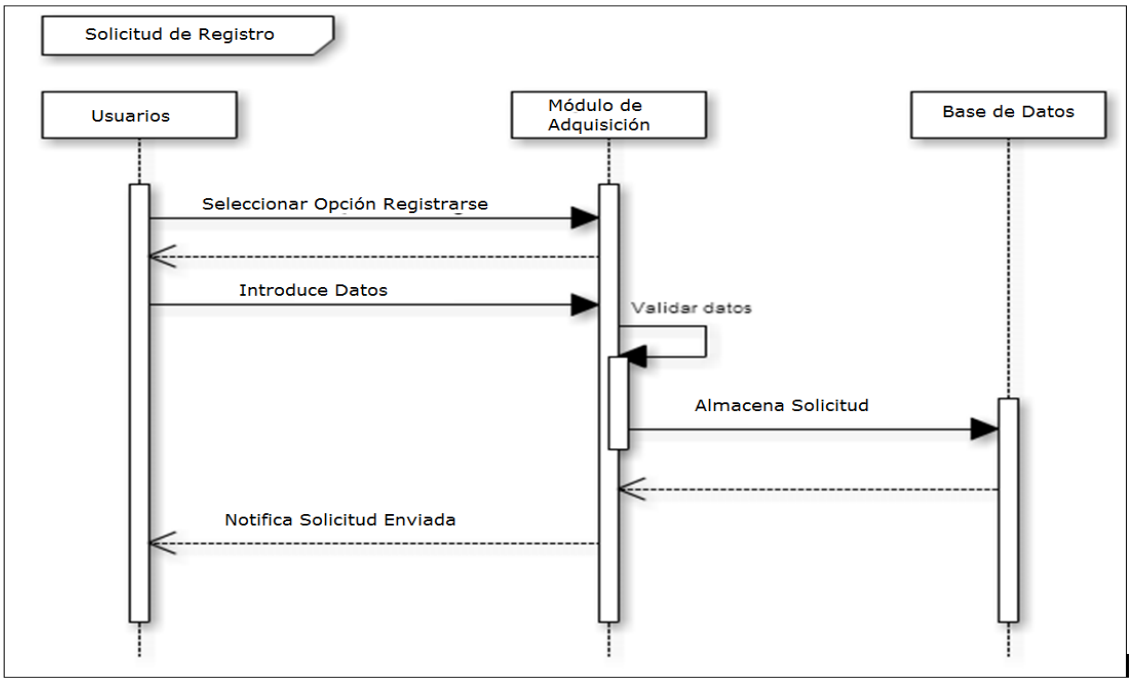


Figura 33. Diagrama de Secuencia Solicitar Registro

Tabla 5. Caso de Uso Contactar al Administrador (García & Rivero, 2013)

Nombre:	Contactar al Administrador
Autor:	Gamar Milano
Número de CU	CU-2
Descripción: Permite al suscriptor o al usuario no registrado la interacción con el administrador **Actores:** Suscriptor y Usuario no registrado **Pre-condiciones:** **Flujo Normal:** 1. El suscriptor o el usuario no registrado selecciona la opción de Contáctanos 2. Ingresa la información que será enviada al administrador 3. Oprime el botón enviar 4. El sistema envía un correo al administrador con la información ingresada **Flujo Alternativo:** **Post-condiciones:**	

Fue enviado el correo al administrador con la información del suscriptor

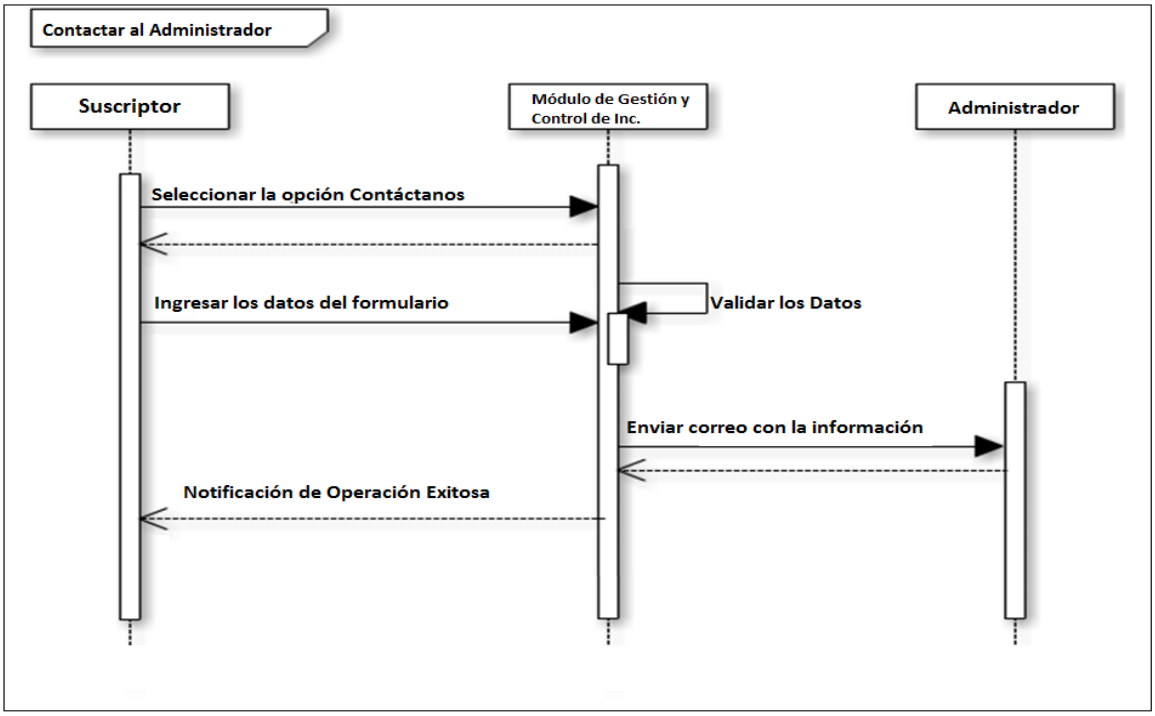


Figura 34. Diagrama de Secuencia Contactar al Administrador

Casos de uso y Diagramas de Secuencia del Módulo de Gestión y Control de Incidencias

En la Figura 35 se puede apreciar el diagrama de casos de uso para el Módulo de Gestión y Control de Incidencias.

Figura 35. Diagrama de casos de uso módulo de gestión y control de incidencias

A continuación se encuentra la especificación de los casos de uso y los diagramas de secuencia del Módulo de Gestión y Control de Incidencias:

Tabla 6. Caso de Uso Iniciar Sesión

Nombre:	Iniciar Sesión
Autor:	Laura Sánchez
Número de CU	CU-3

Descripción:
Le permite al administrador iniciar sesión en el sistema
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador ingresa a la aplicación
2. El administrador selecciona la opción de iniciar sesión
3. El administrador introduce los datos solicitados
4. El sistema valida los datos
5. Se le notifica al administrador que su inicio de sesión fue exitoso
6. Se redirecciona al administrador al inicio de la aplicación

Flujo Alternativo:

1. Si hay un error en los datos se le notifica al administrador que debe corregirlos

Post-condiciones:
El administrador ha iniciado sesión en la aplicación

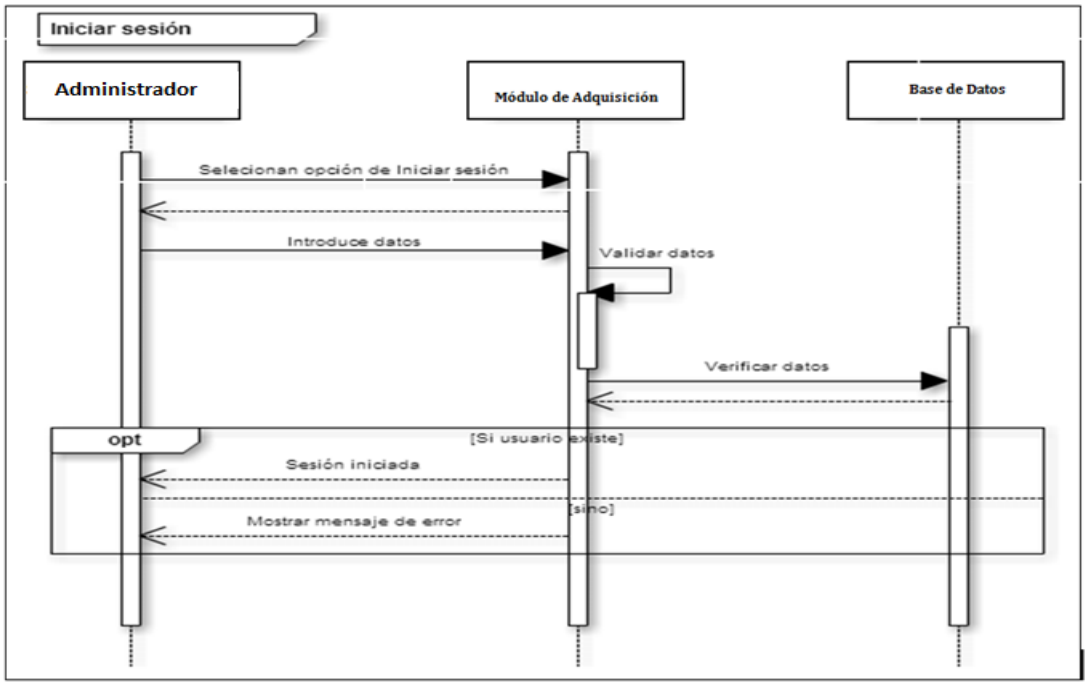


Figura 36. Diagrama de Secuencia Iniciar Sesión Administrador

Tabla 7. Caso de Uso Cambiar Contraseña

Nombre:	Cambiar Contraseña
Autor:	Laura Sánchez
Número de CU	CU-3.1

Descripción:
Le permite al administrador cambiar su contraseña
Actores:
Administrador
Pre-condiciones:
CU-9
Flujo Normal:
1. El administrador selecciona la opción de cambiar contraseña
2. El administrador introduce su contraseña actual y la nueva contraseña
3. El sistema valida los datos
4. El sistema almacena la nueva contraseña

Flujo Alternativo:
1. Si las contraseñas no coinciden se le notifica al administrador

Post-condiciones:
El administrador ha cambiado su contraseña

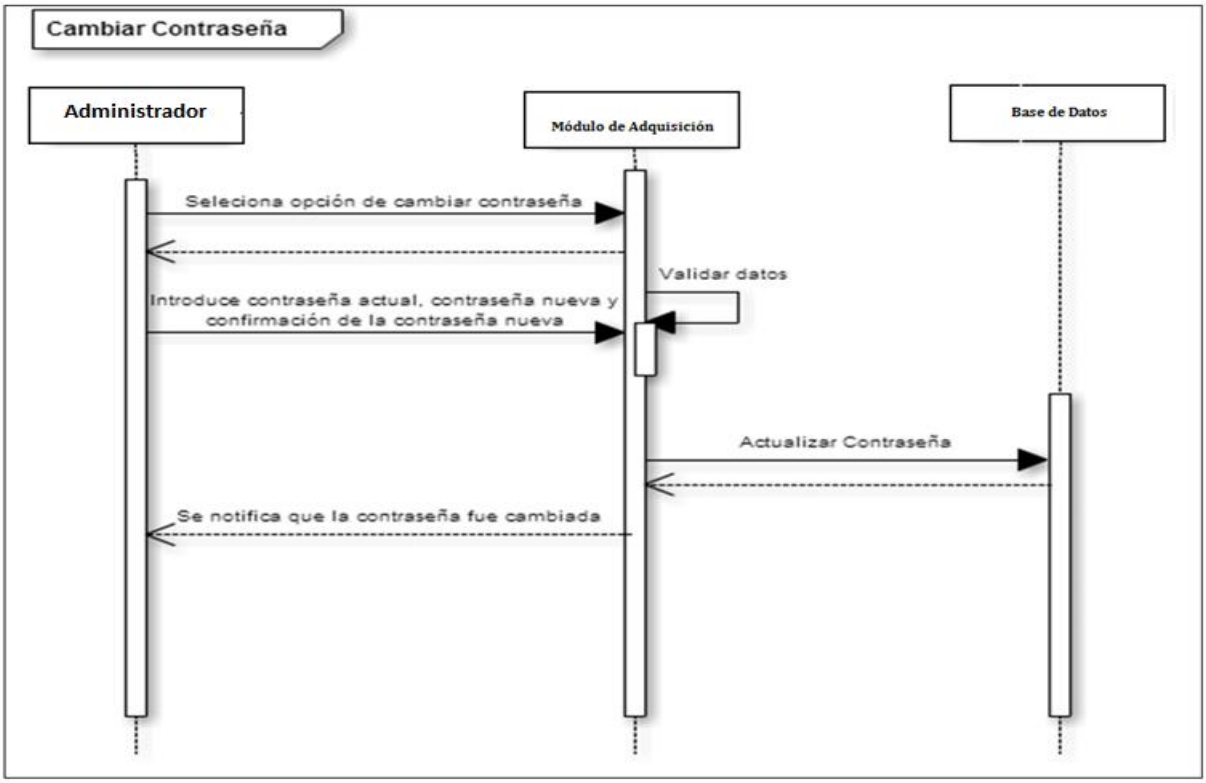


Figura 37. Diagrama de Secuencia Cambiar Contraseña Administrador

Tabla 8. Caso de Uso Visualizar Solicitudes de Usuarios

Nombre:	Visualizar Solicitudes de Usuarios
Autor:	Laura Sánchez
Número de CU	CU-4

Descripción:
Le permite al administrador admitir o denegar el acceso a los suscriptores
Actores:
Administrador
Pre-condiciones:
CU-9
Flujo Normal:
1. El administrador ingresa a la aplicación
2. El sistema consulta en Base de Datos las solicitudes de registro

Flujo Alternativo:
Post-condiciones:
El administrador visualiza la información de los usuarios no registrados

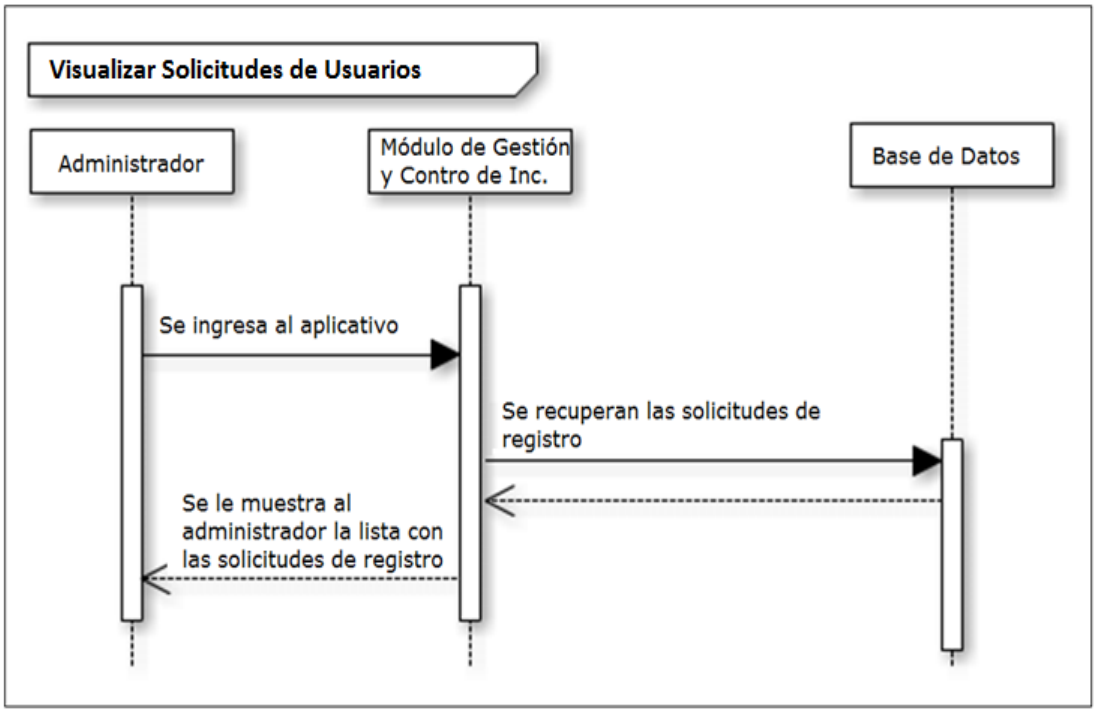


Figura 38. Diagrama de Secuencia Visualizar Solicitudes de Usuarios

Tabla 9. Caso de Uso Autorizar Usuarios

Nombre:	Autorizar Usuarios
Autor:	Laura Sánchez
Número de CU	CU-4.1

Descripción:
Le permite al administrador admitir el acceso a los usuarios
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador ingresa a la aplicación
3. El sistema consulta en Base de Datos las solicitudes de registro
2. El administrador visualiza la solicitud de nuevos usuarios
3. El administrador admite el acceso de los usuarios
4. El sistema envía un correo electrónico informando a los usuarios su estatus en la aplicación

Flujo Alternativo:
Post-condiciones:
Usuario con acceso permitido a la aplicación

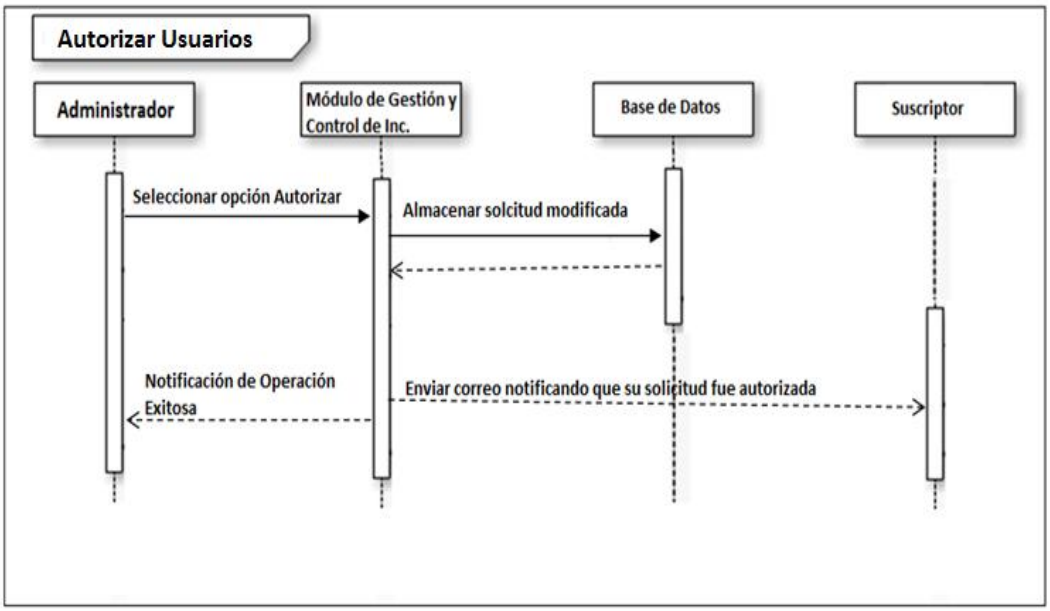


Figura 39. Diagrama de Secuencia Autorizar Usuarios

Tabla 10. Caso de Uso Denegar Acceso a Usuarios

Nombre:	Denegar Acceso a Usuarios
Autor:	Laura Sánchez
Número de CU	CU-4.2

Descripción:
Le permite al administrador denegar el acceso a los usuarios
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador ingresa a la aplicación
4. El sistema consulta en Base de Datos las solicitudes de registro
2. El administrador visualiza la solicitud de nuevos usuarios
3. El administrador rechaza el acceso de los usuarios
4. El sistema envía un correo electrónico informando a los usuarios su estatus en la aplicación

