Universidad Central “Marta Abreu” de Las Villas
Facultad de Matemática, Física y Computación
Departamento de Ciencia de la Computación

Integración de instancias del Sistema de Gestión de Documentos Históricos ArchiVenHIS

Tesis para optar por el título académico de
Máster en Ciencia de la Computación

Autor: Ing. Reynier Pernía Rodríguez.
Tutor: Dr.C Abel Rodríguez Morffi.

Santa Clara, 2013

“Hacer memoria es hacer patria.”

Hugo Rafael Chávez Frías

Resumen

Con el objetivo de extender el Sistema de Gestión de Documentos Históricos ArchiVenHIS para otras instituciones que preservan el acervo histórico de la nación venezolana, en el año 2010 surge un proyecto de colaboración con el Archivo General de la Nación de la República Bolivariana de Venezuela (AGN) en el marco del contrato "CONTRATO SOLUCIÓN INTEGRAL PARA EL SISTEMA NACIONAL DE ARCHIVOS (FASE 1)".

Por ello ha sido necesario identificar y configurar un mecanismo eficiente de integración de datos y desarrollar una aplicación Web (SAHISWEB) con la cual es posible integrar varios Archivos que utilicen ArchiVenHIS para la gestión del patrimonio documental que resguardan, para así facilitar su difusión a través de un punto de acceso único. SAHISWEB tiene además el propósito de brindar un espacio para el intercambio colaborativo entre investigadores y publicar las principales noticias del acontecer archivístico.

En la presente investigación se exponen los mecanismos que pueden emplearse para la recolección e integración de los metadatos y representaciones digitales asociados a los fondos documentales bajo la custodia de Archivos Históricos que utilicen ArchiVenHIS para describirlos. Además se incluyen las características de SAHISWEB, los principales artefactos generados durante el proceso de desarrollo, así como la valoración de la eficiencia de la variante implementada para la recolección de los datos.

Abstract

In order to extend ArchiVenHIS for others institutions that preserve the historical heritage of the Venezuelan nation, in 2010 there arises a collaborative project with the National Archives of the Bolivarian Republic of Venezuela (AGN) under the contract "SOLUTION CONTRACT FOR THE NATIONAL ARCHIVES (Phase 1)".

Thus it has been necessary to identify and configure an efficient mechanism for integration of data and develop a Web application (SAHISWEB) with which it is possible to integrate several Archives using ArchiVenHIS management safeguard documentary heritage, in order to facilitate its dissemination through a single access point. SAHISWEB also has the aim of providing a space for collaborative exchange between researchers and publishing archival major news events.

In this research are exposed mechanisms that can be used for the harvesting and integration of metadata and digital representations associated to the archival patrimony custody by Archives using ArchiVenHIS to describe it. Also included SAHISWEB features, the main artifacts generated during its development process, and an evaluation of the efficiency of the variant implemented for data collection.

Tabla de Contenido

Índice de figuras

Índice de tablas

Introducción

Las funciones de los Archivos Históricos han ido evolucionando a lo largo del tiempo, pasando de ser considerados sólo conservadores de documentos a gestores de toda la actividad que se genera en torno a ellos, con la finalidad de garantizar disponibilidad para su consulta. Sin embargo, el acceso a los documentos originales en papel provoca su deterioro por la manipulación directa que ejercen los usuarios sobre ellos (1).

Las nuevas exigencias de la sociedad en el acceso a la información y la experiencia del trabajo en los Archivos, conllevan a una modernización de sus procesos para poner a disposición de la comunidad la información que atesoran. El Archivo General de la Nación (AGN) “Francisco de Miranda” de la República Bolivariana de Venezuela es una institución adscrita al Ministerio del Poder Popular para la Cultura que tiene entre sus competencias la custodia, conservación y divulgación del patrimonio documental que representa la memoria histórica de aquella nación. En consonancia con su propia visión pretende ser una institución de referencia nacional e internacional, actualizando y adaptándose a las nuevas tecnologías de la información y las comunicaciones (TIC) y desarrollando políticas para lograr un enfoque socializador de la cultura venezolana y brindar facilidades para la investigación e intercambios teóricos (2).