Flujo Alternativo:
Post-condiciones:
Usuario con acceso denegado a la aplicación

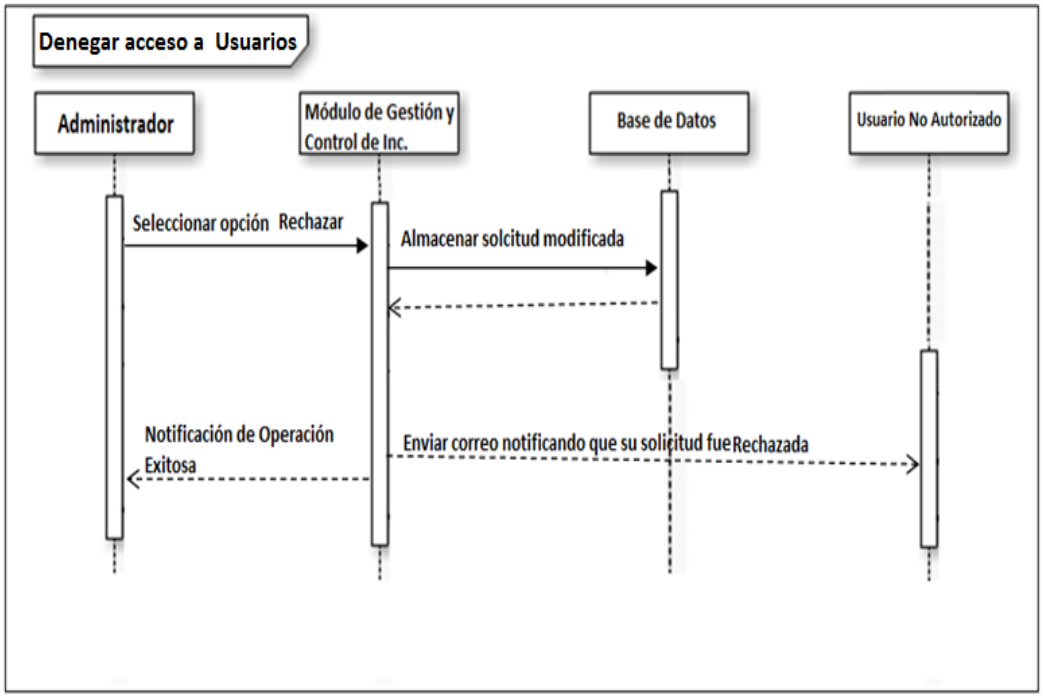


Figura 40. Diagrama de Secuencia Denegar Acceso Usuarios

Tabla 11. Caso de Uso Crear Solicitud de Rastreo

Nombre:	Crear Solicitud de Rastreo
Autor:	Laura Sánchez
Número de CU	CU-5

Descripción:
Permite al administrador crear una solicitud de rastreo
Actores:
Administrador
Pre-condiciones:
CU-9
Flujo Normal:

1. El administrador selecciona la opción de crear rastreo
2. El administrador introduce los parámetros de configuración del rastreo
3. El sistema valida los datos
4. La solicitud de rastreo es almacenada en base de datos
5. Se asigna el rastreo a una de las máquinas virtuales
6. En la máquina virtual seleccionada se crea un *job* con los parámetros proporcionados por el administrador
7. la solicitud de rastreo se agrega en el *crontab* de la máquina virtual seleccionada

Flujo Alternativo:

1. Si hay un error en los datos se le notifica al administrador que debe corregirlos

Post-condiciones:
Se ha creado una nueva solicitud de rastreo.

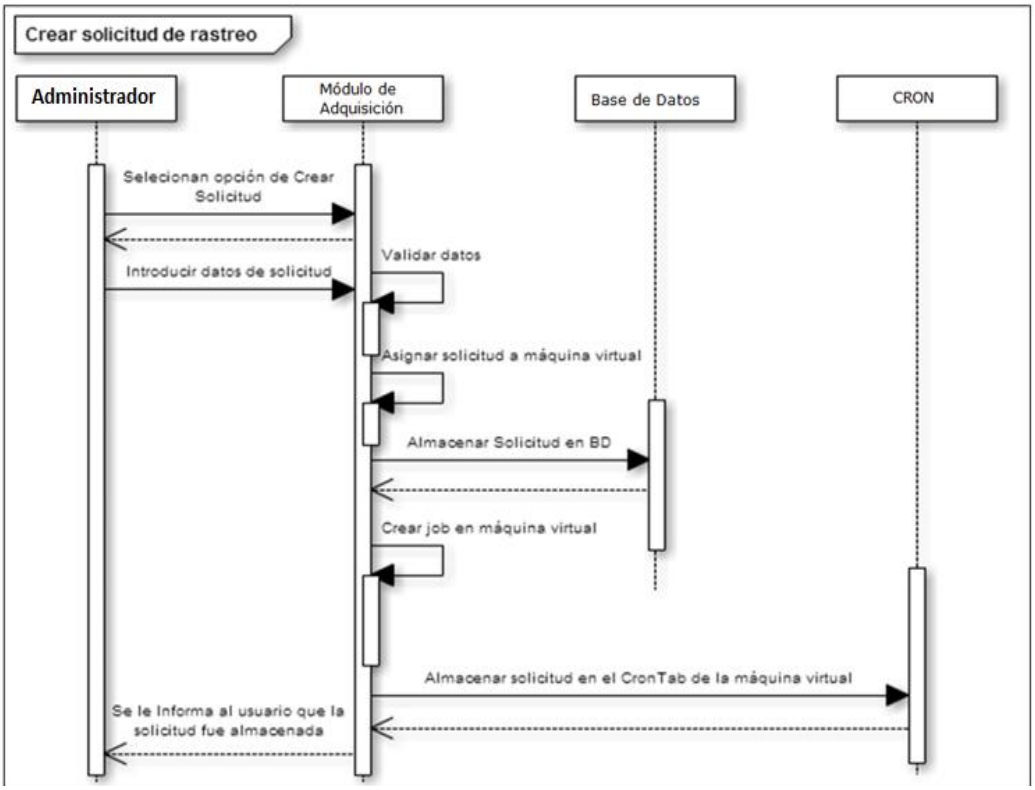


Figura 41. Diagrama de Secuencia Crear Solicitud de Rastreo Administrador

Tabla 12. Caso de Uso Consultar Rastreos

Nombre:	Consultar Rastreos
Autor:	Laura Sánchez
Número de CU	CU-6

Descripción:
Le permite al administrador visualizar los rastreos realizados
Actores:
Administrador
Pre-condiciones:
Flujo Normal:
1. El administrador ingresa a la aplicación
2. El administrador selecciona la opción de Consulta

Flujo Alternativo:
Post-condiciones:
El administrador visualiza la lista de rastreos

Tabla 13. Caso de Uso Visualizar Lista de Rastreos

Nombre:	Visualizar Lista de Rastreos
Autor:	Laura Sánchez
Número de CU	CU-6.1

Descripción:
Le permite al administrador visualizar los rastreos realizados, listando la información de todos los rastreos realizados por los suscriptores
Actores:
Administrador
Pre-condiciones:
Flujo Normal:
1. El administrador ingresa a la aplicación
2. El administrador selecciona la opción de Consulta
3. El sistema consulta en Base de Datos los rastreos generados por suscriptores
4. El sistema muestra las lista de rastreos generadas por suscriptores en forma tabular

Flujo Alternativo:
Post-condiciones:
Realizar gestión de los rastreos visualizados

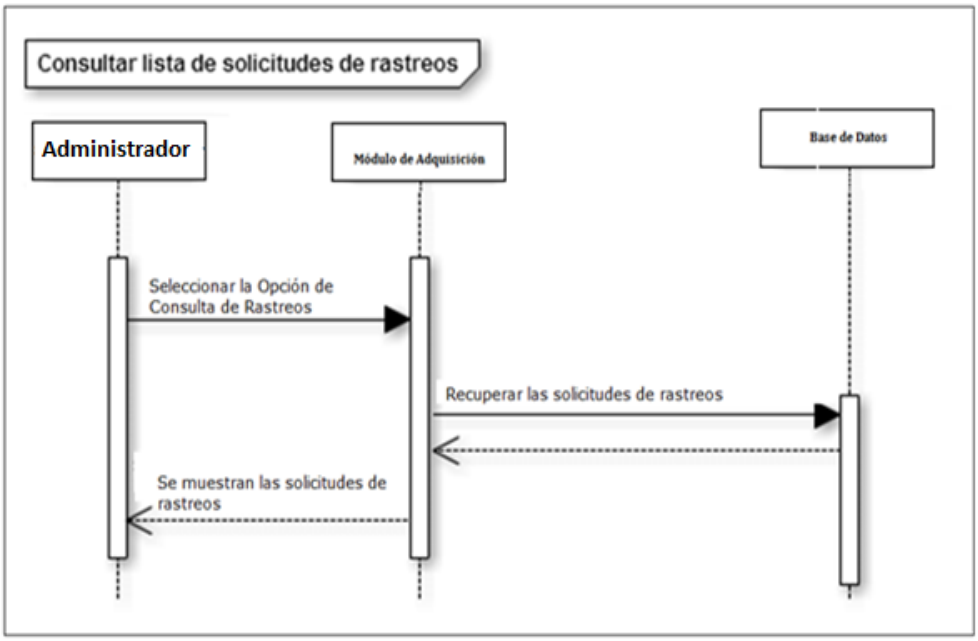


Figura 42. Diagrama de Secuencia Consultar Rastreos Administrador

Tabla 14. Caso de Uso Editar Solicitud de Rastreo

Nombre:	Editar Solicitud de Rastreo
Autor:	Laura Sánchez
Número de CU	CU-6.2

Descripción:
Permite al administrador editar una solicitud de rastreo
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador selecciona el rastreo a editar
2. Se cargan los parámetros del rastreo
3. El administrador modifica los parámetros del rastreo
4. Se almacenan en base de datos los nuevos parámetros
5. Se modifica en la máquina virtual respectiva el archivo de parámetros configuración del *job* asociado al rastreo
6. Si la frecuencia del rastreo fue cambiada , se edita la entrada en el *crontab* de la máquina virtual

Flujo Alternativo:
Post-condiciones:
Se ha editado la solicitud de rastreo

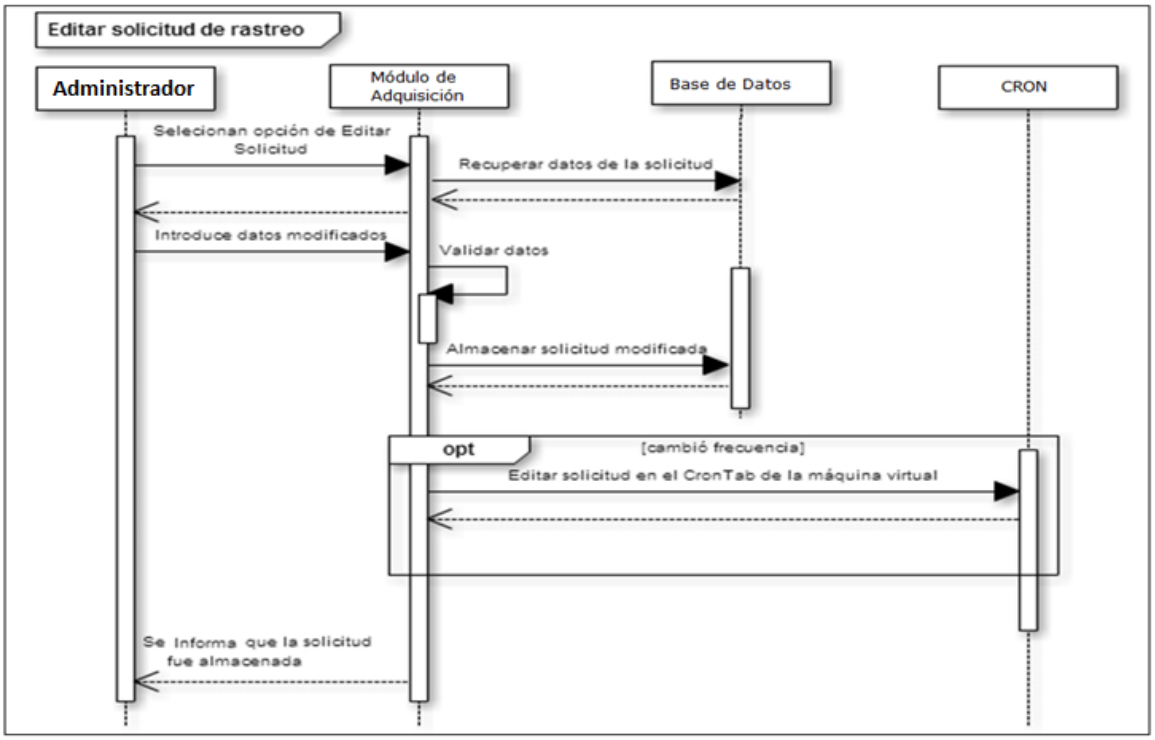


Figura 43. Diagrama de Secuencia Editar Solicitud de Rastreo Administrador

Tabla 15. Caso de Uso Desactivar Solicitud de Rastreo

Nombre:	Desactivar Solicitud de Rastreo
Autor:	Laura Sánchez
Número de CU	CU-6.3

Descripción:
Permite al administrador desactivar una solicitud de rastreo
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador selecciona la opción de desactivar solicitud de rastreo
2. Se le pregunta al administrador si está seguro de desactivar la solicitud de rastreo
3. Si la respuesta del administrador es afirmativa se marca en base de datos la solicitud como inactiva
4. Si la solicitud de rastreo es periódica, se elimina del administrador de tareas programadas la tarea asociada.

Flujo Alternativo:

Post-condiciones: Se le ha eliminado la solicitud de rastreo

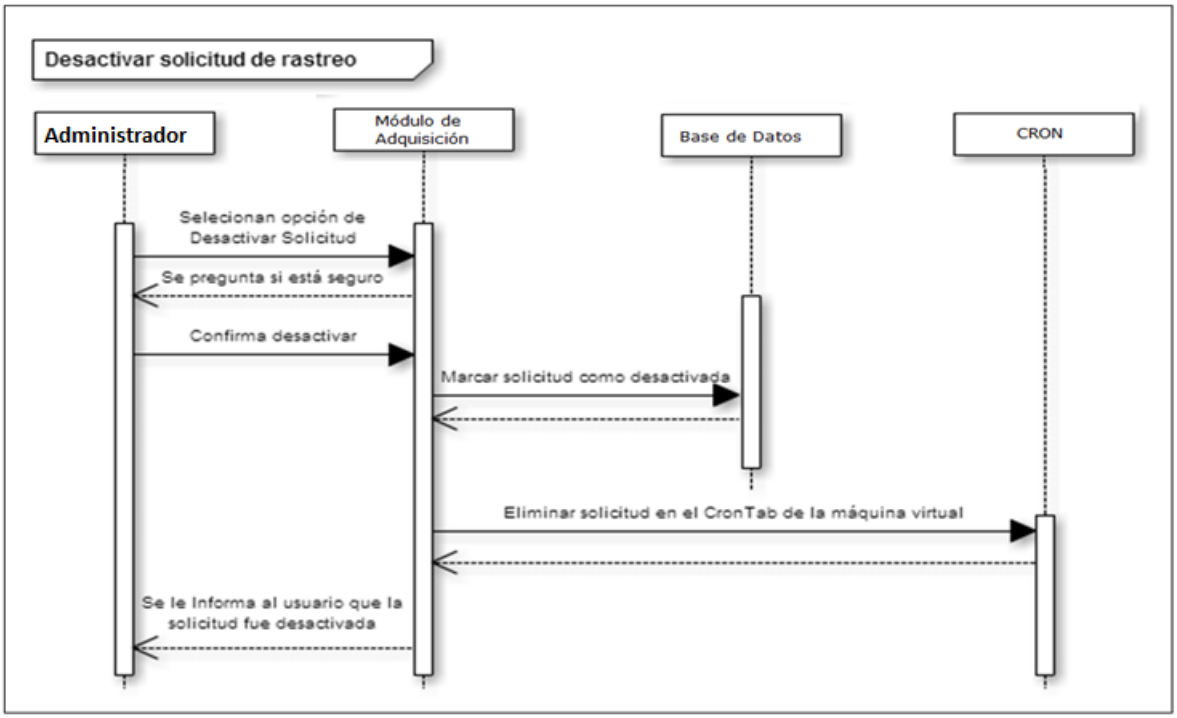


Figura 44. Diagrama de Secuencia Desactivar Solicitud de Rastreo Administrador

Tabla 16. Caso de Uso Consultar Versiones de Solicitud de Rastreo

Nombre:	Consultar Versiones de Solicitud de Rastreo
Autor:	Laura Sánchez
Número de CU	CU-6.4
Descripción: Consultar datos de un rastreo **Actores:** Administrador **Pre-condiciones:** **Flujo Normal:** 1. El administrador selecciona un rastreo para consultar sus versiones 2. Se leen de base de datos las versiones asociadas al rastreo 3. Se le muestran los datos al administrador de forma tabular **Flujo Alternativo:** **Post-condiciones:**	

Se han consultado las versiones de una solicitud de rastreo

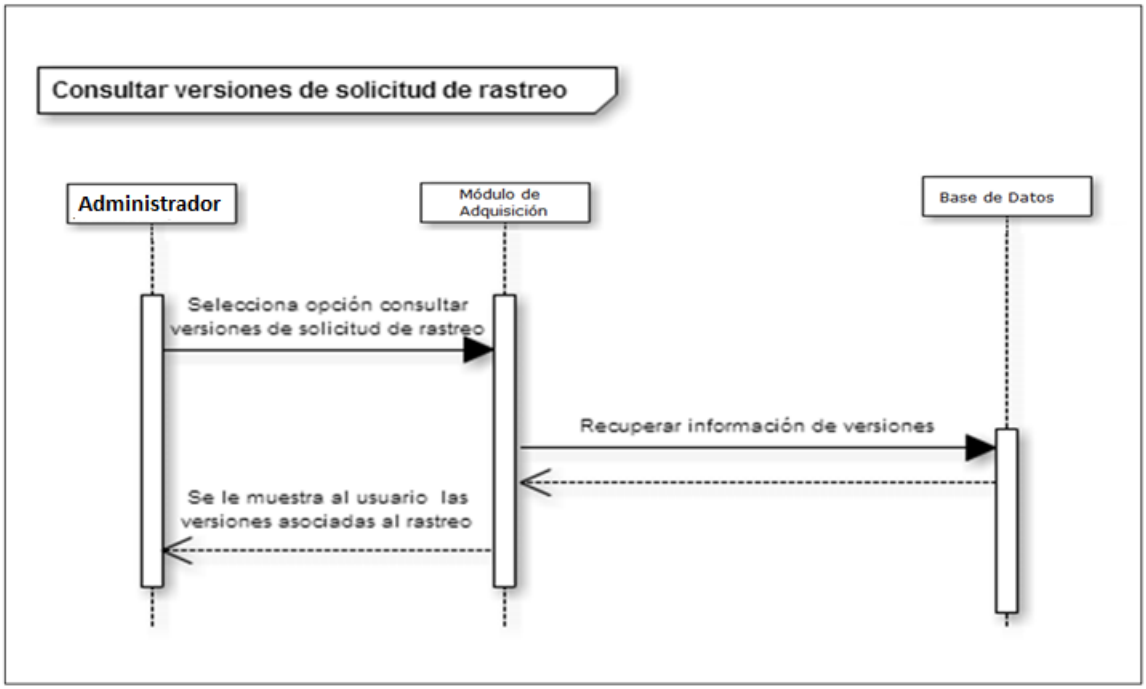


Figura 45. Diagrama de Secuencia Consultar Versiones de Solicitud de Rastreo Administrador

Tabla 17. Caso de Uso Generar Rastreo

Nombre:	Generar Rastreo
Autor:	Laura Sánchez
Número de CU	CU-7
Descripción: Ejecutar un rastreo programado **Actores:** Sistema(Script que realiza la llamada a la herramienta Heritrix) **Pre-condiciones:** **Flujo Normal:** 1. El *crontab* del sistema ejecuta un script para colocar el rastreo en ejecución 2. Se valida que el rastreo pudo ser colocado en ejecución 3. Se almacena en base de datos, los datos asociados a la versión del rastreo que se acaba de colocar en ejecución **Flujo Alternativo:** **Post-condiciones:**	

Se ha colocado en ejecución un rastreo programado

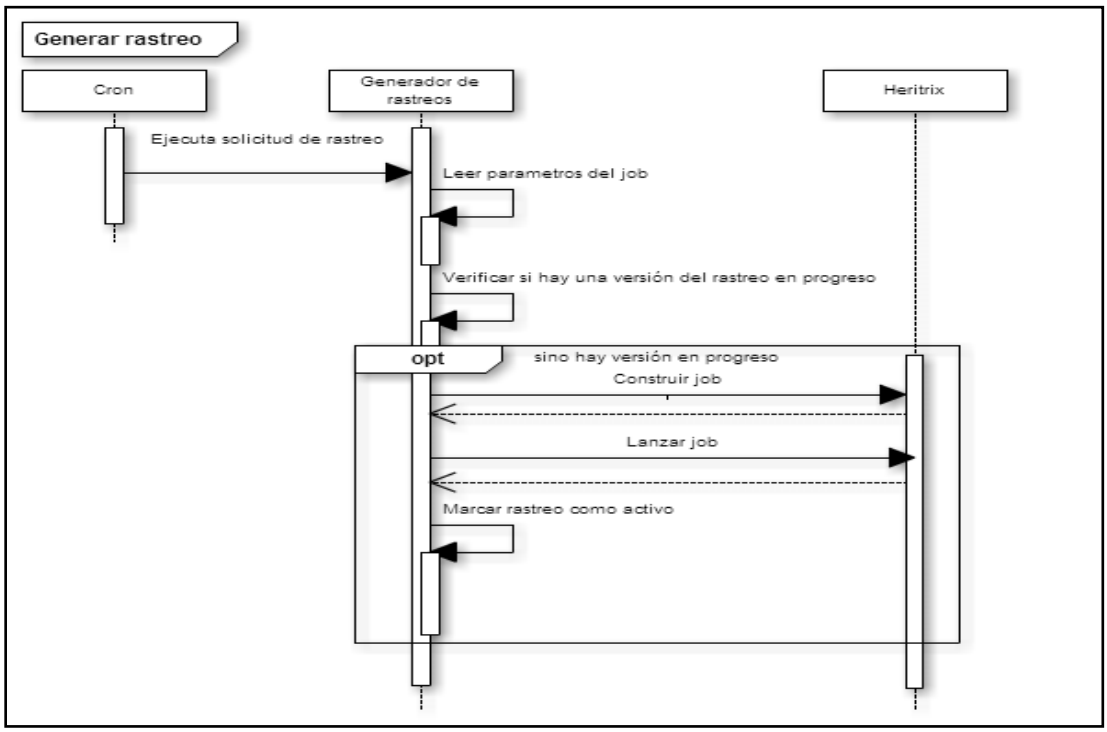


Figura 46. Diagrama de Secuencia Generar Rastreo

Tabla 18. Caso de Uso Verificar Fin de Rastreo

Nombre:	Verificar Fin de Rastreos
Autor:	Laura Sánchez
Número de CU	CU-8

Descripción:
Se verifica que rastreos han finalizado y se marcan para ser indexados
Actores:
Sistema(Script que realiza la llamada a la herramienta Heritrix)
Pre-condiciones:
Flujo Normal:

1. El sistema lee el archivo que contiene la lista de rastreos activos
2. Se recupera el archivo log asociado a un rastreo
3. Se *parsea* el archivo log para obtener el estatus en el que se encuentra el rastreo
4. Si el estatus es igual a "*finish*", se coloca una entrada en el archivo de rastreos por indexar y se borra del archivo de rastreos activos

5. Se repite el proceso desde el paso 2 hasta que toda la lista de rastreos en activos haya sido procesada

Flujo Alternativo:
Post-condiciones:
Se ha comprobado que rastreos han finalizado y se han marcado para ser indexados

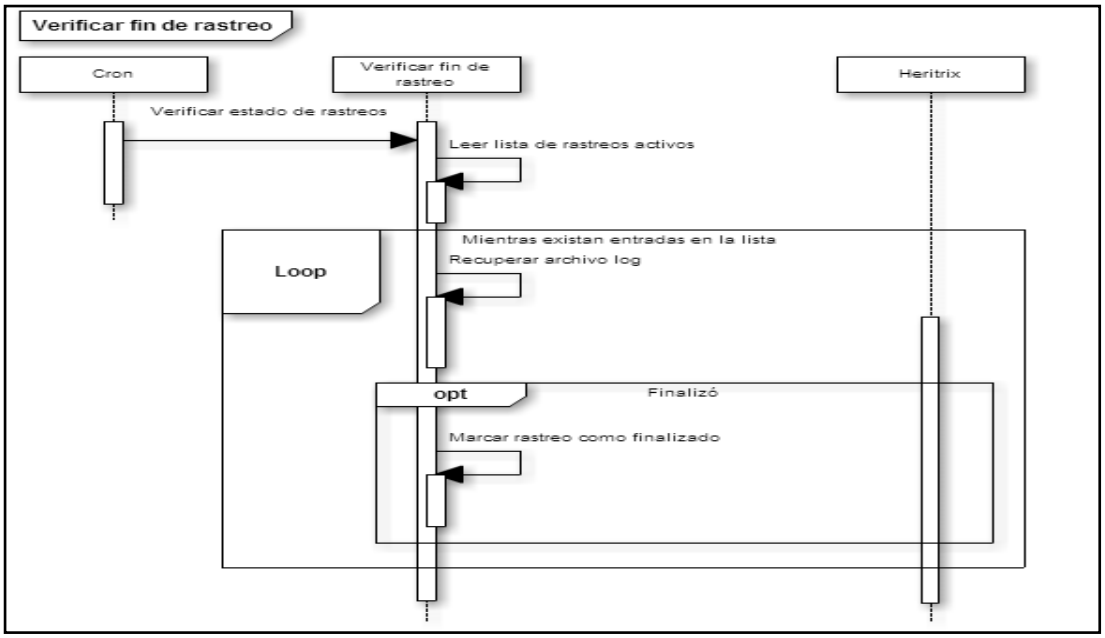


Figura 47. Diagrama de Secuencia Verificar Fin de Rastreo

Tabla 19. Caso de Uso Visualizar Rastreos Fallidos

Nombre:	Visualizar Rastreos Fallidos
Autor:	Gamar Milano
Número de CU	CU-9
Descripción: Permite al administrador visualizarla información de los rastreos fallidos **Actores:** Administrador **Pre-condiciones:** **Flujo Normal:**	

1. El administrador ingresa a la aplicación
2. El administrador selecciona la opción de Rastreos Fallidos
3. El sistema lee el archivo que contiene la lista de rastreos activos
4. El sistema lista de forma tabular los rastreos al administrador

Flujo Alternativo:

Post-condiciones:

Se listan los rastreos fallidos

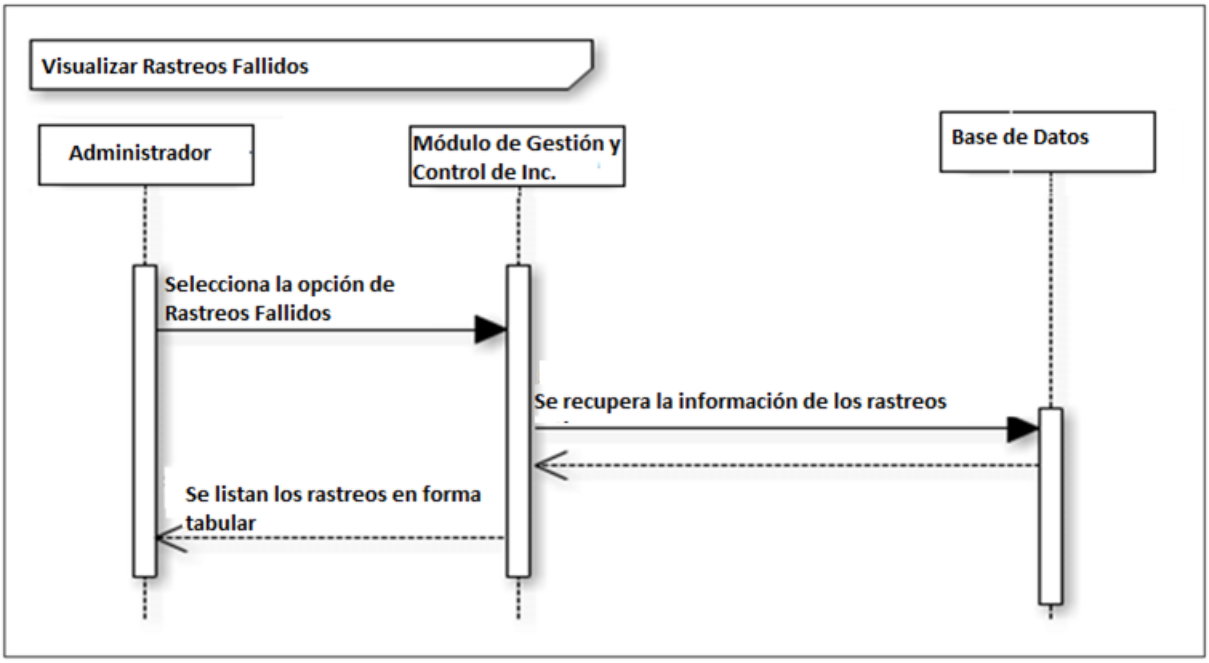


Figura 48. Diagrama de Secuencia Visualizar Rastreos Fallidos

Tabla 20. Caso de Uso Reiniciar Rastreo

Nombre:	Reiniciar Rastreo
Autor:	Gamar Milano
Número de CU	CU-9.1

Descripción:
Se reinicia un rastreo fallido
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador ingresa a la aplicación
2. El administrador selecciona la opción de Rastreos Fallidos
3. El sistema lee el archivo que contiene la lista de rastreos activos
4. El sistema lista de forma tabular los rastreos al administrador
5. El sistema ejecuta un script para colocar el rastreo en ejecución
6. Se valida que el rastreo pudo ser colocado en ejecución
7. Se almacena en base de datos, los datos asociados a la versión del rastreo que se acaba de colocar en ejecución

Flujo Alternativo:
Post-condiciones:
Se reinicia el rastreo fallido

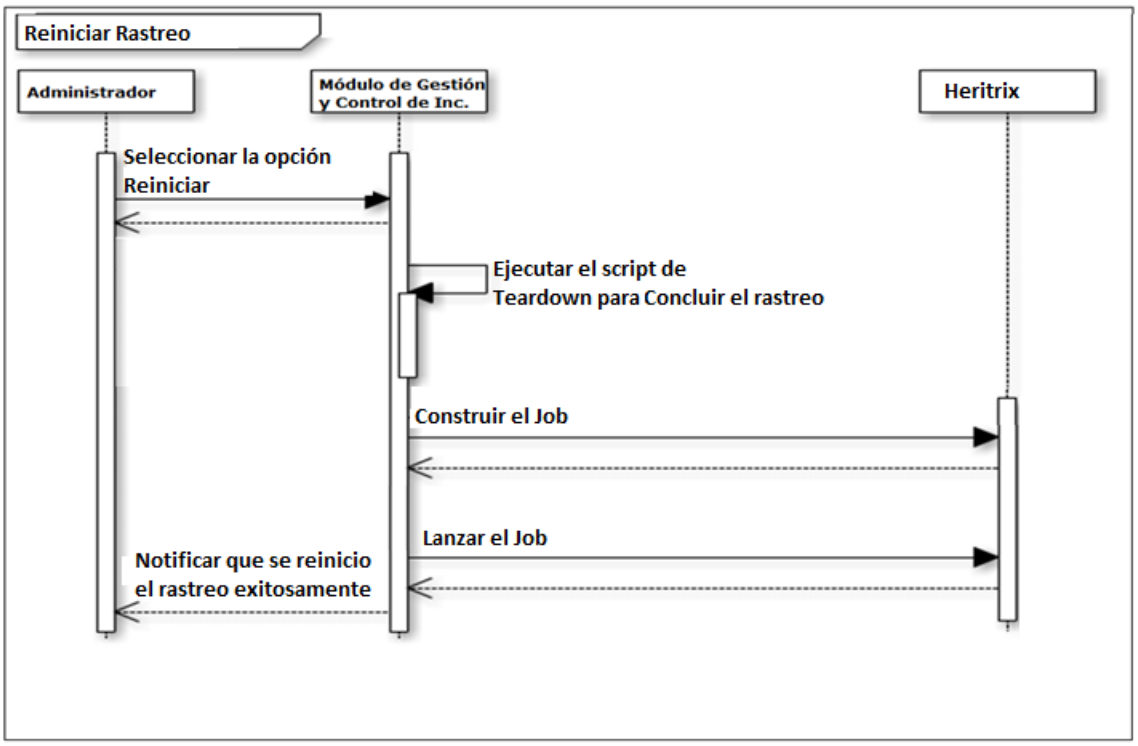


Figura 49. Diagrama de Secuencia Reiniciar Rastreo

Tabla 21. Caso de Uso Negar Acceso a URL ya registradas

Nombre:	Negar acceso a URL ya registradas
Autor:	Gamar Milano
Número de CU	CU-10

Descripción:
Se niega el registro de URL ya Registradas
Actores:
Sistema
Pre-condiciones:
Flujo Normal:

1. El suscriptor ingresa a la aplicación
2. El suscriptor selecciona la opción de crear rastreo
3. El suscriptor introduce los datos solicitados
4. El sistema valida los datos
5. Se le notifica al suscriptor que su solicitud no es permitido ya que se encuentra registrado el URL a rastrear

Flujo Alternativo:
Post-condiciones:
URL no registrado

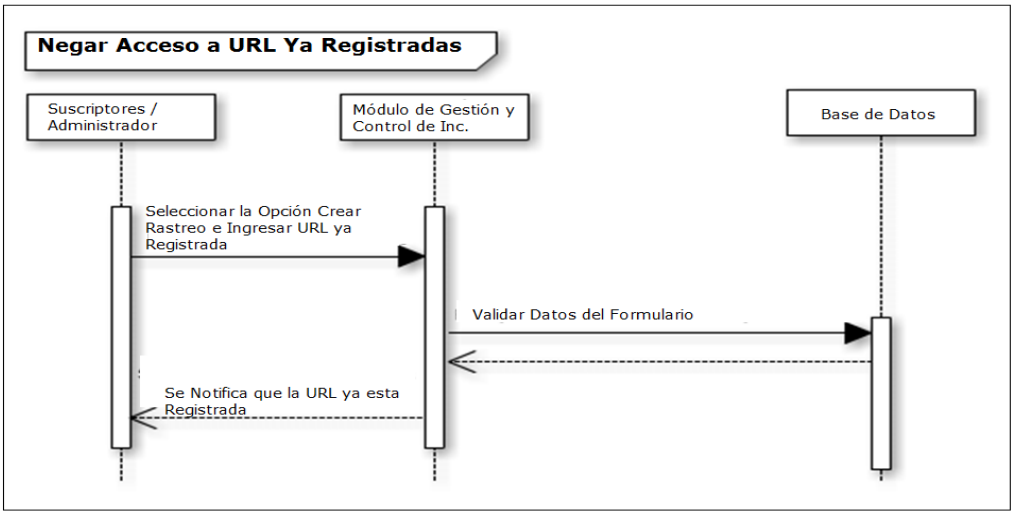


Figura 50. Diagrama de Secuencia Negar acceso a URL Ya Registrada

Tabla 22. Caso de Uso Crear Colección

Nombre:	Crear Colección
Autor:	Gamar Milano
Número de CU	CU-11

Descripción:
Permite crear una colección
Actores:
Administrador
Pre-condiciones:
CU-9
Flujo Normal:
1. El administrador selecciona la opción de crear colección
2. El administrador introduce la información de la colección
3. El sistema valida los datos
4. La colección es almacenada en base de datos

Flujo Alternativo:
Post-condiciones:
Colección creada, podrá ser seleccionada para crear/editar un rastreo.

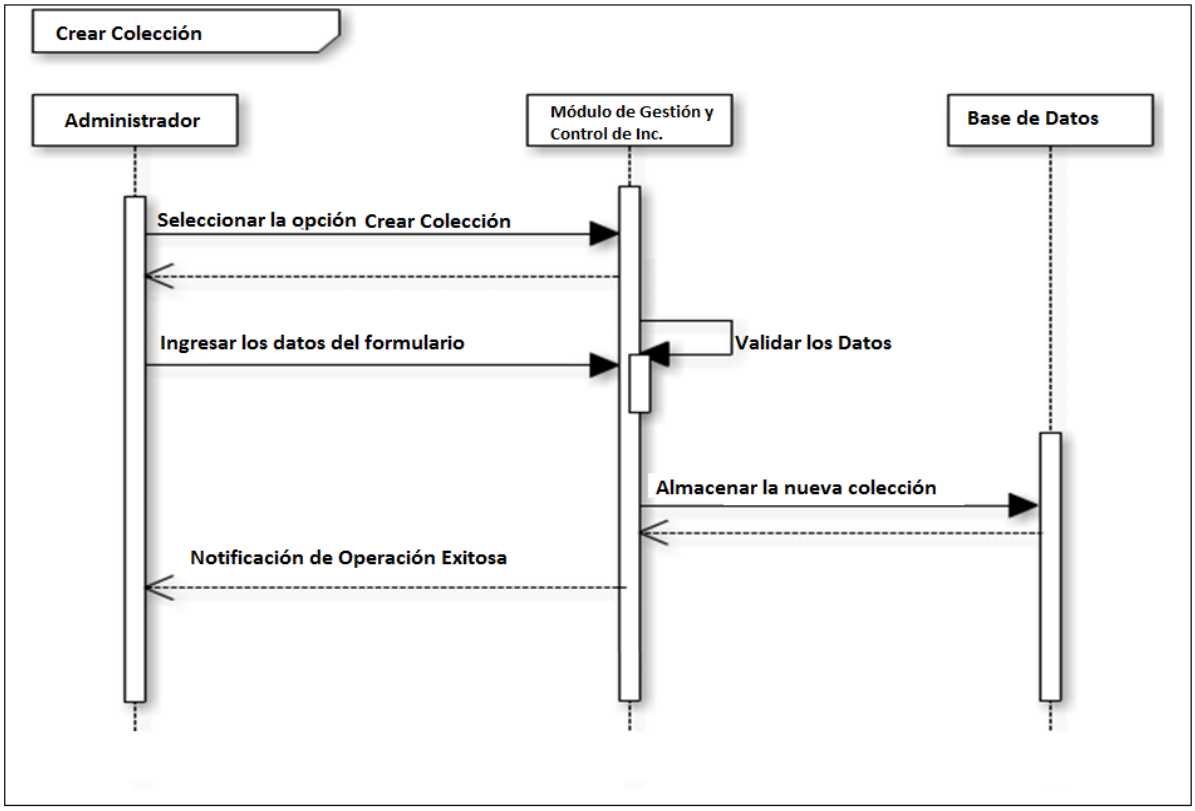


Figura 51. Diagrama de Secuencia Crear Colección

Tabla 23. Caso de Uso Deshabilitar Usuarios

Nombre:	Deshabilitar Usuarios
Autor:	Gamar Milano
Número de CU	CU-12

Descripción:
Le permite al administrador deshabilitar el acceso a un usuario registrado
Actores:
Administrador
Pre-condiciones:
Flujo Normal:

1. El administrador ingresa a la aplicación
2. El administrador ingresa en la opción Usuarios Registrados
3. El administrador visualiza la información de los usuarios registrados
4. El administrador deshabilita el acceso al usuario

Flujo Alternativo:
Post-condiciones:
No se le permite el acceso al usuario en la sistema

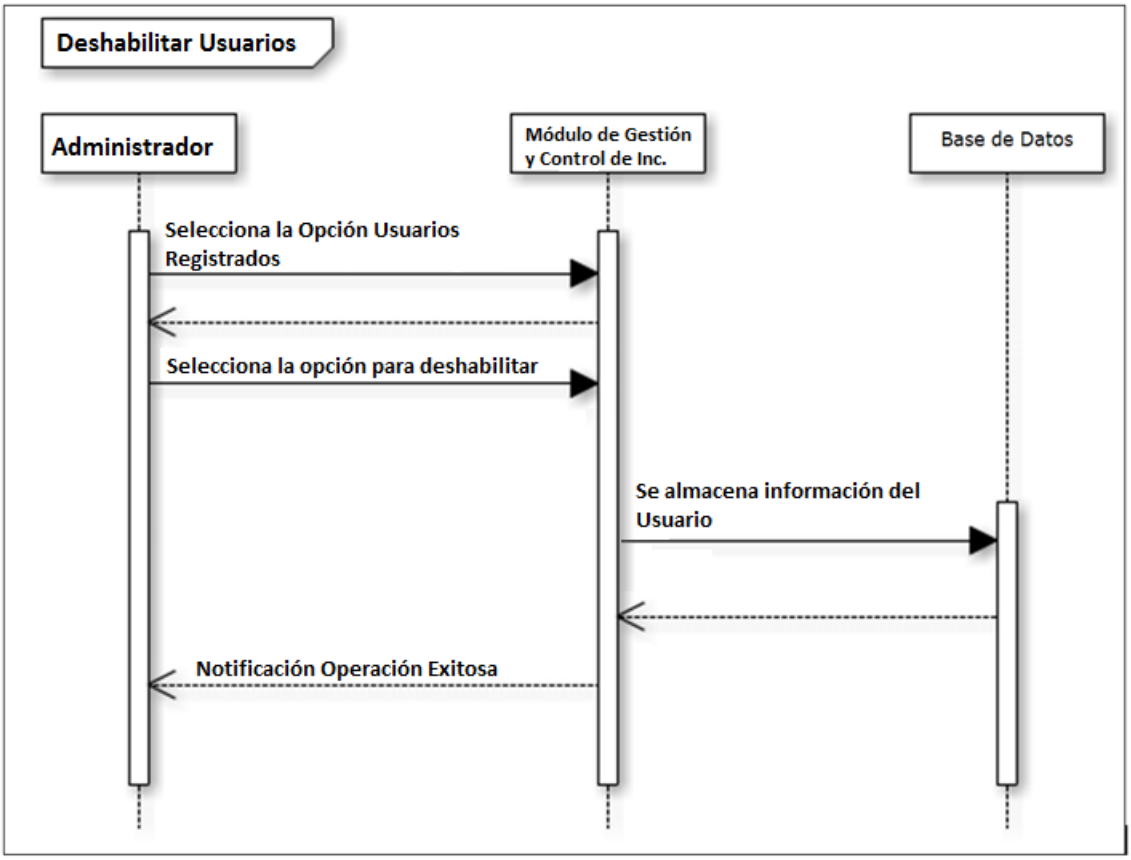


Figura 52. Diagrama de Secuencia Deshabilitar Usuarios

4.4.2.5 Modelo de datos

Fue modificada y actualizada la base de datos utilizada en el Módulo de Adquisición (García & Rivero, 2013), En esta base de datos cuyo modelo se puede observar en la figura 53 se mantuvieron:

- Los datos de las semillas o URLs de los sitios a preservar y los datos de configuración de los rastreos (app_traces),
- Los datos descriptivos de las versiones (app_versions),
- Los datos de las colecciones (app_coleccions).

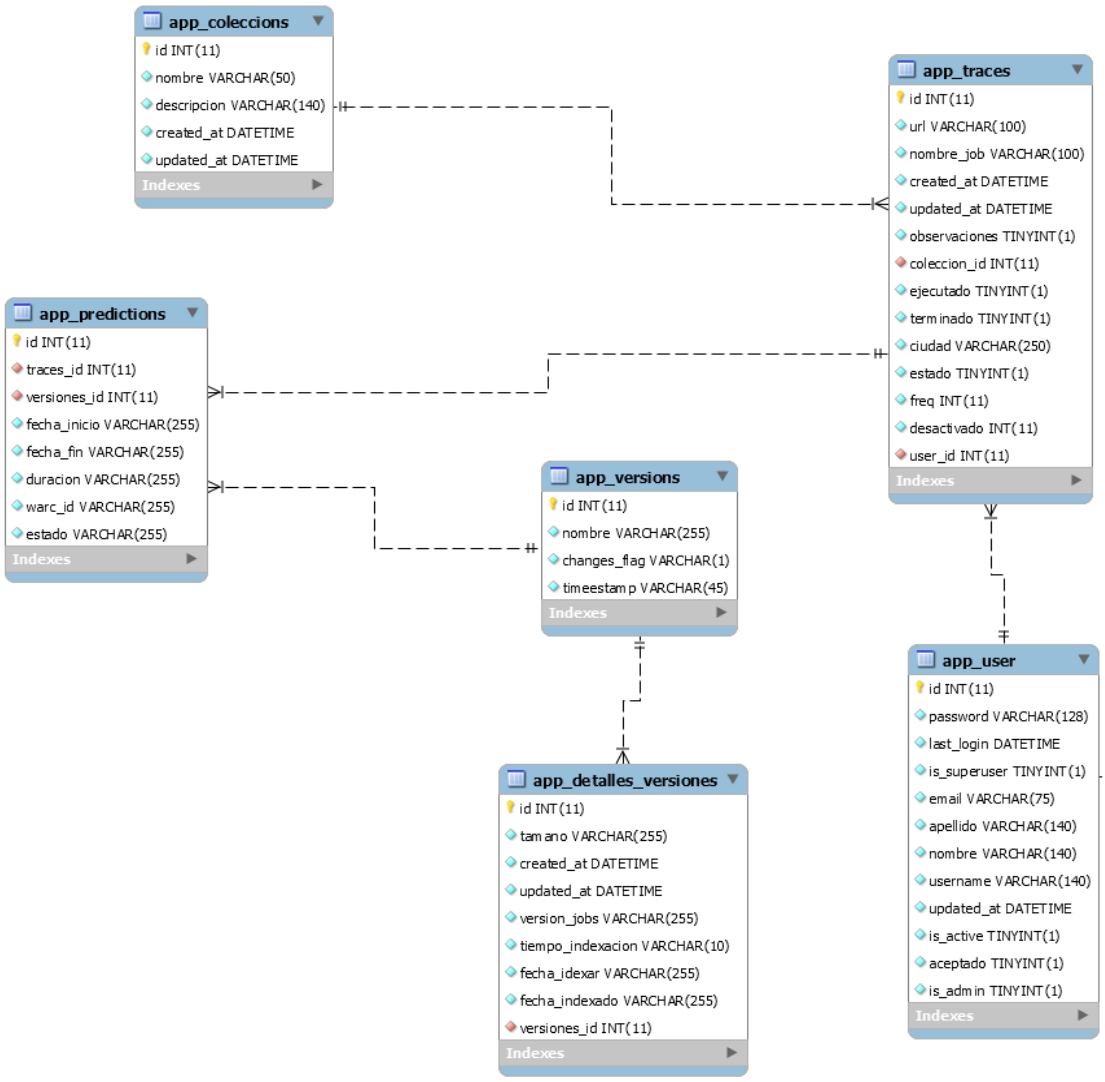


Figura 53. Modelo relacional de la base de datos utilizada en la aplicación

Fueron modificados:

- La tabla de usuarios para diferenciar los roles de administrador y suscriptor,
- Los datos de georeferencia de los URL fueron añadidos a la tabla app_traces.

Y fue agregada la tabla para manejar las predicciones de cambio (app_predictions) por solicitud del Módulo de Predicción de Rastreos actualmente en desarrollo.

4.4.3 Fase de Construcción

En este punto se explica cómo se llevó a cabo la implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web de Venezuela, adaptando el desarrollo a una arquitectura basada en componentes, el lenguaje utilizado.

4.4.3.1 Implementación de componentes

Para el desarrollo del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web de Venezuela se utilizaron software existentes tales como el rastreador Heritrix, la base de datos MySQL y el programador de tareas automáticas cron del sistema operativo Debian.

Adicionalmente se implementaron los componentes de Gestionar Solicitudes de Usuarios (ver tabla 24), Generador de Rastreo (ver tabla 25) y Reiniciar Rastreos (ver tabla 26), para cubrir todos los requerimientos que el Prototipo debe satisfacer, a continuación se explica cada uno de ellos.

Tabla 24. Componente Gestionar Solicitudes de Usuarios

Componente: Gestionar Solicitudes de Usuarios	
Interfaz que proporciona: • Http: se puede acceder a ella usando el protocolo http para que los usuarios realicen las solicitudes de registro, contacto con el administrador, adicionalmente permite que el usuario con rol administrador la gestión y supervisión de los usuarios tanto registrados como no registrados en la aplicación. Permite que los suscriptores realicen solicitudes de rastreo.	
Funcionalidades: Aplicación Web que permite: • Realizar registro de solicitudes de nuevos usuarios. • Envío de correos al administrador con información relevante para los usuarios.	**Entrada:** • Para realizar el registro de nuevos usuarios recibe la información del usuario a registrar. • Para el envío de correos al administrador recibe la información del mensaje a enviar,

Componente: Gestionar Solicitudes de Usuarios	
• Autorización o rechazo de ingreso a la aplicación de nuevos usuarios. • Habilitar o deshabilitar el acceso de la aplicación de usuarios registrados. • Suscribir nuevos sitios web. • Interactúa con el cron del sistema para programar trabajos de rastreo. • Crear colecciones de sitios web • Autorizar nuevos suscriptores. • Monitorear el estado de los rastreos, al usuario administrados se le habilita la opción para los reiniciar rastreos en caso de fallas.	el asunto del mensaje, el nombre y la dirección de correo. • Para suscribir sitios: recibe la URL, la colección y si el rastreo incluye subdominios. • Para crear colecciones recibe el nombre y descripción de la colección. **Salida:** • Permite el ingreso a la aplicación a los usuarios con acceso autorizado y habilitado. No permite el ingreso a los usuarios con acceso denegado o inhabilitado. • Se realiza el envío de correos al administrador. • Muestra los URL suscritos, sus datos de geolocalización y detalles de los rastreos realizados para cada URL. • Reinicia los rastreos que fueron seleccionados por el administrador.

Tabla 25. Componente Generador de Rastreos

Componente: Generador de Rastreos	
Interfaz que proporciona: • Usa la interfaz para proveer el servicio de generar la creación de un trabajo de rastreo.	
Función: • Se conecta con el rastreador para que se lleven a cabo los rastreos programados asociados a una solicitud de rastreos creada por un suscriptor, actualizando lista de rastreos activos y el estado de los rastreos en la base de datos.	**Entrada:** Recibe los parámetros del sitio web a rastrear. **Salida:** • Envía los parámetros a Heritrix para generar los nuevos rastreos. • Actualiza la lista de rastreos activos en el archivo y la base de datos.

Tabla 26. Componente Reiniciar Rastreo

Componente: Reiniciar Rastreos	
Interfaz que proporciona: • Usa la interfaz para proveer el servicio de generar la creación de un trabajo de rastreo.	
Función: • Se conecta con el rastreador para que se lleven a cabo el reinicio de los rastreos que fueron seleccionados por el administrador, actualizando lista de rastreos activos y el estado de los rastreos en la base de datos.	**Entrada:** Recibe los parámetros del rastreo a reiniciar. **Salida:** • Envía los parámetros a Heritrix para generar el reinicio del rastreo. • Actualiza la lista de rastreos activos en el archivo y la base de datos.

Módulo de Gestión y Control de Incidencias.

Se desarrolló la modificación de la interfaz para realizar la creación y configuración de rastreos, consultar su estatus presentes en el Módulo de Adquisición y Almacenamiento (García & Rivero, 2013).

Esta modificación de la aplicación fue implementada utilizando el lenguaje Python y su framework Django, para la persistencia de datos se utilizó la base de datos MySQL.

La aplicación cuenta con autenticación y registro de usuarios, en la Figura 55 se visualiza la pantalla de inicio donde se aprecia el formulario para la autenticación en la aplicación. En la Figura 54 se puede apreciar el formulario de registro de usuarios no registrados, el cual es validado y notifica que la solicitud debe ser aprobada por el administrador para permitir el ingreso.

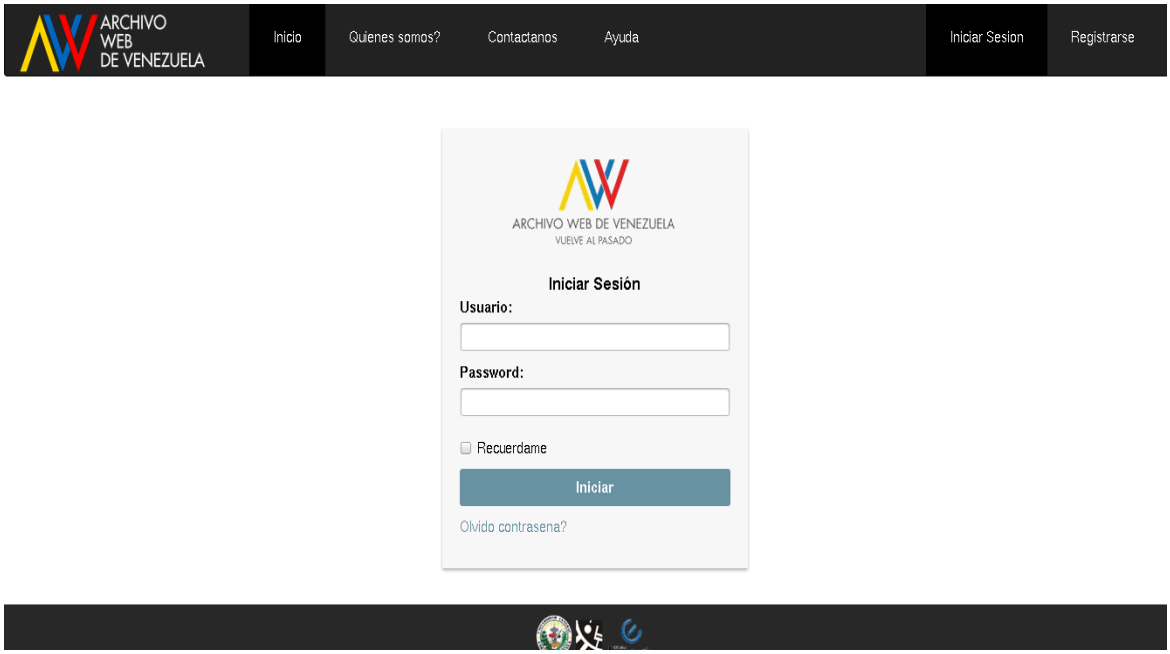


Figura 54. Formulario de Autenticación de Usuarios

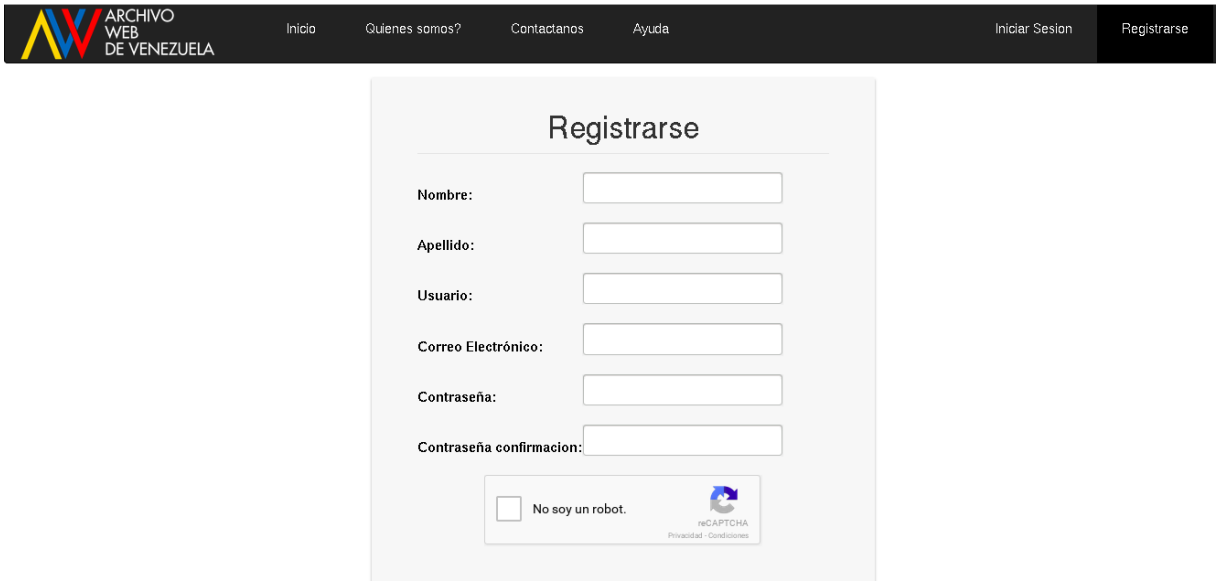


Figura 55. Formulario de Registro de Usuarios

Autorizar o Denegar el Acceso de Nuevos Usuarios

En el momento de que el usuario no registrado realice una solicitud de registro, al administrador se le listara la información ingresada por el usuario para que sea autorizado o negado el acceso a la aplicación, esto se puede apreciar en la Figura 56.

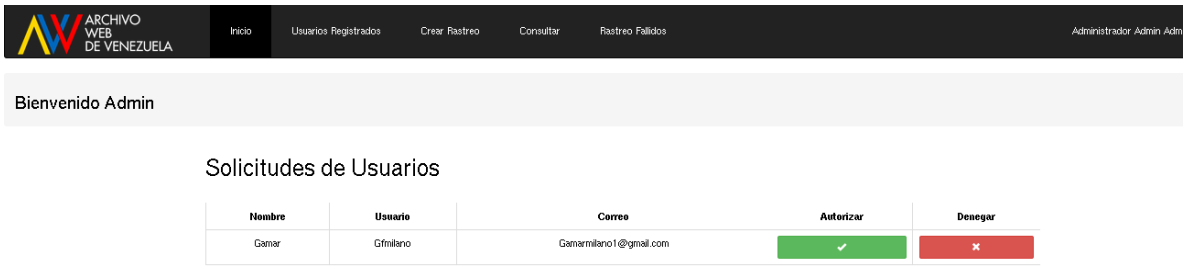


Figura 56. Opción de Solicitudes de Usuarios

Las opciones para gestionar los rastreos se describen a continuación:

Crear Rastreo, Editar Rastreo

Para la creación son solicitados los datos que servirán para la solicitud del rastreo, el suscriptor debe introducir los mismos, esto se puede apreciar en la Figura 57.