El AGN, coherente con las políticas de integración latinoamericanas y caribeñas, se enmarca dentro del Convenio Integral de Cooperación Cuba-Venezuela. Durante el año 2007 se desarrolla el Sistema de Gestión de Archivos Históricos ArchiVenHIS, en el marco del proyecto "Uso y Aplicación de las TIC para el mejoramiento de la Gobernabilidad y Aumento de la Soberanía Tecnológica (Fase 1)", en el que se proporciona una solución que contribuye a la conservación y difusión del acervo histórico custodiado por el AGN bajo tecnologías libres y estándares abiertos. Sin embargo, esto no es suficiente en aras de hacer de dominio público la memoria histórica de la nación venezolana ya que:

- Otros Archivos Históricos que también custodian parte valiosa del patrimonio documental que representa la memoria histórica de la nación, no cuentan con herramientas que le permitan ponerla a disposición de los investigadores y público en general.

- La información disponible en un Archivo Histórico acerca del patrimonio documental resguardado en otros es escasa. Esto impone limitaciones en la calidad del servicio que se les puede brindar a investigadores y usuarios en general. Los usuarios interesados en localizar determinada documentación deben contactar y/o desplazarse hasta el (los) Archivo(s) donde presumiblemente se encuentra la información de su interés.
- No existe un espacio para el intercambio entre investigadores, historiadores, estudiantes y en general todos los usuarios interesados en consultar el patrimonio documental que representa parte de la memoria histórica de la nación venezolana.

A partir de la situación problemática existente se plantea el siguiente problema científico:

¿Cómo desarrollar e integrar los recursos informáticos necesarios para facilitar, de forma eficiente, la difusión del patrimonio documental resguardado y descrito en varios Archivos Históricos a través del sistema ArchiVenHIS y potenciar el intercambio entre investigadores?

Se define como objetivo general de la investigación: Integrar la información dispersa en varios Archivos Históricos asociada al patrimonio documental descrito a través de ArchiVenHIS mediante un mecanismo automático de adquisición de datos y una aplicación Web que facilite su difusión y potencie el intercambio entre investigadores.

Para el desarrollo de la investigación se proponen los siguientes objetivos específicos:

1. Examinar los mecanismos que pueden emplearse para la recolección e integración de metadatos y representaciones digitales asociados a los fondos documentales bajo la custodia de Archivos Históricos que utilizan ArchiVenHIS para describirlos.
2. Identificar un mecanismo apropiado y eficiente para la integración de los datos.
3. Desarrollar una aplicación Web que permita la consulta, una vez integrados, de los metadatos y representaciones digitales del patrimonio documental descrito a través de ArchiVenHIS en varios Archivos Históricos y que potencie el intercambio entre investigadores.
4. Valorar la efectividad y aplicabilidad de la solución propuesta.

Teniendo en cuenta lo descrito previamente, se formula la siguiente hipótesis: Con la integración de la información dispersa en varios Archivos Históricos asociada al

patrimonio documental descrito a través de ArchiVenHIS mediante un mecanismo automático de adquisición de datos y una aplicación Web, se facilita su difusión de forma eficiente y se potencia el intercambio entre investigadores.

Durante la investigación se emplearon los métodos científicos que se describen a continuación. Los métodos teóricos utilizados fueron: histórico-lógico con el objetivo de conocer los antecedentes del problema, la evolución del mismo y las investigaciones que se han llevado a cabo con anterioridad en el área temática; análisis-síntesis para analizar y detectar el problema mediante la interpretación de los resultados obtenidos luego de la realización de las entrevistas. Los métodos empíricos usados fueron**:** entrevistas para identificar los requerimientos del sistema; estadísticos para comprobar la efectividad de la propuesta.

El aporte metodológico de la investigación se concreta en la sistematización del proceso de recolección y replicación de datos permitiendo crear comunidades alrededor del patrimonio histórico de una nación garantizándose su difusión.

El aporte práctico se evidencia en el desarrollo satisfactorio de una aplicación Web, basada en tecnologías libres y estándares abiertos, que complementa la misión como entes socializadores de los fondos documentales bajo su custodia, de los Archivos Históricos que emplean ArchiVenHIS.

El presente trabajo está estructurado de la siguiente forma: una introducción, donde se fundamenta el valor científico del problema y se recoge el diseño teórico y metodológico de la investigación. Además de tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y los anexos. En el primer capítulo se caracteriza el sistema ArchiVenHIS y se analizan alternativas para la integración de los datos mediante replicación (MySQL y rsync) o recolección según el estándar OAI-PMH. En el segundo capítulo se describen las características principales de la solución propuesta y los artefactos generados durante la creación del sistema. Se precisa la forma de realizar la integración de los datos a través del proceso de replicación de MySQL y Rsync, y se analizan las tecnologías utilizadas. En el tercer y último capítulo se analizan los resultados de las pruebas realizadas a la aplicación. Además, se presenta un análisis comparativo entre las alternativas identificadas para la integración

de los datos, lo que permite valorar la efectividad y aplicabilidad de la solución propuesta.