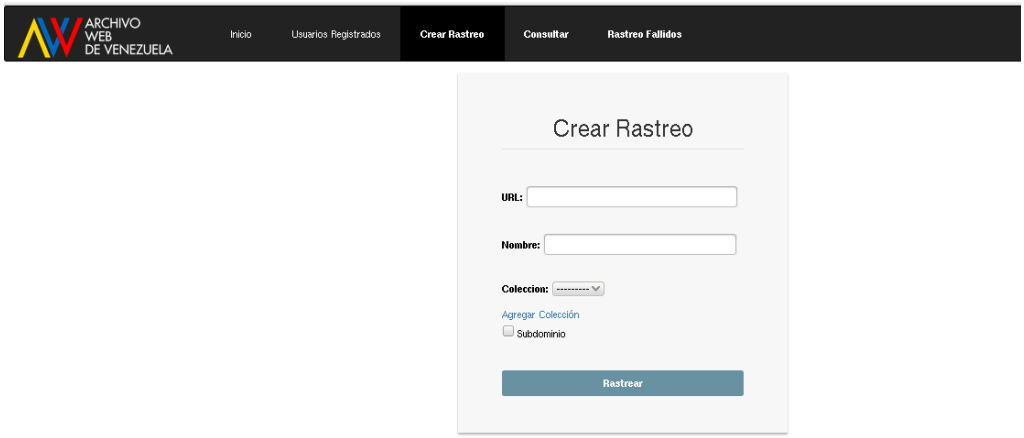


Figura 57. Formulario para Creación de Rastreos

En caso de que el usuario quiera editar un rastreo, en el formulario se cargan los datos asociados, tal como se puede apreciar en la Figura 58, los cuales son modificados según lo ingresado por el suscriptor.

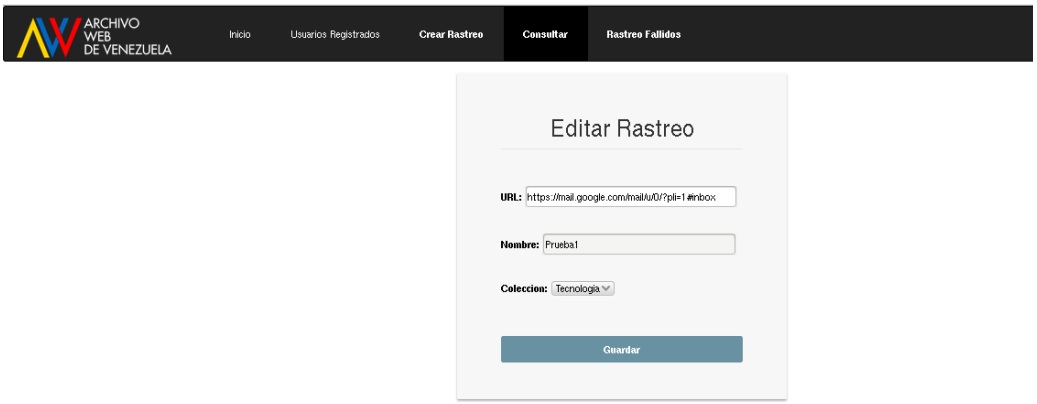


Figura 58. Formulario para la Edición de Rastreos

Luego de que los datos son ingresados se procede a validar los siguientes puntos:

- El URL no debe estar registrado con anterioridad.
- El nombre del rastreo no debe estar registrado con anterioridad.
- No se deben dejar campos en blanco.
- Solo deben ser ingresados URL de Venezuela, debe poseer la extensión .ve
- El URL debe estar bien formado. Para realizar los rastreos Heritrix solicita que sea una URL con estructura valida, ejemplo: http://ejemplo.ejemplo.

Al culminar de validar los datos ingresados se procede a seleccionar la máquina virtual donde residirá el *job* asociado al rastreo, la selección se hace utilizando turno rotatorio.

Seguidamente se edita el archivo de configuración del *job* crawler-beans.cxml, como se mencionó en las características de Heritrix sus archivos de configuración están basados en *beans* que tienen estructura XML. Fueron configurados los siguientes puntos en el archivo del *job* crawler-beans.cxml:

- Nombre del archivo donde estará el URL ingresada por el usuario para realizar el rastreo, tal y como se ilustra en la figura 59.

```
<!-- SEEDS ALTERNATE APPROACH: specifying external seeds.txt file in
     the job directory, similar to the H1 approach.
     Use either the above, or this, but not both. -->

<bean id="seeds" class="org.archive.modules.seeds.TextSeedModule">
 <property name="textSource">
  <bean class="org.archive.spring.ConfigFile">
   <property name="path" value="seeds.txt" />
  </bean>
 </property>
 <property name='sourceTagSeeds' value='false'/>
 <property name='blockAwaitingSeedLines' value='-1'/>
</bean>
```

Figura 59. Archivo crawler-beans.cxml Configuración URL a Rastrear

- Tamaño de los archivos WARC, configurados en un 1G información obtenida de (The WARC File Format, 2007), esto se puede apreciar en la figura 60.

```
<!-- DISPOSITION CHAIN -->
<!-- first, processors are declared as top-level named beans  -->
<bean id="warcWriter" class="org.archive.modules.writer.WARCWriterProcessor">
 <!-- <property name="compress" value="true" /> -->
 <!-- <property name="prefix" value="IAH" /> -->
 <!-- <property name="suffix" value="${HOSTNAME}" /> -->
 <property name="maxFileSizeBytes" value="1073741824" />
 <property name="poolMaxActive" value="5" />
 <!-- <property name="MaxWaitForIdleMs" value="500" /> -->
 <!-- <property name="skipIdenticalDigests" value="false" /> -->
 <!-- <property name="maxTotalBytesToWrite" value="0" /> -->
 <!-- <property name="directory" value="${launchId}" /> -->
 <!-- <property name="storePaths">
```

Figura 60. Archivo crawler-beans.cxml Configuración Archivos Warc

- Configuración para permitir rastreos en paralelo (Ver Figura 61).

```
<bean id="queueAssignmentPolicy"
  class="org.archive.crawler.frontier.SurtAuthorityQueueAssignmentPolicy">
 <property name="forceQueueAssignment" value="" />
 <property name="deferToPrevious" value="true" />
 <property name="parallelQueues" value="1" />
</bean>
```

Figura 61. Archivo crawler-beans.cxml Configuración Rastreos en Paralelo

- Rechazar los rastreos cuyas extensiones sean (Ver Figura 62):
    - Rar, zip y tar.

- avi, wmv y mpg.
- doc, xls y odd.
- xml
- txt, conf y pdf.
- Swf.
- Js.
- bmp, gif, jpg, png, svg, y tiff

Esta configuración realizada para evitar la descarga de archives en el sistema.

```
<!-- ...and REJECT those from a configurable (initially empty) set of URI regexes... -->
<bean class="org.archive.modules.deciderules.MatchesListRegexDecideRule">
      <property name="decision" value="REJECT"/>
 <property name="listLogicalOr" value="true" />
      <property name="regexList">
   <list>
    <value>.*(?i)(\.(rar|zip|tar))$</value>
         <value>.*(?i)(\.(avi|wmv|mpe?g))$</value>
         <value>.*(?i)(\.(rar|zip|tar))$</value>
         <value>.*(?i)(\.(doc|xls|odd))$</value>
         <value>.*(?i)(\.(xml))$</value>
         <value>.*(?i)(\.(txt|conf|pdf))$</value>
         <value>.*(?i)(\.(swf))$</value>
         <value>.*(?i)(\.(js))$</value>
         <value>.*(?i)(\.(bmp|gif|jpe?g|png|svg|tiff?))$</value>
       </list>
      </property>
</bean>
```

Figura 62. Archivo crawler-beans.cxml Configuración Rechazo de URL con Extensiones Definidas

Seguidamente se ejecuta el script agregar_cron.sh para crear una entrada en el *crontab* de la máquina virtual seleccionada, tal como se puede ver en la Figura 63.

```
#agregar solicitud de rastreo al crontab de la máquina remota
_accion_job = "ssh -n root@"+@maquina+" "+"sh /home/agregar_cron.sh "+time1.min.to_s()+" "
+time1.hour.to_s()+" "+"'*/"+@rastreo.frecuencia+"'"+" "+@rastreo.nombre_job
xml = %x[ #{_accion_job} ]
```

Figura 63. Agregar Solicitud del Rastreo al Crontab

Desactivar Rastreo

Esta opción permite desactivar una solicitud de rastreo, tal como se ilustra en la Figura 64, su implementación consiste en eliminar del *crontab* de la máquina virtual donde fue asignado su entrada correspondiente, los rastreos que son desactivados por el Administrador no pueden ser activados por el Suscriptor, esto se realiza en el momento de la supervisión de los rastreos y observar que el rastreo presenta algún inconveniente. El rastreo puede ser desactivado tanto por el administrador como por el usuario.

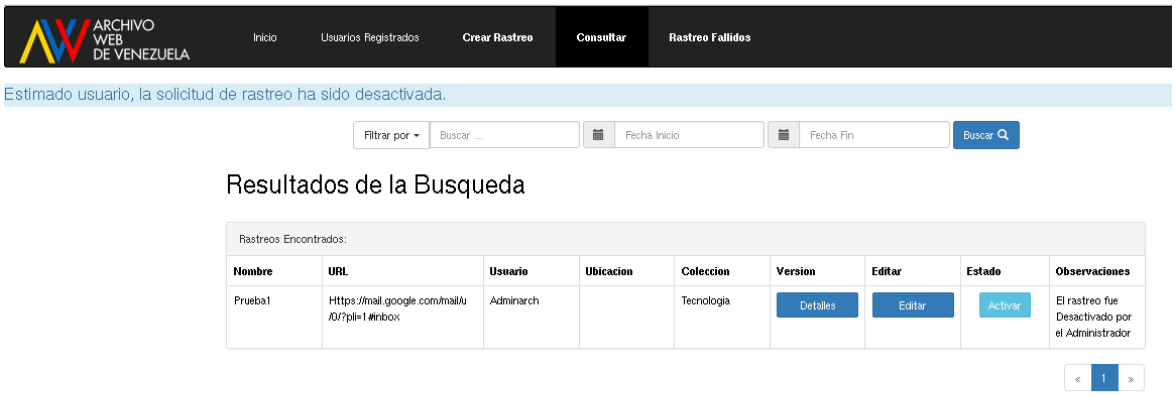


Figura 64. Opción Desactivar Rastreos

Visualizar Versiones de un Rastreo

Esta opción permite visualizar de forma tabular todas las versiones pertenecientes a una solicitud de rastreo, tal y como se observa en la Figura 65.

Figura 65. Opción Visualizar Versiones de un Rastreo

Generador de Rastreos

Este componente fue implementado utilizando el lenguaje *shell script*, en él se declaran los comandos necesarios para interactuar con el *Api Rest* de Heritrix con el fin de colocar en ejecución un rastreo programado.

```
#!/bin/sh

url="https://localhost:8443/engine/job/"
curl -d "action=build" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
curl -d "action=launch" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
curl -d "action=unpause" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
```

Figura 66. Generar Rastreo

En la Figura 66 se pueden apreciar los comandos necesarios para colocar un rastreo en ejecución:

Con el comando *build* se lee el archivo de configuración y se crean todas las instancias necesarias y el entorno para que el rastreo pueda ser colocado en ejecución

Con el comando *launch* se le da la orden a Heritrix de que cree las instancias necesarias para crear una nueva versión del rastreo.

Con el comando *unpause* el rastreo se coloca en progreso (por defecto Heritrix 'lanza los rastreos con estatus de *pause'*).

Antes de correr estos comando se valida que no exista una versión del *job* en ejecución, ya que solo puede estar en progreso una versión del job en un momento dado, si se encuentra una versión en progreso el script finaliza.

Adicionalmente a las llamadas al *Api Rest* de heritrix , se hace conexión con la base de datos MySQL para insertar los datos correspondiente a la versión del rastreo que se acaba de colocar en ejecución, en la Figura 67 se puede apreciar el código que invoca al script que se encarga de hacer la inserción en base de datos.

```
9   sh /home/insert_databse.sh $1
```

Figura 67. Invocación de script para Inserción en Base de Datos

El código del script insert_databse.sh se puede apreciar en la Figura 67:

```
#buscar el id que identica al rastreo
id_rastreo="$(ssh -n solr@190.169.69.154 sh /home/select_id.sh "$1")"

ssh -n solr@190.169.69.154 sh /home/insertar_version.sh $1 $2 null en_progreso desconocido desconocido $id_rastreo
```

Figura 68. Script insert_database.sh

Verificador de Rastreos Finalizados

Este componente fue implementado utilizando el lenguaje *shell script*, es ejecutado periódicamente por el *crontab* del sistema, la periodicidad de ejecución se decidió de que fuese de 2 veces por día.

Internamente se lee un archivo llamado <rastreos_activos>.txt que contiene una lista de los rastreos que se encuentran en progreso, en la Figura 69 se puede ver un ejemplo de cómo son las entradas de este archivo.

```
/data/heritrix-3.2.0/jobs/ucv-ve/20150410095241
/data/heritrix-3.2.0/jobs/bcv/20150410110025
/data/heritrix-3.2.0/jobs/usb/20150410112536
/data/heritrix-3.2.0/jobs/santamaria/20150410121540
```

Figura 69. Ejemplo de Archivo rastreos_activos.txt

Para verificar en qué estatus se encuentra un rastreo, se revisa en el directorio correspondiente a la versión del *job* si existe el directorio *reports*, si este directorio existe quiere decir que el rastreo finalizó y debe ser indexado, seguidamente el rastreo se marca como por indexar, es decir, se agrega una entrada a la lista de rastreos por indexar en el archivo <ip_maquina>.txt, en la Figura 70 se puede visualizar parte del código que implementa esta funcionalidad.

```
#!/bin/sh

FILE='/home/rastreos_activos.txt'
HERITRIX='/data/heritrix-3.1.1/jobs/'
if [ -f $FILE ];
    then
        cat $FILE | while read LINE
        do
            ruta=`echo $LINE | cut -d ' ' -f 1`

            # Si existe la carpeta reports quiere decir que el rastreo finalizó
            if  test -e $ruta"/reports" ;
                then
                    #Variables auxiliares para armar la expresión regular para eliminar la entrada de rastreos_activos.txt
                    aux="/"
                    aux5="/d"
                    sed -i $aux`echo $ruta | cut -d '/' -f 5`$aux5 /home/rastreos_activos.txt
                    echo $LINE >> 190.169.69.155.txt

                    #Se obtiene la fecha en que finalizó el rastreo
                    log="job.log"
                    LOG_FILE=$LINE"/"$log
                    echo $LOG_FILE
                    if [ -f $LOG_FILE ];
                        then
                            cat $LOG_FILE | while read LINE
                            do
                                ruta=`echo $LINE | cut -d ' ' -f 3`

                                if [ $ruta = "FINISHED" ];
                                    then

                                        fecha=`echo $LINE | cut -d ' ' -f 4`
                                        ano=`echo $fecha | cut -c 1-4`
                                        mes=`echo $fecha | cut -c 5-6`
                                        dia=`echo $fecha | cut -c 7-8`
                                        hora=`echo $fecha | cut -c 9-10`
                                        minuto=`echo $fecha | cut -c 11-12`
                                        segundo=`echo $fecha | cut -c 13-14`
                                        fecha_final=$dia"/"$mes"/"$ano"-"$hora":"$minuto":"$segundo
                                        echo $fecha_final
```

Figura 70.Verificados de Rastreos Finalizados

Reiniciar Rastreos Fallidos

Al sistema recuperarse de una falla, ocasionada por inconvenientes con el servicio eléctrico y el servidor donde está alojada la aplicación y las máquinas virtuales con los rastreos, se detuvo, el administrador ingresara a la aplicación en la opción rastreos fallidos y reiniciara los rastreos que se encontraban activos para ese momento en que ocurrió la falla, tal y como se observa en la Figura 71.

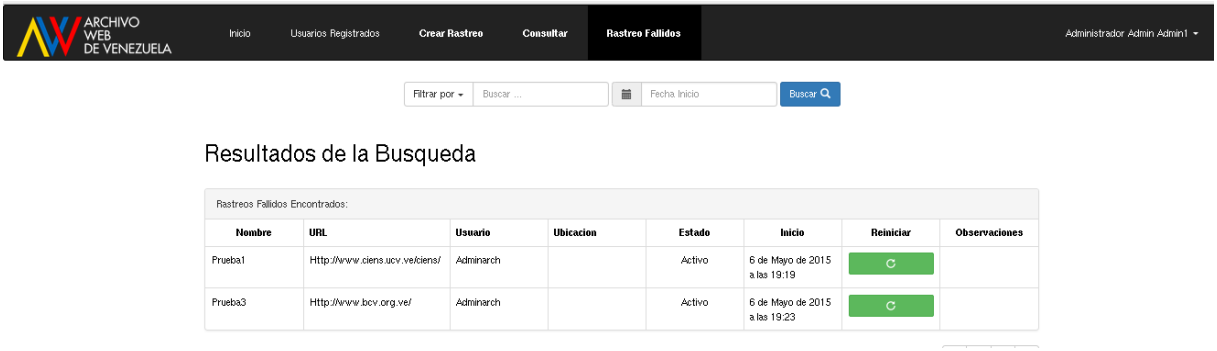


Figura 71. Opción Reiniciar Rastreos Fallidos

Para ello se ejecuta el script de reinicio_job.sh, donde se finaliza el rastreo que se estaba realizando para el momento en que ocurrió la falla, posteriormente se reinicia la generación del rastreo, realizando la construcción, el lanzamiento y ejecución del rastreo, tal y como se muestra en la figura 72.

```
#!/bin/sh

url="https://localhost:8443/engine/job/"
curl -d "action=teardown" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
curl -d "action=build" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
curl -d "action=launch" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
curl -d "action=unpause" -k -u AdminBD:heritrix --anyauth -H "Accept: application/xml" --location $url$1
```

Figura 72. Script reiniciar_rastreo.sh

Contactar al Administrador

Al sistema permite que tanto los usuarios no registrados como los suscriptores de la aplicación puedan enviar un correo electrónico al administrador, con información que consideren de interés, algún reclamo, sugerencia o solicitud de alguna información, para ello se ingresan los datos solicitados en el formulario ilustrado en la Figura 73.

Figura 73. Formulario de la Opción Contáctanos

4.4.4 Fase de Transición

En esta sección se explicará cómo fue realizada la transición desde el entorno de desarrollo hacia el entorno de producción, se explicará qué pruebas se realizaron para validar el Módulo de Gestión y Control de Incidencias.

4.4.4.1 Puesta en producción del Prototipo de Preservación Web

En este punto se hace la transición desde el entorno de desarrollo hacia el entorno de producción, a continuación se listan las máquinas virtuales y se menciona que componentes del sistema fueron alojados en cada una de ellas:

- **Solr**: En esta máquina se encuentra la aplicación con los diferentes Módulos desarrollados, Módulo de Acceso, Módulo de Adquisición y Almacenamiento y el Módulo de Gestión y Control de Incidencias. Heritrix, generador de rastreos, verificador de rastreos finalizados y herramientas WARCTools.
- **Heritrix1**: Heritrix, generador de rastreos, verificador de rastreos finalizados y herramientas WARCTools.
- **Heritrix2**: Heritrix, generador de rastreos, verificador de rastreos finalizados y herramientas WARCTools.
- **Heritrix3**: Heritrix, generador de rastreos, verificador de rastreos finalizados y herramientas WARCTools.

4.4.4.2 Pruebas Funcionales

En esta sección se muestran las pruebas realizadas para verificar el comportamiento del Prototipo y el cumplimiento de los requisitos funcionales del Módulo de Gestión y Control de Incidencias.