Capítulo 1: Mecanismos para la integración de instancias del sistema de gestión de documentos históricos ArchiVenHIS

En este capítulo se caracterizan los mecanismos que pueden emplearse para la recolección e integración de datos. Además, se resumen las principales características del sistema gestor de documentos históricos ArchiVenHIS para facilitar la aplicación de estos mecanismos en la integración de los metadatos y representaciones digitales asociados a los fondos documentales bajo la custodia de Archivos Históricos que lo utilicen para describirlos.

1.1 Sistema gestor de documentos históricos ArchiVenHIS

El proyecto Sistema de Gestión de Documentos Históricos surge en el 2007 con el propósito de ofrecer una herramienta desarrollada sobre tecnologías libres y estándares abiertos al AGN de la República Bolivariana de Venezuela que automatice el proceso de descripción archivística y la localización de documentos. En el año 2008 se le proporciona al AGN la solución de software ArchiVenHIS que provee una serie de funcionalidades que permiten gestionar los documentos históricos que conserva la institución como parte del Contrato “Uso y Aplicación de las TIC para el mejoramiento de la Gobernabilidad y Aumento de la Soberanía Tecnológica”, de la VII Empresa Mixta Cuba-Venezuela. En el año 2010, a solicitud de esta misma entidad, se extiende el sistema mediante el desarrollo de un módulo para la gestión de la información del Área de Conservación, Preservación y Restauración Documental. Su versión 1.0 se registra en el Centro Nacional de Derecho de Autor (CENDA) en febrero de 2010.

ArchiVenHIS se ha personalizado como sistema gestor de documentos históricos, con el fin de extender su aplicación a otras instituciones cubanas tales como la Oficina de Asuntos Históricos del Consejo de Estado (OAH) y el Archivo Central del Ministerio de Comercio Exterior y la Inversión Extranjera (MINCEX) (3).

1.1.1 Características del sistema ArchiVenHIS

El sistema ArchiVenHIS incluye diversas funcionalidades que se presentan a los usuarios en dependencia de su rol en el sistema. Las mismas garantizan la descripción

de los fondos documentales que custodia una institución de Archivo según la Norma Internacional General para la Descripción Archivística ISAD(G) (4). El proceso de descripción tiene lugar a través de un flujo de trabajo en el que intervienen varios roles y durante el cual una descripción es sometida a varias revisiones, mínimo dos, antes de su aprobación. Permite además llevar un control de las consultas, tanto físicas como digitales, que se realizan sobre los documentos, así como los servicios que se brindan a los investigadores. A continuación se presentan las características más relevantes de ArchiVenHIS:

- Conformación del cuadro de clasificación de la documentación: El coordinador del área Procesos Técnicos puede definir los niveles de organización de la documentación que conforman el cuadro de clasificación. Cuenta con funcionalidades para crear y describir las distintas instancias de los tipos de niveles empleados por la institución para la organización del fondo documental (fondo, subfondo, sección, serie, etc).
- Descripción de documentos: Las descripciones de los documentos son incorporadas al sistema por transcriptores, descriptores o coordinadores del área Procesos Técnicos. Se establece un flujo de trabajo entre ellos que garantiza varios niveles de revisión para su aprobación y puesta a disposición del público. La descripción se realiza según la norma ISAD (G). El sistema garantiza que los seis campos obligatorios que establece la norma no se encuentren vacíos (4).
- Trascripción paleográfica de documentos: Los transcriptores paleográficos asocian su trascripción paleográfica a los documentos descritos que así lo requieran.
- Incorporación de imágenes: Los digitalizadores asocian las representaciones digitales con los documentos descritos.
- Actualización de la estructura de almacenamiento: El coordinador del área Sala de Lectura puede actualizar la estructura de almacenamiento empleada para la custodia de los fondos documentales. Cuenta con funcionalidades para definir los tipos de niveles (celda, estante, anaquel, caja, etc.) empleados por la institución y su estructura jerárquica, así como para crear las distintas instancias de los mismos.