Lista de sitios Web a cosechar

Para hacer pruebas en el Módulo de Gestión y Control de Incidencias se decidió elaborar una lista de sitios web a cosechar, estos sitios son pertenecientes a universidades venezolanas así como a instituciones culturales y/o investigativas, la lista es la siguiente:

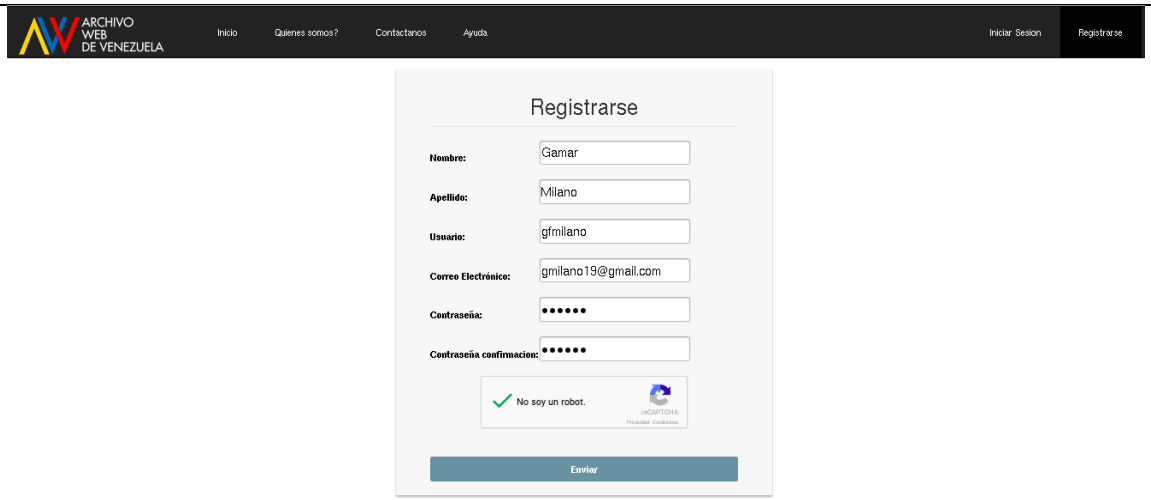
- Universidad Central de Venezuela http://www.ucv.ve/
- Universidad Simón Bolívar http://www.usb.ve/
- Universidad de los Andes http://www.ula.ve/
- Universidad de Carabobo http://www.uc.edu.ve/
- Banco Central de Venezuela http://www.bcv.org.ve/
- PDVSA la estancia http://www.pdvsalaestancia.com/
- Instituto Venezolano de Investigaciones Científicas (IVIC) http://www.ivic.gob.ve/

Casos de prueba

Se listarán a continuación los casos de pruebas diseñados, el resultado esperado y la evidencia tomada al momento de ejecutar el caso.

Tabla 27. Caso de Prueba 001 Registrar Nuevo Usuario

Número de Caso de Prueba: 1	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar registro de nuevo usuario	
Resultado: Registro exitoso, solicitud enviada al administrador para validación.	

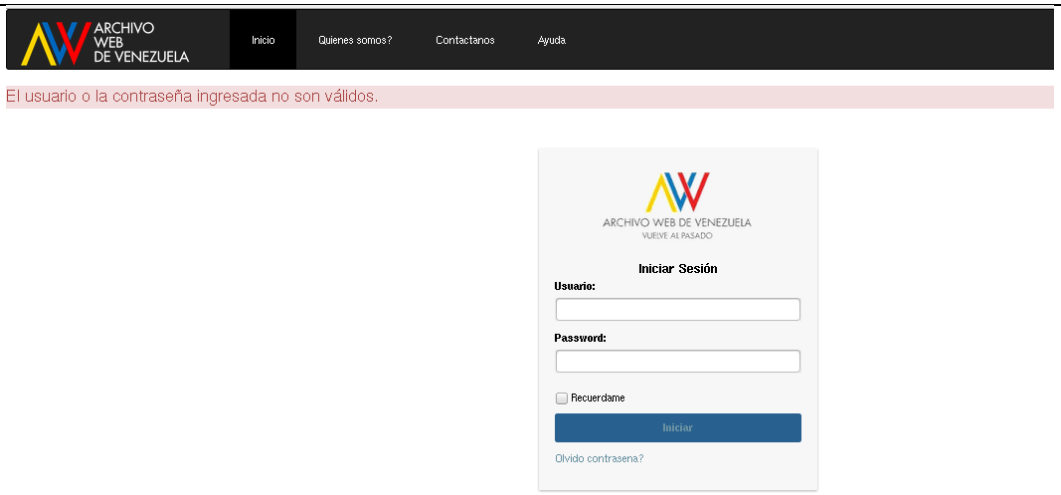


Falla Obtenida:

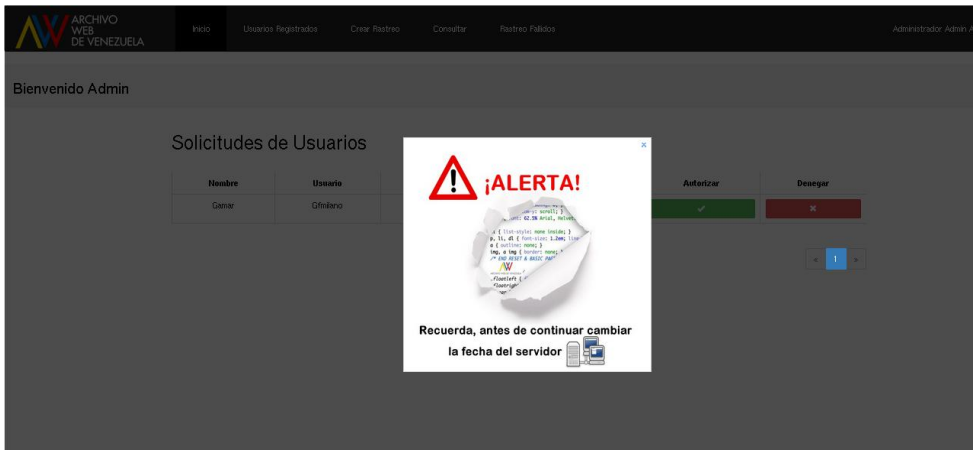
1. Inconvenientes con el api de *reCAPTCHA* de google, para el ingreso de la clave generada en las maquinas virtuales. **Solución**: se realizo nueva generación de una clave del api para lograr que se mostrara *captcha* como medida de seguridad.

Tabla 28. Caso de Prueba Validar Datos Ingresados

Número de Caso de Prueba: 2	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar ingreso a la aplicación con usuarios registrados	
Resultado: Ingreso exitoso, en lugar de ingresar con datos incorrectos se emite el mensaje de error indicando que los datos son inválidos.	



Ingreso con datos correctos, se visualiza el menú principal de la aplicación, según el rol que posea el usuario.

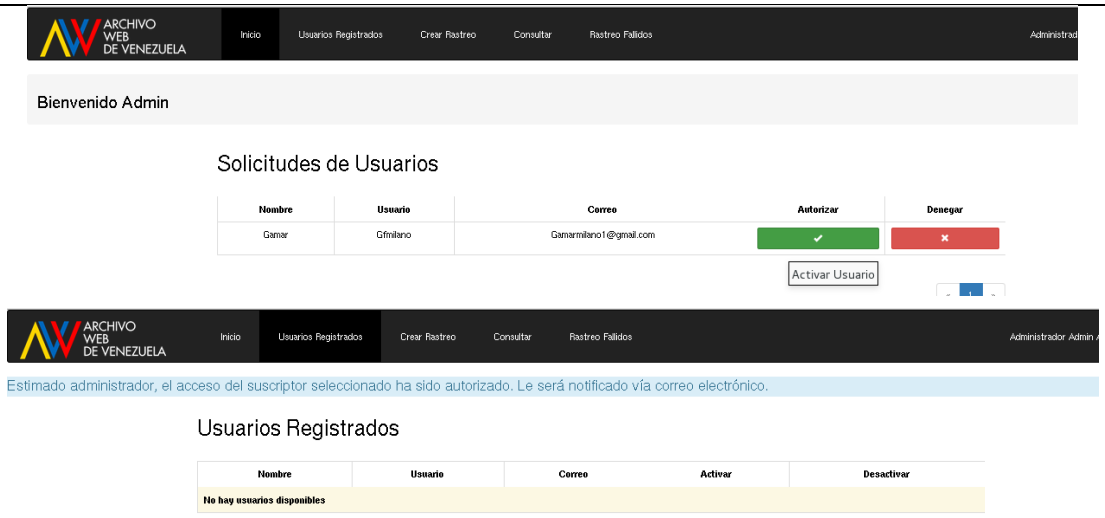


Fallas Encontradas:

1. Falla en la diferenciación de los roles de usuario y el ingreso al portal. **Solución**: modificación del models.py para agregar el atributo *is_admin* a la *Class User.*
2. Falla en la inclusión de la alerta al administrador indicando que debe ser modificada la hora del servidor. **Solución:** Inlcuir popup con mensaje de alerta al realizar el ingreso a la aplicación.

Tabla 29. Caso de Prueba Validar Autorizar Solicitud de Registro

Número de Caso de Prueba: 3	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar autorizar nuevos usuarios con el rol administrador	
Resultado: Se emite el mensaje correspondiente de autorización del usuario.	



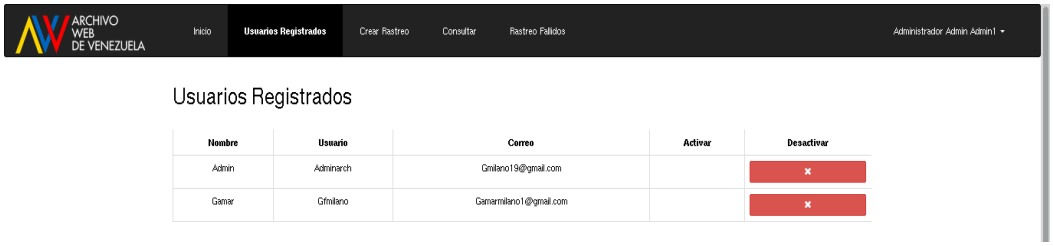
Falla Encontrada:

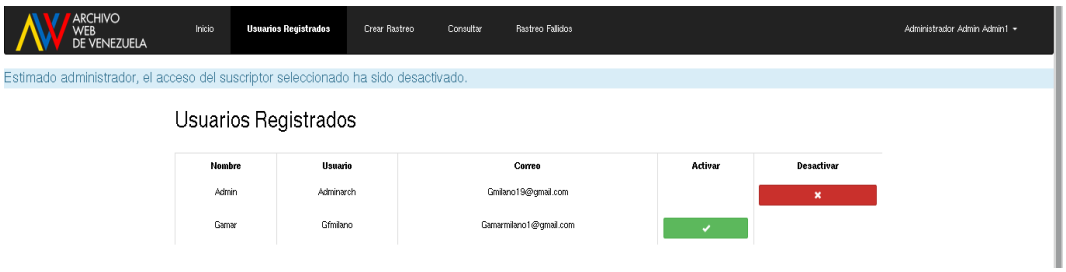
1. No se recibía notificación por parte de la aplicación que el acceso del usuario fue autorizado o negado. **Solución**: Inclusión de envío de correos para la notificación al usuario.

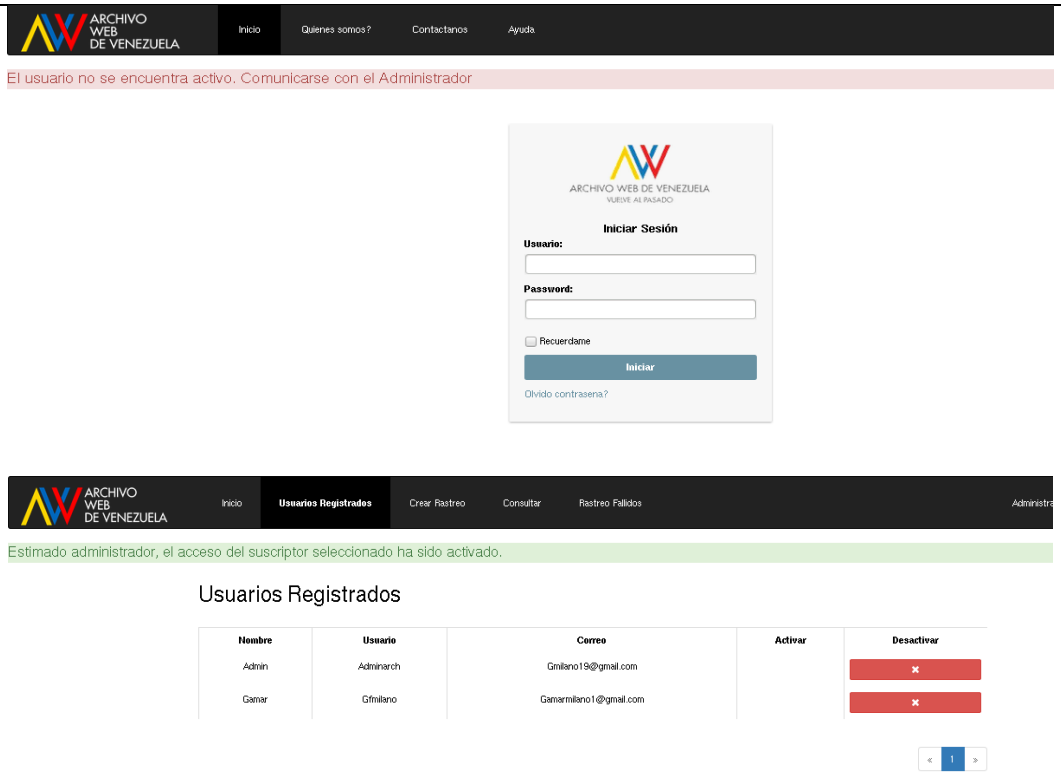
Tabla 30. Caso de Prueba Habilitar/Deshabilitar Acceso a Suscriptores

Número de Caso de Prueba: 4	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar habilitar o deshabilitar el acceso de los suscriptores	

Resultado: Se deshabilita el acceso del suscriptor al cual no se le permite ingresar a la aplicación, se habilita el acceso al suscriptor.





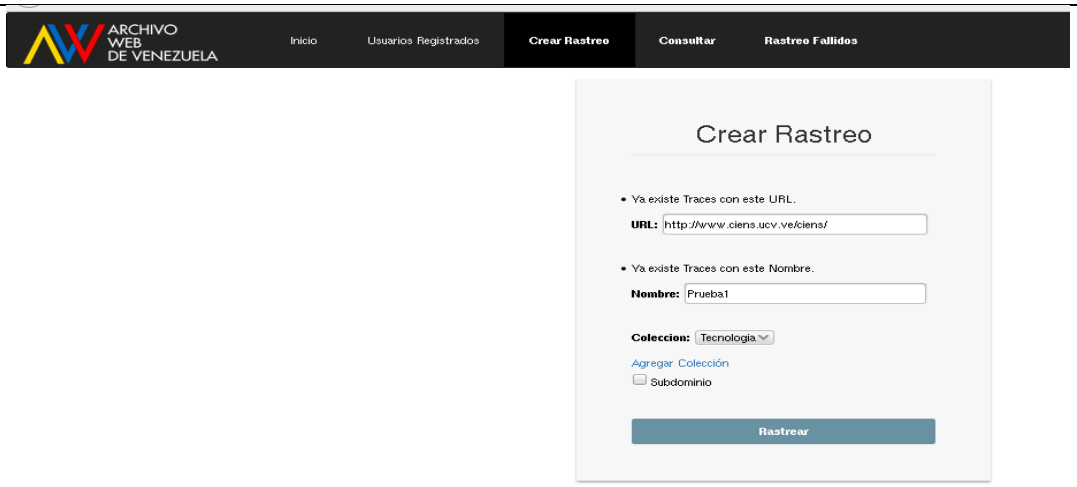


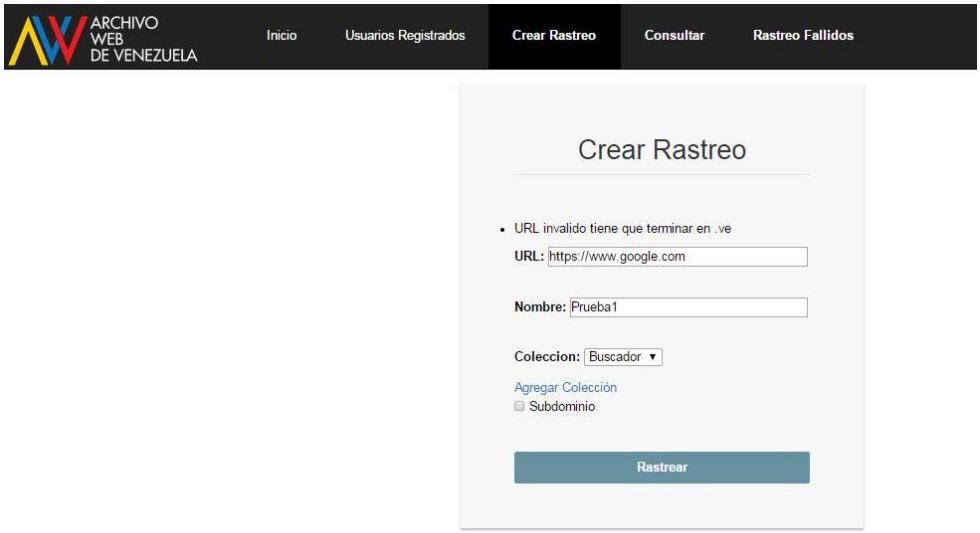
Falla Encontrada:

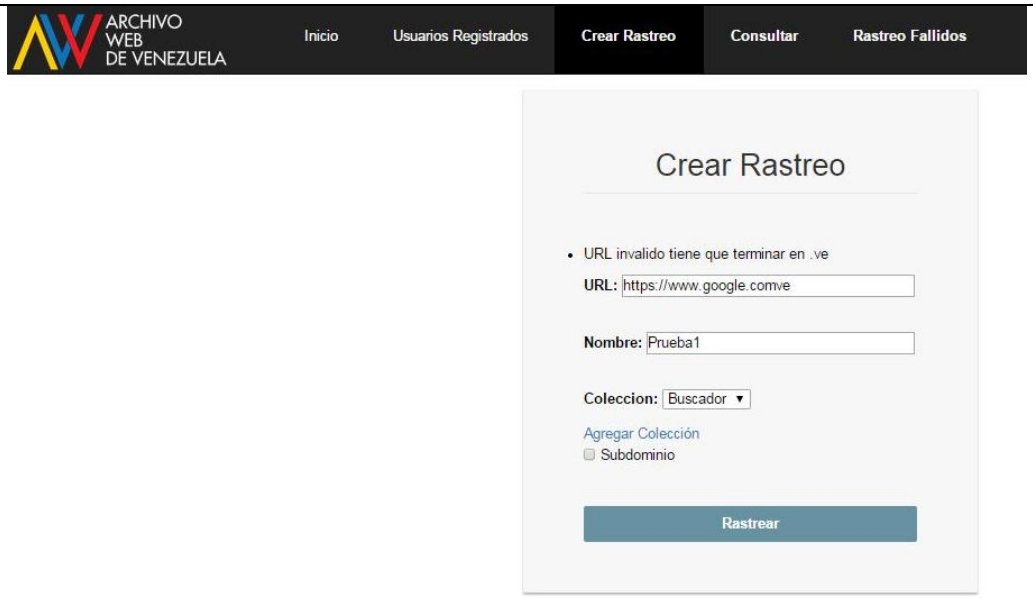
1. Falla en el ingreso al portal cuando el acceso había sido deshabilitado. **Solución**: modificación del models.py para agregar el atributo *is_active* a la *Class User.*

Tabla 31. Caso de Prueba Validar Datos de Entrada Solicitud de Rastreo

Número de Caso de Prueba: 5	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar datos de entrada al crear nueva solicitud de rastreo	
Resultado: Se emiten los mensajes correspondientes al momento de crear una solicitud de rastreo con información incorrecta.	