- Solicitud interna de documentación: Cada coordinador de área puede solicitar, a la Sala de Lectura, documentos para su consulta física.
- Control de préstamos de documentación: Los trabajadores y coordinadores del área Sala de Lectura pueden controlar los préstamos de documentos, mediante el registro de entrega y devolución de los mismos.
- Consulta de la documentación: Los usuarios externos pueden realizar búsquedas generales o avanzadas de acuerdo a los siguientes criterios: materia, tipología, onomástico, geográfico, institucional, fecha y expediente incorporado. Además, el sistema brinda la posibilidad de explorar la documentación existente en el Archivo. Los documentos que coincidan con los patrones de búsqueda o que fueron seleccionados mediante la exploración, si existen en formato digital, se presentan al usuario para su visualización. El usuario puede realizar la solicitud de revisión en formato físico o de algún otro servicio sobre el documento.
- Conservación: los coordinadores del área de Conservación, Preservación y Restauración Documental pueden gestionar la información referente al estado de conservación de los fondos documentales y las condiciones ambientales de los depósitos donde estos se encuentran ubicados.
- Información: El sistema brinda reportes que se pueden imprimir sobre documentos consultados, documentos descritos, documentos transcritos, usuarios registrados, usuarios atendidos, peticiones internas, aprobación de servicios, estado de conservación de los fondos documentales, resultados de los tratamientos aplicados y parámetros ambientales de los locales empleados en la custodia de documentos.

Aporte social del sistema ArchiVenHIS:

- Informatización de la gestión archivística, facilitando la descripción, acceso y preservación del patrimonio documental.
- Aumento de la eficiencia y calidad de los servicios que brida la institución que custodia los fondos documentales.
- Difusión del acervo histórico que custodian los Archivos.

- Mejor visibilidad de documentos al poder realizar tratamiento a sus representaciones digitales una vez incorporadas al sistema.
- Sencillez y rapidez para localizar la información pues no es necesaria la consulta, por parte de los usuarios, de un catálogo para localizar la documentación, las búsquedas se realizan a partir de interfaces amigables y sencillas del sistema.
- Conservación de los documentos al minimizar la manipulación directa de los documentos ya que se accede a su copia digital.

1.1.2 Persistencia de los datos en el sistema ArchiVenHIS

La información tratada por el sistema ArchiVenHIS se almacena en una base de datos administrada por el Sistema Gestor de Bases de Datos MySQL que consta de 59 tablas y vistas, once triggers y nueve funciones y procedimientos almacenados. Por otra parte, las representaciones digitales asociadas con las descripciones de los documentos se encuentran de manera física en un repositorio relativo a la dirección de instalación del sistema. A continuación se describen las estructuras más relevantes de la base de datos, del repositorio y la forma en que ambos se relacionan.

La tabla 1.1 resume la cantidad de relaciones por área temática dentro de la base de datos de ArchiVenHIS, especificando en cada área cuántas corresponden a vistas y nomencladores.

Tabla 1.1 Relaciones por área temática.

Área temática	Total de relaciones	Total de vistas	Total de nomencladores
Control de Acceso y Privilegios	16	2	4
Descripción Documental	16	3	5
Representaciones Digitales	2	0	0
Ubicación Física	4	1	2
Servicios	11	0	4
Conservación	10	0	5

En las relaciones referidas al área Control de Acceso se mantiene actualizada la información relativa a los usuarios, roles que han desempeñado, área a la que pertenecen, funcionalidades que pueden desempeñar los usuarios acorde a su rol actual y la organización de estas funcionalidades para visualizarlas en el menú de la aplicación.

En las relaciones del área Descripción Documental se mantiene actualizada la información correspondiente al proceso de descripción archivística, según lo que establece la norma ISAD(G) y dejando constancia de los usuarios que intervienen en el proceso.

En las relaciones pertenecientes a Representaciones Digitales se especifican las asociaciones entre las representaciones digitales de cada documento con cada documento en sí. Además de cada representación digital, se almacenan cuáles folios del documento representa y su estado de aprobación. En particular, la relación archivo_digital contiene información que permite obtener la dirección, relativa al repositorio de ArchiVenHIS, del fichero que contiene la representación digital. A partir de este momento se utilizará el término fichero para denotar archivo para que el lector no confunda éste con el Archivo como institución.

En las relaciones correspondientes al área Ubicación Física se almacena la información relativa a la asociación entre el documento y su ubicación física dentro de la institución.

De los Servicios se mantiene actualizada la información acerca de los flujos de trabajo que se establecen para satisfacerlos: usuarios que los solicitan, razones por la que los solicitan, las fechas de solicitud y cumplimiento, así como los trabajadores que intervienen en la solicitud (los que aprueban, los que ejecutan, etc.)

En las relaciones relativas al área Conservación se mantiene actualizada la información sobre la “Planilla para el control de temperatura, humedad e iluminación en los depósitos de los fondos documentales” y la “Ficha técnica”. Esta última incluye datos acerca del estado de conservación de los fondos documentales y de los tratamientos que se le aplican según proceda.

Sin embargo, la información relativa al acervo histórico de la nación se encuentra dispersa en varios Archivos Históricos, por lo que es necesario recolectar e integrar

estos datos que se describen a través del sistema ArchiVenHIS, en aras de garantizar su difusión de manera eficiente.

1.2 Mecanismos para la recolección e integración de datos

Para la recolección e integración de los metadatos y las representaciones digitales asociadas a los fondos documentales que se encuentren bajo la custodia de Archivos Históricos que empleen el sistema gestor de documentos históricos ArchiVenHIS para describirlos, es necesario analizar los diferentes mecanismos que pueden ser utilizados para este propósito.

El Protocolo para la Recolección de Metadatos de la Iniciativa de Archivos Abiertos OAI-PMH (5) es un estándar que se utiliza para facilitar la interoperabilidad entre sistemas, por lo que se requiere tener en cuenta esta alternativa para la recolección de los datos. Por otra parte, el sistema ArchiVenHIS utiliza MySQL, siendo conveniente analizar las potencialidades del mismo en cuanto a la replicación de los datos como posible variante a implementar.

1.2.1 Recolección de datos utilizando el estándar OAI-PMH

El protocolo estándar OAI-PMH (por sus siglas en inglés, Open Archives Initiative Protocol for Metadata Harvesting) provee un marco de trabajo que garantiza interoperabilidad para la recolección de metadatos en formato XML[1], sobre la base del protocolo HTTP[2]. Existen dos clases de participantes que intervienen en OAI-PMH: Proveedores de datos y proveedores de servicios. Los proveedores de datos administran sistemas que soportan OAI-PMH como un medio para exponer metadatos, mientras que los proveedores de servicios emplean los metadatos recolectados vía OAI-PMH como base para la construcción de servicios de valor añadido.

Asociados a OAI-PMH existen varios conceptos y definiciones que son convenientes describir para una correcta sistematización del protocolo. Un recolector es una aplicación cliente que ejecuta solicitudes OAI-PMH. Éste es operado por un proveedor de servicios como medio para la recolección de metadatos desde los repositorios. El

[1] XML: eXtensible Markup Language.

[2] HTTP: Hyper Text Transfer Protocol.

repositorio es un servidor accesible a través de la red capaz de procesar los seis tipos de solicitudes o verbos soportadas por el protocolo OAI-PMH: Identify, ListMetadataFormats, ListSets, ListIdentifiers, ListRecords, GetRecord.

Este protocolo distingue tres tipos diferentes de entidades relacionadas con los metadatos accesibles a través de él: Resource, Item y Record. Un resource o recurso es el objeto acerca del cual tratan los metadatos. La naturaleza del recurso, ya sea física o digital, si está almacenado o no en el repositorio o forma parte de otra base de datos, está fuera del alcance de OAI-PMH. Un item es un componente del repositorio a partir del cual pueden diseminarse los metadatos acerca del recurso. Conceptualmente es un contenedor que almacena o genera dinámicamente metadatos acerca de un recurso en múltiples formatos. Cada componente tiene un identificador que es único dentro del repositorio al que pertenece. Este identificador permite reconocer sin ambigüedad un componente dentro del repositorio y se utiliza en las solicitudes OAI-PMH para extraer los metadatos del componente correspondiente. El mismo juega dos roles fundamentales dentro del protocolo:

1. En las Respuestas: Los identificadores son devueltos en las respuestas a solicitudes de los tipos ListIdentifiers y ListRecords.
2. En las solicitudes: En una solicitud del tipo GetRecord se emplea un identificador en combinación con el argumento metadataPrefix para solicitar un registro en un formato específico de metadatos de un determinado componente.