ARCHIVO WEB DE VENEZUELA
Inicio
Usuarios Registrados
Crear Rastreo
Consultar
Rastreo Fallidos
Crear Rastreo
Ya existe Traces con este URL.
URL: http://www.ciens.ucv.ve/ciens/
Ya existe Traces con este Nombre.
Nombre: Prueba1
Coleccion: Tecnologia
Agregar Colección
Subdominio
Rastrear


ARCHIVO WEB DE VENEZUELA
Inicio
Usuarios Registrados
Crear Rastreo
Consultar
Rastreo Fallidos
Crear Rastreo
URL invalido tiene que terminar en .ve
URL: https://www.google.com
Nombre: Prueba1
Coleccion: Buscador
Agregar Colección
Subdominio
Rastrear

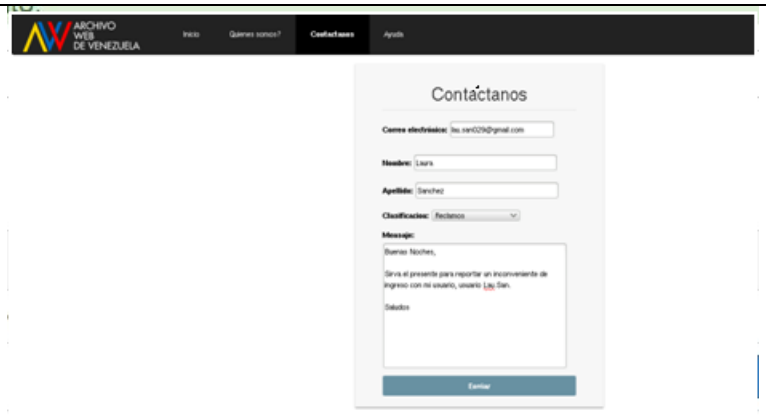


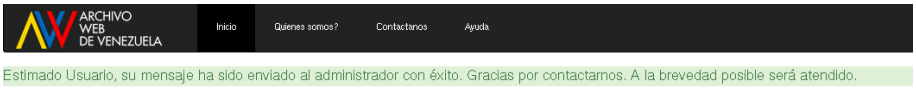
Falla Encontrada:

1. Falla al realizar los mensajes de verificación de los URL ingresado por el usuario. Django solo implementa mensajes de alerta cuando el rastreo existe no para ingresar páginas con características especiales. **Solución**: Inclusión de la validación relacionada a solo rastrear paginas con extensiones .ve en Class RastreoForm() de Forms.py

Tabla 32. Caso de Prueba Validar Opción Contactar al Administrador

Número de Caso de Prueba: 6	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Opción Contactar al Administrador	
Resultado: Se envía de manera exitosa el correo al administrador con la información ingresada por el usuario.	

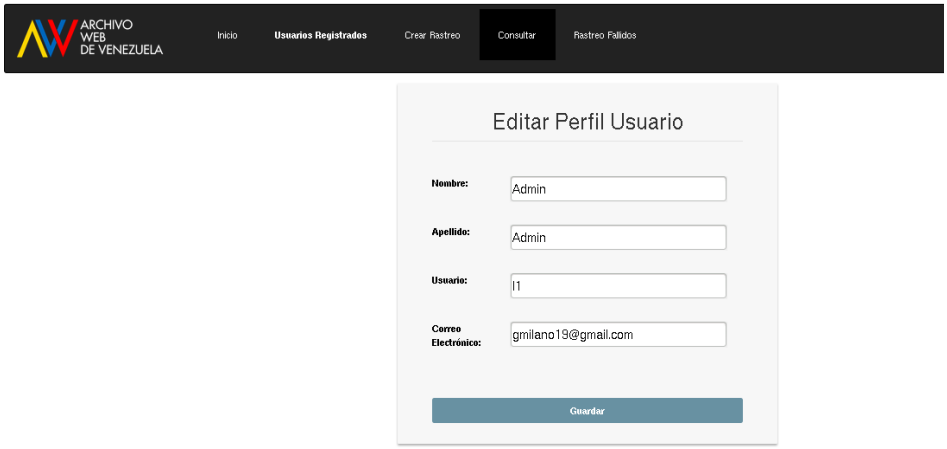


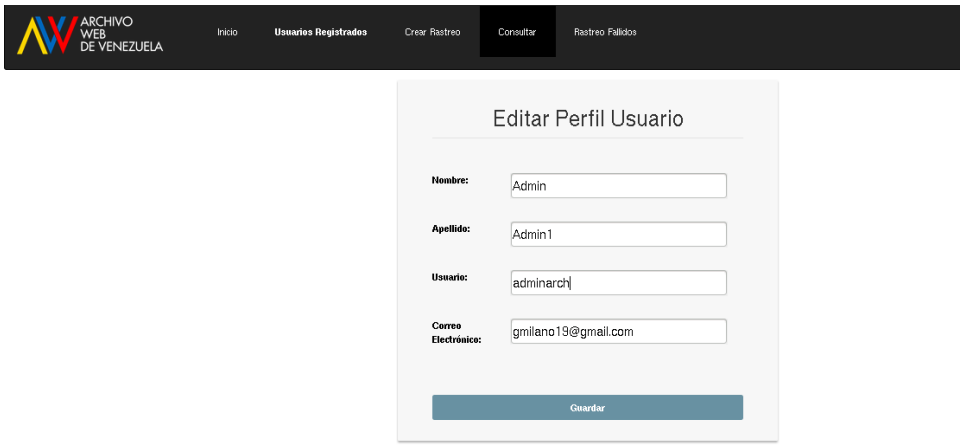


Estimado Usuario, su mensaje ha sido enviado al administrador con éxito. Gracias por contactarnos. A la brevedad posible será atendido.

Tabla 33. Caso de Prueba Validar Opción Editar Perfil de Suscriptor

Número de Caso de Prueba: 7	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Opción Editar Perfil de Suscriptor	
Resultado: Se modifica la información del suscriptor de manera exitosa.	





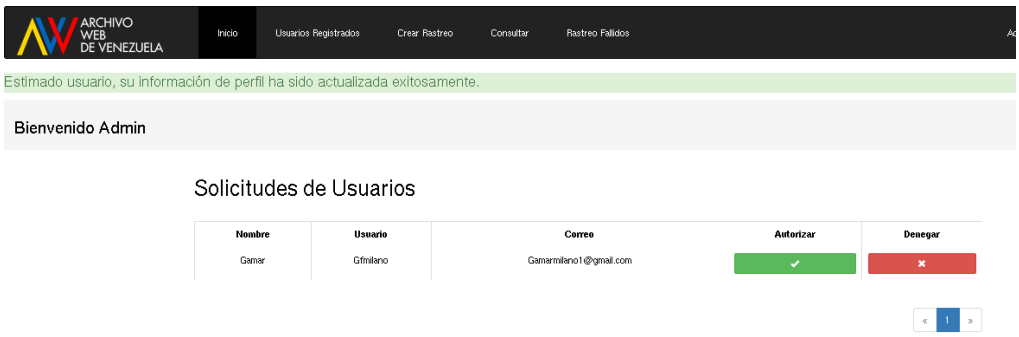
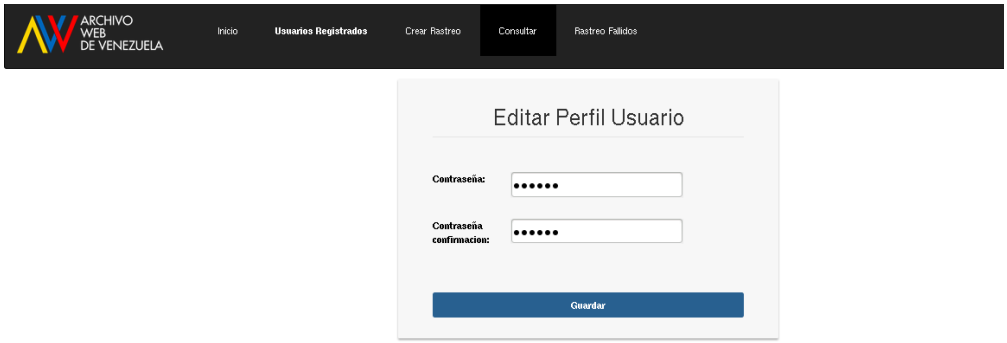


Tabla 34. Caso de Prueba Validar Modificar Contraseña

Número de Caso de Prueba: 8	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Modificar Contraseña	
Resultado: Se modifica la contraseña de manera exitosa.	



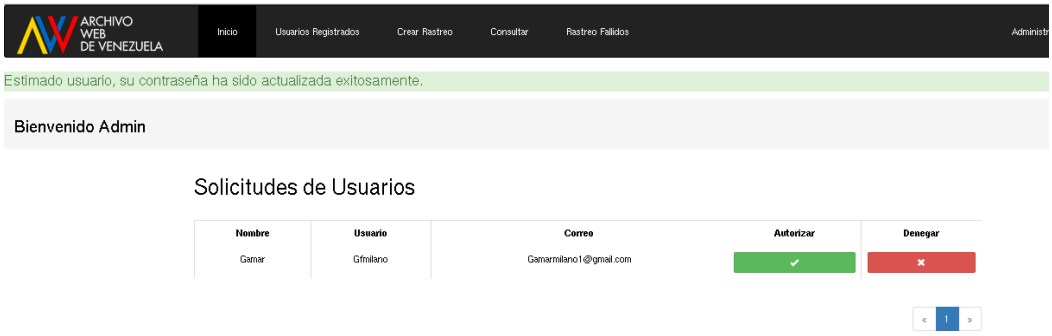


Tabla 35. Caso de Prueba Creación de Solicitud de Rastreo

Número de Caso de Prueba: 9	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Creación de Solicitud de Rastreo	
Resultado: Se crea la solicitud de manera exitosa.	
Falla Encontrada: 1. Falla no se realiza el rastreo por encontrarse Heritrix caído. **Solución**: Realizar ejecución de Heritrix antes de iniciar los trabajos de rastreos, utilizar el comando $HERITRIX_HOME/bin/heritrix	

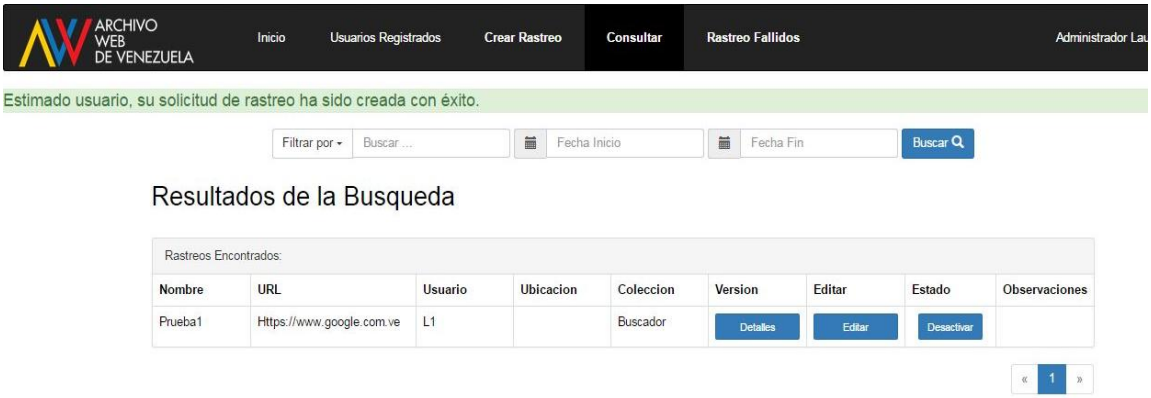
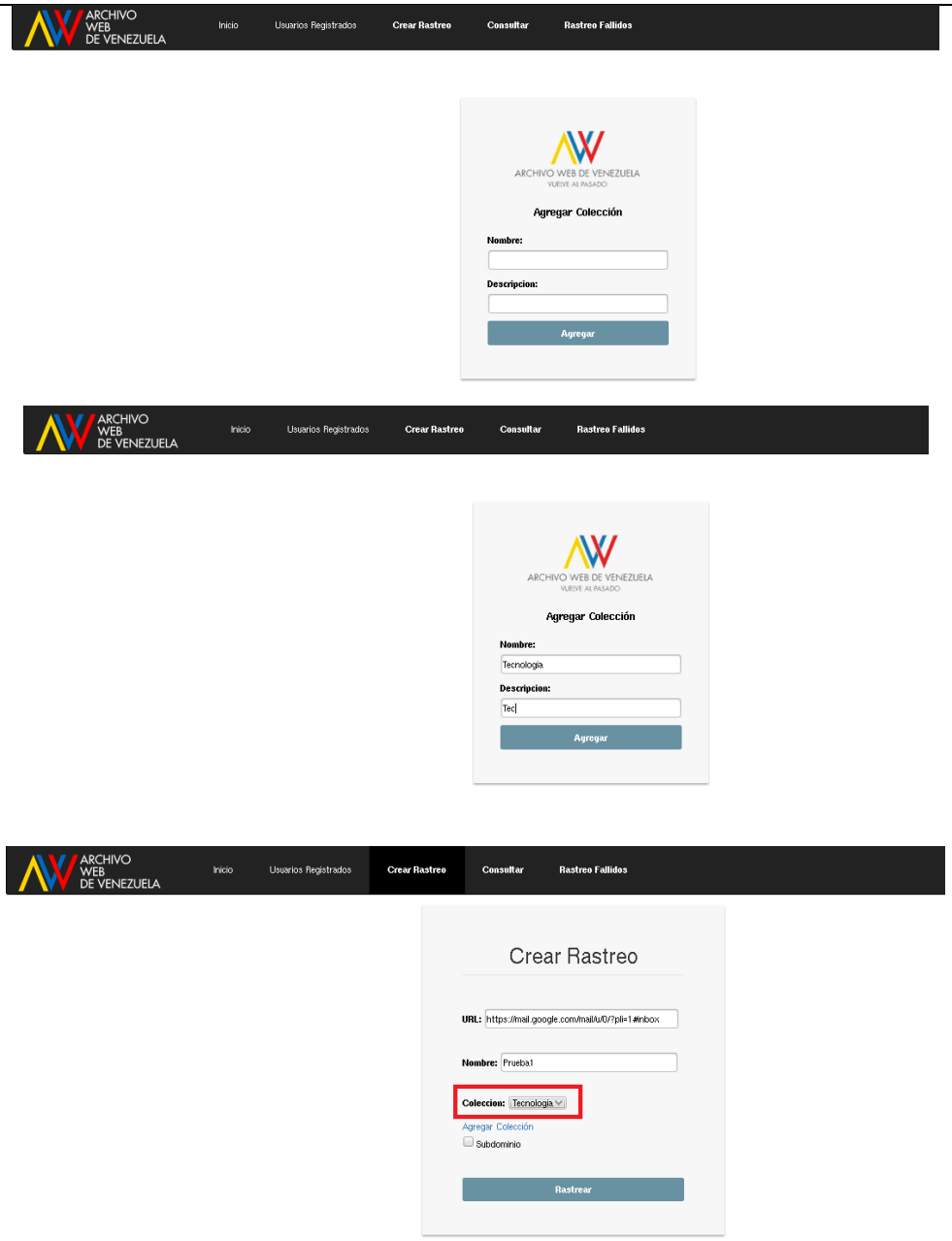


Tabla 36. Caso de Prueba Validar Creación de Colección

Número de Caso de Prueba:10	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Creación de Colección	
Resultado: Se crea la colección de manera exitosa.	



Falla Encontrada:

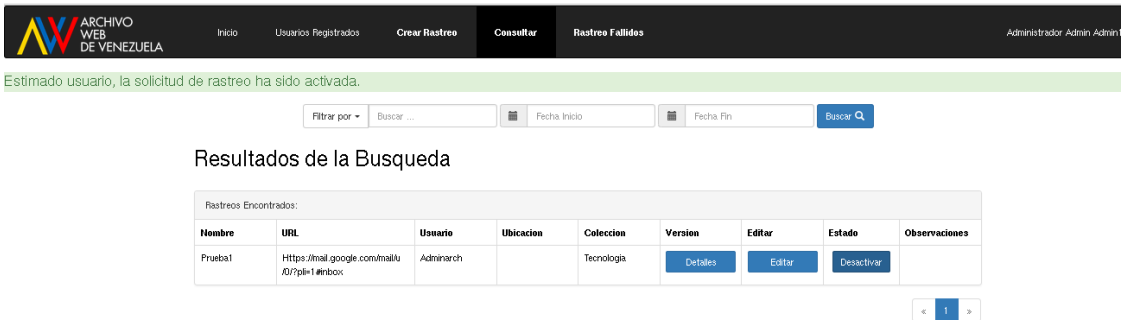
1. Al ser primera vez que se utiliza el sistema no se cuenta con colecciones creadas. **Solución**: se debe realizar la inclusión de una colección nueva antes de comenzar los trabajos de rastreos por primera vez.

Tabla 37. Caso de Prueba Desactivar - Activar Rastreos

Número de Caso de Prueba: 11	**Módulo de Gestión y Control de Incidencias**

Descripción: Validar Desactivar/Activar Rastreos

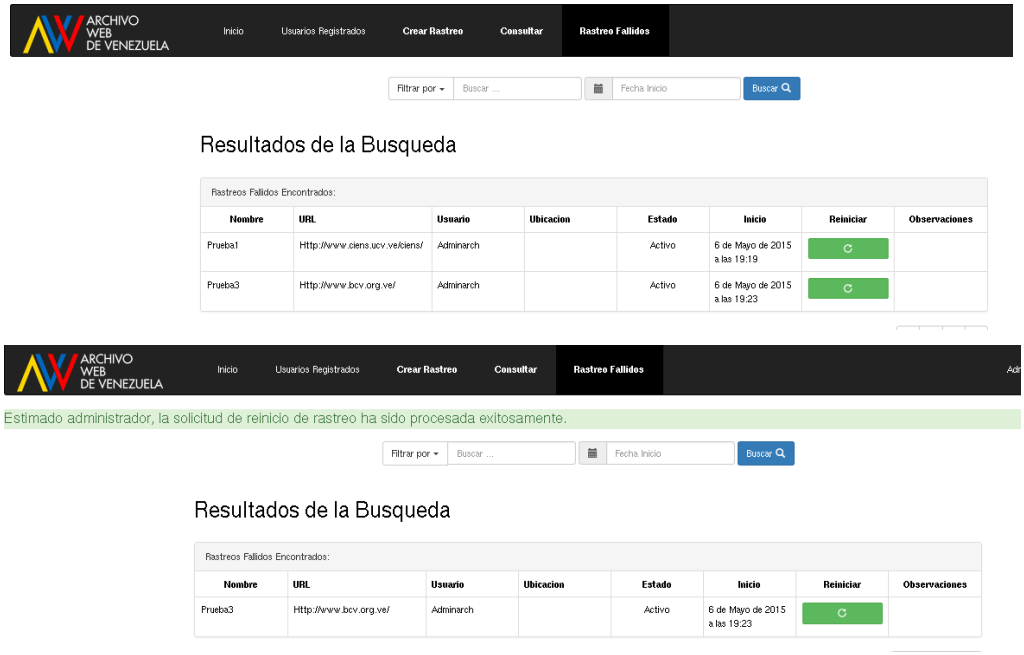
Resultado: Se desactiva o activa de manera exitosa el rastreo.