Se debe reconocer que el identificador descrito no es del recurso. La naturaleza del identificador del recurso está fuera del alcance de OAI-PMH. Para facilitar el acceso al recurso asociado con los metadatos recolectados, los repositorios pueden emplear algún elemento dentro del registro de metadatos para establecer el enlace entre el registro a través del identificador del componente y el identificador del recurso asociado. El formato de Dublin Core[3], establecido como obligatorio por el protocolo, provee el elemento identificador que puede ser empleado para este propósito.

Por otra parte, un record representa un registro de metadatos en un formato específico. Se retorna en un flujo de bytes codificado en formato XML en respuesta a una solicitud

[3] Dublin Core: estándar de metadatos mantenidos por Dublin Core Metadata Iniciative (DCMI). En su forma más básica proporciona un vocabulario de quince propiedades genéricas útiles para la descripción de una amplia gama de recursos.

OAI-PMH. Se identifica por la combinación del identificador único del componente desde el cual se genera el registro, el valor de metadataPrefix identificando su formato de metadatos y una marca de tiempo, datestamp. La codificación XML de los registros se organiza en las siguientes secciones:

1. Encabezado: Compuesto por varios elementos que incluyen las propiedades para la recolección selectiva:
    a. El identificador único de un componente dentro del repositorio.
    b. La fecha de creación o modificación del registro.
    c. Cero o más elementos de tipo setSpec para indicar la pertenencia del componente a determinados conjuntos.
2. Metadatos: OAI-PMH soporta componentes con múltiples formatos de metadatos. Como mínimo, los repositorios deben tener la capacidad de retornar registros de metadatos expresados en formato Dublin Core. Opcionalmente, un repositorio puede diseminar otros formatos de metadatos. Las solicitudes del tipo ListMetadataFormats retornan una lista de todos los formatos de metadatos disponibles en el repositorio o para un ítem determinado que puede ser especificado como argumento de la solicitud ListMetadataFormats. El formato de metadatos específico de los registros que serán diseminados se indica por medio del argumento metadataPrefix en las solicitudes del tipo GetRecord y ListRecords.

Los repositorios pueden organizar sus componentes en sets. Un set es una construcción opcional para el agrupamiento de componentes en conjuntos, donde cada uno puede estar organizado en cero, uno o varios conjuntos, con el propósito de facilitar la recolección selectiva. La organización en conjuntos puede ser plana, como una lista simple, o jerárquica, siendo posible tener múltiples jerarquías con raíces distintas e independientes.

Cuando un repositorio define una organización basada en conjunto debe incluir información acerca de la membresía de los componentes en el encabezado de los registros devueltos como respuestas a solicitudes del tipo ListIdentifiers, ListRecords y GetRecord. La organización jerárquica en conjuntos se expresa acorde a la sintaxis del parámetro setSpec como se describe a continuación, pues cada nodo en una organización de conjunto de un repositorio contiene:

- setSpec: una lista separada por dos puntos [:] indicando el camino desde la raíz de la jerarquía del conjunto hasta determinado nodo. Cada elemento de la lista es una cadena consistente en una URI[4] válida que no puede contener el carácter dos puntos [:]. Las organizaciones planas solamente incluyen conjuntos con setSpec que no contienen dos puntos.
- setName: una cadena de caracteres legible que da nombre al conjunto.

El siguiente ejemplo ilustra una posible jerarquía de conjunto dentro de un repositorio:

- Instituciones
    - Universidad de Holguín "Oscar Lucero Moya"
    - Universidad Central "Marta Abreu" de la Villas
- Materias
    - Sistemas de Bases de Datos
    - Historia

La tabla 1.2 muestra una posible representación de la jerarquía anterior por medio de los valores de setName y setSpec:

Tabla 1.2 Representación de una jerarquía por medio de valores de setName y setSpec. (Fuente: elaboración propia).

setName	setSpec
Instituciones	instituciones
Universidad de Holguín "Oscar Lucero Moya"	instituciones:uho
Universidad Central "Marta Abreu" de la Villas	instituciones:uclv
Materias	materias
Sistemas de Bases de Datos	materias:sbd
Historia	materias:historia

El significado real de un conjunto o de la disposición de los conjuntos dentro de un repositorio no está definido por el protocolo. Se espera que las comunidades

[4] URI: Unified Resource Identifier.

individuales puedan formular configuraciones de conjunto adecuadas con un vocabulario controlado para setNames y setSpec, e incluso desarrollar mecanismos para presentarlos a los recolectores (5).

Una solicitud de tipo ListSets devuelve una lista indicando la configuración de los conjuntos dentro del repositorio. Cada miembro de esta lista debe incluir un setSpec y un setName y puede, además, incluir un setDescription. Las peticiones del tipo ListRecords y ListIdentifiers pueden incluir el argumento opcional set cuyo valor es el de un setSpec para especificar el conjunto objetivo de la recolección selectiva. En la jerarquía de conjunto del ejemplo mostrado en la tabla 1.2, el setSpec materias:sbd podría emplearse en una solicitud para obtener solamente aquellos registros que son diseminados a partir de componentes organizados dentro del conjunto representado por dicho setSpec. Se deben tener en cuenta cinco elementos a la hora de implementar el protocolo:

- Si un repositorio soporta conjuntos, entonces debe incluir la información relativa a la membresía en las respuestas a las solicitudes de tipo ListIdentifiers, ListRecords y GetRecord. Una lista de elementos de tipo setSpec debe incluirse solamente con el mínimo número de elementos setSpec requeridos para especificar la membresía. Según la jerarquía del ejemplo anterior, el encabezado para un componente organizado en el conjunto materias:sbd no debe incluir el setSpec materias ya que esto está implicado por el setSpec materias:sbd.
- Un componente puede estar contenido en más de un conjunto, lo que significa que diferentes argumentos de tipo setSpec pueden retornar los mismos registros.
- Un componente no necesariamente tiene que estar organizado en algún conjunto. Esto significa que una repetición exhaustiva de solicitudes de tipo ListRecords con todos los posibles setSpec como argumentos, no garantiza la obtención de todos los registros del repositorio. El único método garantizado para la recolección de todos los registros o encabezados es la realización de solicitudes de tipo ListRecords o ListIdentifiers sin argumentos setSpec.
- Cuando un setSpec se emplea como argumento, la respuesta debe incluir los registros o encabezados de todos los componentes en el conjunto especificado por el setSpec, y todos los registros y encabezados de los componentes dentro de los

conjuntos que son descendientes del conjunto especificado. Empleando la jerarquía de conjunto del ejemplo anterior, especificando el setSpec materias a una solicitud de tipo ListRecords retornará todos los registros organizados dentro del conjunto con valor de setSpec igual a materia y todos los conjuntos descendientes cuyos valores setSpec son materias:sbd y materias:historia.

- La jerarquía de conjuntos de un repositorio puede incluir conjuntos vacíos.

Determinadas solicitudes OAI-PMH retornan una lista de entidades discretas: ListRecords devuelve una lista de recursos, ListIdentifier una lista de encabezados y ListSets una lista de conjuntos. A estas solicitudes se les denomina solicitudes de listas. En algunos casos estas listas pueden ser grandes y puede resultar conveniente particionarlas entre una serie de solicitudes y respuestas. El particionamiento se consigue de la siguiente forma:

- El repositorio responde a una solicitud con una lista incompleta (sub-lista) y un resumptionToken.
- En aras de obtener la lista completa correspondiente, el recolector necesitará ejecutar una o más solicitudes con un resumptionToken como argumento. La lista completa consiste entonces en la concatenación de las sub-listas de la secuencia de solicitudes.

El resumptionToken debe ser empleado únicamente de la siguiente forma:

- El repositorio debe incluir un elemento de tipo resumptionToken como parte de cada respuesta que incluya una lista incompleta.
- Para recuperar los fragmentos sucesivos, las siguientes solicitudes emplearán como argumento resumptionToken el valor del elemento mencionado en la respuesta de la solicitud anterior.
- La respuesta que contiene el fragmento que completa la lista debe incluir un elemento de tipo resumptionToken vacío.
- Cualquier otro uso del *resumptionToken* es ilegal y debe retornar un error.

Para consultar una descripción más detallada del elemento resumptionToken véase el anexo 1.

En caso de condiciones de errores o excepciones los repositorios deben indicar los errores OAI-PMH, distinguiéndolos de los códigos de estado de HTTP, mediante la

inclusión de uno o más elementos de tipo error en la respuesta. Aunque un elemento de tipo error es suficiente para indicar la presencia de una condición de excepción, los repositorios pueden reportar todos los errores o excepciones que surjan del procesamiento de una solicitud. Cada elemento error debe tener un atributo de tipo code (véase el anexo 2 para consultar los posibles valores del atributo) y puede, además, incluir un texto para proveer información acerca del error lo cual es útil para el lector, aunque estas cadenas no están definidas por OAI-PMH. En el anexo 3 puede consultarse además la descripción de los tipos de solicitudes, o verbos, definidos por OAI-PMH.