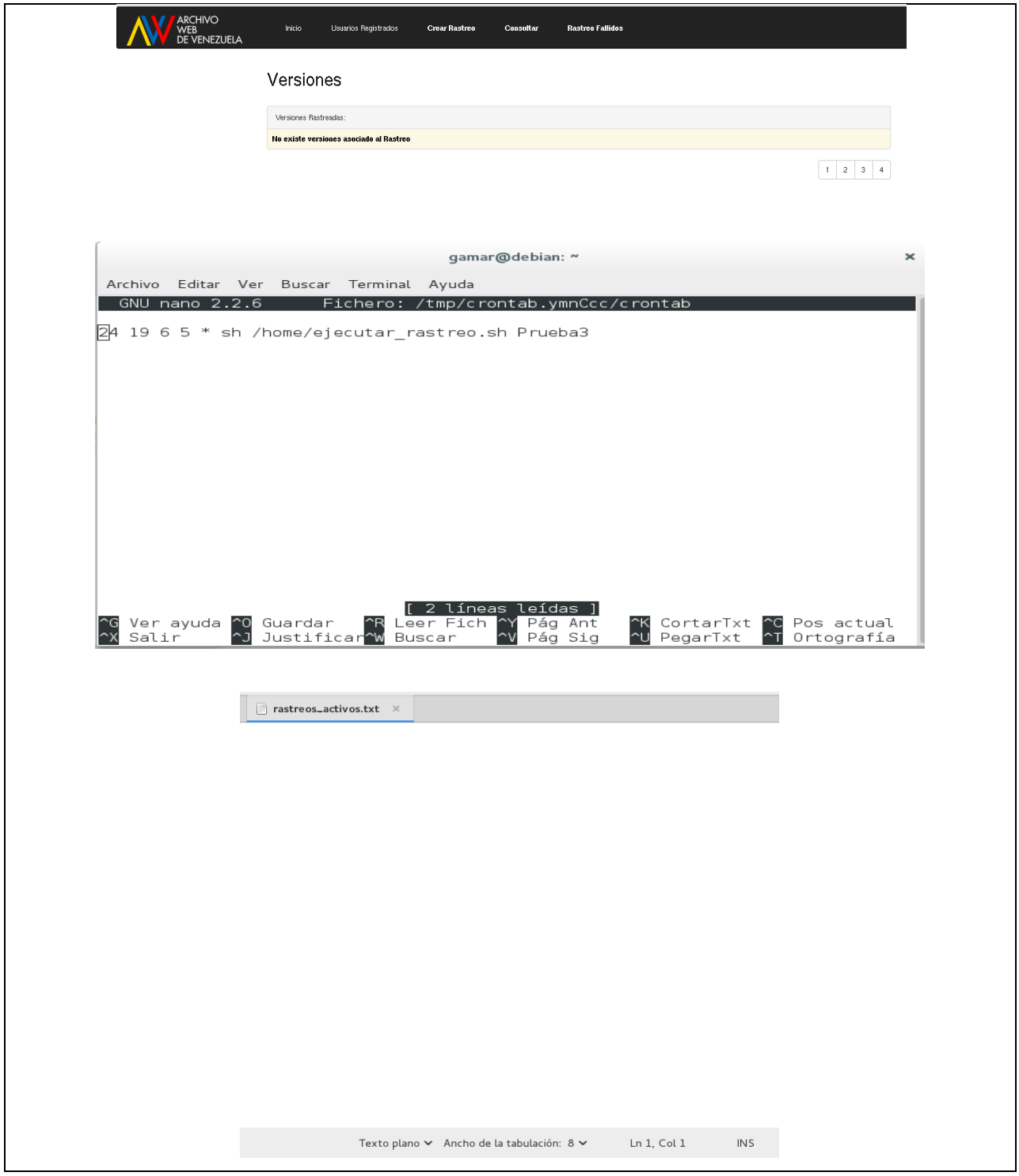


```
gamar@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
  GNU nano 2.2.6        Fichero: /tmp/crontab.ymnCcc/crontab

30 19 6 5 * sh /home/ejecutar_rastreo.sh Prueba1

                                [ 2 líneas leídas ]
^G Ver ayuda  ^O Guardar    ^R Leer Fich  ^Y Pág Ant    ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar ^W Buscar     ^V Pág Sig    ^U PegarTxt   ^T Ortografía
```

Tabla 38. Caso de Prueba Validar Opción Reiniciar Rastreo

Número de Caso de Prueba: 12	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Reiniciar Rastreo	
Resultado: Se reinicia manera exitosa el rastreo.	




ARCHIVO WEB DE VENEZUELA
Inicio
Usuarios Registrados
Crear Rastreo
Consultar
Rastreo Fallidos
Versiones
Versiones Rastreadas:
No existe versiones asociado al Rastreo
1 2 3 4
gamar@debian: ~
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /tmp/crontab.ymnCcc/crontab
24 19 6 5 * sh /home/ejecutar_rastreo.sh Prueba3
[2 líneas leídas]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y Pág Ant ^K CortarTxt ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^V Pág Sig ^U PegarTxt ^T Ortografía
rastreos_activos.txt
Texto plano
Ancho de la tabulación: 8
Ln 1, Col 1
INS

rastreos_activos.txt /home/gamar/heritrix-3.2.0/jobs/Prueba3/20150506235406
Falla Encontrada: 1. Establecer los pasos para reiniciar un rastreo en caso de falla por inconvenientes de luz. **Solución**: Los inconvenientes con servicio eléctrico dejaban al rastreo sin terminar, el rastreo queda en estado running sin realizar ningún proceso, se realizó un script el cual termina y coloca el rastreo en ejecución.

Proceso de Pruebas en Ambiente de Producción

Número de Caso de Prueba: 12	**Módulo de Gestión y Control de Incidencias**
Descripción: Validar Ejecución de Rastreo en las maquinas virtuales del servidor	

Resultado: Rastreo realizado de manera exitosa, aplicación ejecutándose en la máquina 190.169.69.154 y ejecución del rastreo realizada en la 190.169.69.155.

En la siguiente imagen se visualiza que el job que creado exista y haya sido colocado en ejecución.

```
gamar@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Heritrix1: ~/allData# ls
Usb
root@Heritrix1:~/allData# cd Usb/
root@Heritrix1:~/allData# cd Usb/ ls
20150513020106  crawler-beans.cxml  job.log.lck  scratch    state
action          job.log             latest       seeds.txt
root@Heritrix1: ~/allData#
```

En la siguiente imagen se visualiza el rastreo finalizado.

```
gamar@debian: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
root@Heritrix1:/home/allData# cd Usb/
root@Heritrix1:/home/allData/Usb# cd 20150512212303/
root@Heritrix1:/home/allData/Usb/20150512212303# cd warcs/
root@Heritrix1:/home/allData/Usb/20150512212303/warcs# ls
WEB-20150512212305-00000-2223~Heritrix1.warc.gz
root@Heritrix1:/home/allData/Usb/20150512212303/warcs#
```

Fallas Encontradas:

1. Falla no se realiza el rastreo por encontrarse Heritrix caído. **Solución**: Realizar ejecución de Heritrix antes de iniciar los trabajos de rastreos, ingresar en el comando cd /home/Heritrix-3.2.0/bin para ubicarse en la carpeta de Heritrix, luego ejecutar la siguiente línea de código: ./Heritrix –a <usuario de Heritrix de la maquina utilizada>:<clave de Heritrix de la maquina utilizada> -j /home/allData –b ip de la maquina utilizada.
2. Falla en la comunicación entre las maquinas virtuales para ejecución de los script generados. Solución: fue implementado la comunicación ssh con llave privada/llave pública, visualizar pasos realizados en el Anexo B.

Conclusiones

Al culminar este Trabajo Especial de Grado llamado "Implementación del Módulo de Gestión y Control de Incidencias del Prototipo de Archivo Web en Venezuela", se considera que se cumplieron los objetivos planteados debido a que se implementó el Módulo de Gestión y Control de Incidencias, el cual es una aplicación Web funcional y operativa para el control de incidencias de los rastreos fallidos ocasionados por los inconvenientes con el servicio eléctrico, esto permite que los rastreos no se vean afectados, se le brinda una ayuda al usuario en los aspectos relacionados a la creación y configuración de rastreos que de otra manera deberían hacerse de forma manual y requeriría tener conocimientos sobre las herramientas utilizadas y los componentes internos del rastreador, en este caso de Heritrix.

De igual manera se realiza la gestión de usuarios que podrán acceder al sistema, la supervisión de los trabajos de rastreos, esto efectuado por un usuario bajo el rol de administrador.

Se realizó la modificación de la interfaz del Módulo de Adquisición, lo cual proporciona al usuario confianza en el portal Web, lo que produce que lo usen regularmente y cumpla con el fin para el cual fue creado.

Con las pruebas que se le hicieron al Módulo de Gestión y Control de Incidencias del Prototipo se pudo constatar que los procesos desarrollados funcionan de forma correcta con respecto a los requerimientos funcionales establecidos al inicio de la presente investigación, siempre teniendo en cuenta que todo software puede ser mejorado y difícilmente exista un software infalible.

Se concluye que en la actualidad se cuenta con un Prototipo capaz de realizar el rastreo y almacenamiento de sitios Web de índole cultural, educativa o que en general sea representativo de las manifestaciones y cultura venezolana, esto en aras de salvaguardar la información para generaciones futuras.

Trabajos Futuros

Utilización de un clúster como por ejemplo *Hadoop*, para cómputo distribuido, en la gestión y asignación de rastreos, para lograr un equilibrio en la carga de los rastreos de una manera más eficiente.

Implementación de un módulo de predicción, encargado de realizar la medición del tiempo en cuánto tiempo será realizado el próximo trabajo de rastreo, ello para realizar rastreos de frecuencias variables.

Agregar un módulo de análisis de métricas, que permita realizar reportes a demanda del usuario a partir de las mediciones almacenadas y que permita determinar, por un lado el comportamiento de los sitios web y por otro lado, necesidades de crecimiento del Archivo a mediano y largo plazo.

Establecer políticas de revisión de las páginas a rastrear antes de realizar el proceso de indexación, en aras de optimizar el trabajo realizado por los motores de búsqueda. Políticas como describir con precisión el contenido de la página, utilizar la metaetiqueta *description* la cual brinda a los motores de búsqueda un resumen sobre la página.

En cuanto a la búsqueda de contenido, se propone brindarle al usuario la posibilidad de filtrar sus búsquedas de acuerdo a información contenida en el WARC, esto para que el usuario pueda hacer una consulta más amplia y/o específica sobre los contenedores.

Referencias Bibliográficas

Chris Oliver. (2015).*Ruby vs. Python*. Obtenido de https://onemonth.com/blog/ruby-vs-python

Django . (2005). Obtenido de https://docs.djangoproject.com/en/1.8/

García, J., & Rivero, L. (2013).*Desarrollo de prototipo para el Archivo Web de Venezuela. (Tesis de pregrado).* Universidad Central de Venezuela, Caracas - Venezuela.

Ghosh, S. T. (2007).*Systemic Comparison of the Application of EVM in Traditional and Agile Software.* The University of Maryland, College Park .

***Heritrix*. (2014).** Obtenido de https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

Ian, S. (2006).*Ingenieria del Software.* Pearson Addison Wesley.

***IIPC*. (2012).** Obtenido de http://netpreserve.org/working-groups

IIPC. (2014). Obtenido de http://netpreserve.org/about-us/members

Ingeniería de software basada en componentes. (Febrero de 2013). Obtenido de http://netpreserve.org/http://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_

Internet Archive. (2013).*Internet Archive*. Recuperado el 2014, de http://crawler.archive.org/

ISO. (2009).*ISO. 28500 Information and documentation-WARC file format.* Nueva Zelanda.

Jack, P., & Binns, A. (2012).*Web Archive - Heritrix*. Obtenido de https://webarchive.jira.com/wiki/display/Heritrix/Heritrix

Jacobson, I. (1998).*Applying UML in the Unified Process*. Obtenido de http://www.powershow.com/view/f2189-ZTk1N/Applying_UML_in_The_Unified_Process_Ivar_Jacobson_Rational_Software_email_ivar_rationalcom_powerpoint_ppt_presentation

Lambert, K. (2012).*Fundamentals of Python: First Programs.*Canada: Course Technology.

Mantura, K., & Martínez, M. (2014).*Definición de las estrategias para el desarrollo del Módulo de Acceso de los contenidos Preservados en Formato*

WARC para el Prototipo de Archivo Web en Venezuela. (Tesis de Pregrado). Caracas. Universidad Central de Venezuela, Caracas - Venezuela.

Masanès, J. (2006).*Web Archive.* New York: ISBN-10 3-540-23338-5.

Organización de las Naciones Unidas para la Educación, l. C. (2003). *DIRECTRICES PARA LA PRESERVACIÓN DEL PATRIMONIO DIGITAL.* Australia : Biblioteca Nacional de Australia.

Ospina, M. (2014).*Un marco de referencia para la implementación de Archivos Web. Trabajo de Grado de Maestría.* Universidad Central de Venezuela. Caracas - Venezuela.

Ospina, M., Martinez, M., Kabchi, M., & León, C. (2014). Desarrollo de una Aplicación para Acceder a Contenidos de un Archivo Web. *CONCISA Segunda Conferencia.* Universidad Católica Andrés Bello, Caracas - Venezuela.

Ospina, Martinez, Leon y Kabchi. (2014).*Desarrollo de una Aplicación para Acceder a Contenidos de un Archivo Web en Formato*Universidad Central de Venezuela, Facultad de Ciencias, Escuela de Computación

PANDORA. (1996). Obtenido de http://pandora.nla.gov.au/

Python Software Foundation. (2001). Obtenido de https://docs.python.org/2/faq/general.html#why-was-python-created-in-the-first-place

***Ruby*. (s.f.).** Recuperado el 2015, de https://www.ruby-lang.org/es/

***The WARC File Format*. (Marzo de 2007).**Obtenido de http://archive-access.sourceforge.net/warc/warc_file_format-0.16.html#anchor46

UNESCO. (2003).*Noción de preservación digital*. Obtenido de http://www.unesco.org/new/es/communication-and-information/access-to-knowledge/preservation-of-documentary-heritage/digital-heritage/concept-of-digital-preservation/

Anexo A Instalación de Heritrix versión 3.2.0

Se descarga Heritrix de su página web:

http://builds.archive.org/maven2/org/archive/heritrix/heritrix/3.2.0/heritrix-3.2.0-dist.tar.gz

Una vez descargado se descomprime el archivo, utilizar el comando unzip heritrix-3.2.0-dist.zip

JAVA_HOME: Debe apuntar al directorio de instalación de JRE 1.6 o mayor, por ejemplo JAVA_HOME=/usr/local/java/jre

HERITRIX_HOME: debe apuntar al directorio de instalación de heritrix, por ejemplo HERITRIX_HOME=/home/user/heritrix3.2.0

JAVA_OPTS: Esta variable es usada para definir la cantidad de memoria asignada a heritrix, por ejemplo un valor igual a -Xmx1024M, indica que se está asignando 1Gb de memoria

Adicionalmente se le debe dar permiso de ejecución al archivo HERITRIX_HOME/bin/heritrix, como se muestra en la Figura 74.

```
1  chmod u+x $HERITRIX_HOME/bin/heritrix
```

Figura 74.Heritrix permisos

Ejecutar Heritrix

Para ejecutar Heritrix se usa el comando que se puede apreciar en la Figura 75.

```
1  $HERITRIX_HOME/bin/heritrix
```

Figura 75. Comando para ejecutar Heritrix

API REST

Heritrix usa REST para exponer sus funcionalidades, la implementación REST de heritrix está basada en Restlet que es un frameworkRestFul para java.

Heritrix expone esta API a través de HTTPS, a través de este protocolo se hacen peticiones para recuperar o modificar configuraciones y manejos de rastreos.

Cualquier cliente que soporte HTTPS puede ser usado para invocar el API REST, por ejemplo se puede usar el cliente de línea de comandos curl.

CrearJobs

Para crear un nuevo *job* se utiliza el comando que puede apreciarse en la Figura 76.

```
1  curl -v -d "createpath=myjob&action=create" -k -u <usuario>:<contraseña>
2  --anyauth  --location https://<HeritrixHost>:8443/engine
```

Figura 76. Crear Job

Donde:

createpath: Nombre del job

<usuario>: usuario de heritrix

<contrasena>: contraseña del usuario heritrix

<HeritrixHost>: Host donde está corriendo heritrix

Construir el *Job*

Luego de que un *job* es creado se debe leer el archivo de configuración necesario para correr el rastreo, a partir de las especificaciones dictadas en este archivo se construye el *job*, el comando utilizado puede apreciarse en la Figura 77.

```
1  curl -v -d "action=build" -k -u <usuario>:<contraseña> --anyauth
2  --location https://<HeritrixHost>:8443/engine/job/<nombreJob>
```

Figura 77. Construir Job

Donde:

<usuario>: usuario de heritrix

<contrasena>: contraseña del usuario heritrix

<HeritrixHost>: Host donde está corriendo heritrix

<nombreJob>: Nombre del *job*

Lanzar el *job*

Una vez el *job* ha sido construido, es necesario lanzarlo o ponerlo a correr, para ello se usa el comando que puede apreciarse en la Figura 78.

```
1  curl -v -d "action=launch" -k -u <usuario>:<contraseña> --anyauth
2  --location https://<HeritrixHost>:8443/engine/job/<nombreJob>
```

Figura 78. Lanzar Job

Donde:

<usuario>: usuario de heritrix

<contrasena>: contraseña del usuario heritrix

<HeritrixHost>: Host donde está corriendo heritrix

<nombreJob> : Nombre del *job*

Quitar la pausa a un *job*

Por defecto el rastreo es lanzando en modo pausa, por lo que es necesario quitarle la pausa para que pueda comenzar a procesar, el comando usado puede apreciarse en la Figura 79.

```
1  curl -v -d "action=unpause" -k -u <usuario>:<contraseña> --anyauth
2  --location https://<HeritrixHost>:8443/engine/job/<nombreJob>
```

Figura 79. Despausar job

Donde:

<usuario>: usuario de heritrix

<contrasena>: contraseña del usuario heritrix

<HeritrixHost>: Host donde está corriendo heritrix

<nombreJob>: Nombre del *job*

Anexo B Conexión entre las diferentes maquinas virtuales existentes

Pasos para realizar la conexión entres Solr, Heritrix1, Heritrix2 y Heritrix3.

1. Ingresar en la terminal en una de las maquinas virtuales y escribir el siguiente comando: "***ssh-keygen***", esto generará una llave pública
2. Se solicitara la información de la ubicación del archivo con las llaves públicas y privadas y un *phrase* para establecer la comunicación, simplemente presionemos **[Enter]**, un segundo después volvemos a presionar **[Enter]**, y un segundo después volvemos una vez más a presionar **[Enter]**. Es decir, presionaríamos **[Enter]** un total de tres (3) veces, solo lo presionamos, no escribimos nada, con esto ya fueron generadas la llave pública y privada.
3. Ingresamos el siguiente comando **scp id_rsa.pub root@<IP de la maquina con la cual se establecera conexión>:/root/.ssh**
4. Se conecta a la maquina a la cual se envío la clave pública y se ubica en la ruta /root/.ssh.
5. Se procede a escribir los siguientes comandos:
    a. Touch authorized_keys
    b. Cat id_ras.pub >> authorized_keys
6. Ya la conexión unidireccional se encuentra establecida, para establecer la conexión bidireccional repetir los pasos del 1 al 5 intercambiando las maquinas antes utilizadas.