1.2.2 Replicación de datos en MySQL

El mecanismo básico de replicación en MySQL se basa en un servidor maestro que mantiene un seguimiento de todos los cambios realizados a las bases de datos almacenadas en él, en sus archivos de log binario o bitácora. La bitácora contiene registros de todas las sentencias que modificaron ya sea la estructura de la base de datos o los datos contenidos en ella. Cada esclavo que se conecta al maestro recibe una copia de la misma y ejecuta los eventos que el mismo contiene. Esto provoca el efecto de repetir las sentencias y cambios originales tal como fueron realizados en el maestro. Las tablas se crean o su estructura se modifica y los datos se insertan, eliminan y actualizan acorde a las sentencias que fueron originalmente ejecutadas en el maestro. Cada esclavo es independiente y recibe una copia de la bitácora solicitándolo al maestro. El esclavo extrae (pull) los datos del maestro en lugar del maestro inyectar (push) los datos al esclavo. Por tanto, el esclavo es capaz de leer y actualizar la copia de la base de datos a su propio ritmo, así como iniciar y detener el proceso de replicación por decisión propia sin afectar la capacidad del maestro o de otros esclavos para actualizar el estado de sus bases de datos.

Las funcionalidades de la replicación en MySQL están implementadas empleando tres hilos, uno en el maestro y dos en el esclavo. Cuando se inicia un servidor como esclavo, este crea un hilo I/O[5] que se conecta al maestro y le solicita el envío de las actualizaciones registradas en sus bitácoras. El maestro crea un hilo para enviar el

[5] I/O: Input/Output.

contenido de la bitácora al esclavo, identificado como Binlog Dump en la salida del comando SHOW PROCESSLIST. El hilo I/O en el esclavo lee las actualizaciones que el hilo Binlog Dump del maestro envía y las copia a ficheros locales conocidos como relay logs. El tercer hilo es el SQL, que crea el esclavo para leer los relay logs y ejecutar las actualizaciones contenidas en ellos (6). MySQL ofrece mecanismos para obtener información sobre el estado de cada uno de estos hilos en los servidores maestro y esclavo. Si el maestro no escribe determinada sentencia a su bitácora, esta no se replica. Pero si la sentencia se registra, esta se envía a todos los esclavos y cada uno determina si la ejecuta o la ignora.

La decisión acerca de ejecutar o ignorar las sentencias recibidas del maestro se toma de acuerdo a las opciones --replicate-* con que fue iniciado el servidor esclavo. Éste evalúa dichas reglas empleando el procedimiento que se describe a continuación, el cual primero chequea las reglas establecidas a nivel de base de datos (etapa 1) y luego a nivel de tabla (etapa 2). En el caso más simple, cuando no se han establecido opciones --replicate-* el esclavo ejecuta todas las sentencias recibidas del maestro.

Etapa 1:

En esta primera etapa, el esclavo chequea si se han establecido opciones de los tipos --replicate-do-db o --replicate-ignore-db.

- No: Pasa a la fase de chequeo a nivel de tabla.
- Sí: Se analizan las reglas establecidas de forma análoga a la manera en que se procesan las opciones --binlog-do-db y --binlog-ignore-db para determinar si permitir o ignorar la sentencia.
    - Permitir: Pasa a la fase de chequeo a nivel de tabla.
    - Ignorar: Ignora la sentencia.

Esta etapa puede permitir que una sentencia pase a la siguiente o ignorarla. Sin embargo, las sentencias que son permitidas en esta fase no son realmente ejecutadas todavía.

Etapa 2:

Como condición preliminar, el esclavo verifica si la replicación basada en sentencias está habilitada. Si es así y la sentencia ocurre dentro de una función almacenada, esta se ejecuta y se da por terminado el procesamiento. Luego, el esclavo chequea y evalúa

las opciones de tabla. Si el servidor alcanza este punto, ejecuta todas las sentencias si no se han establecido reglas para las tablas. Si se han establecido reglas do, la sentencia debe coincidir con una de ellas para ser ejecutada, de lo contrario se ignora. Si se ha establecido alguna opción ignore, todas las sentencias se ejecutan excepto aquellas que coinciden con alguna regla ignore. En el diagrama de flujo de la figura 1.1 se precisa la forma en que ocurre esta evaluación.

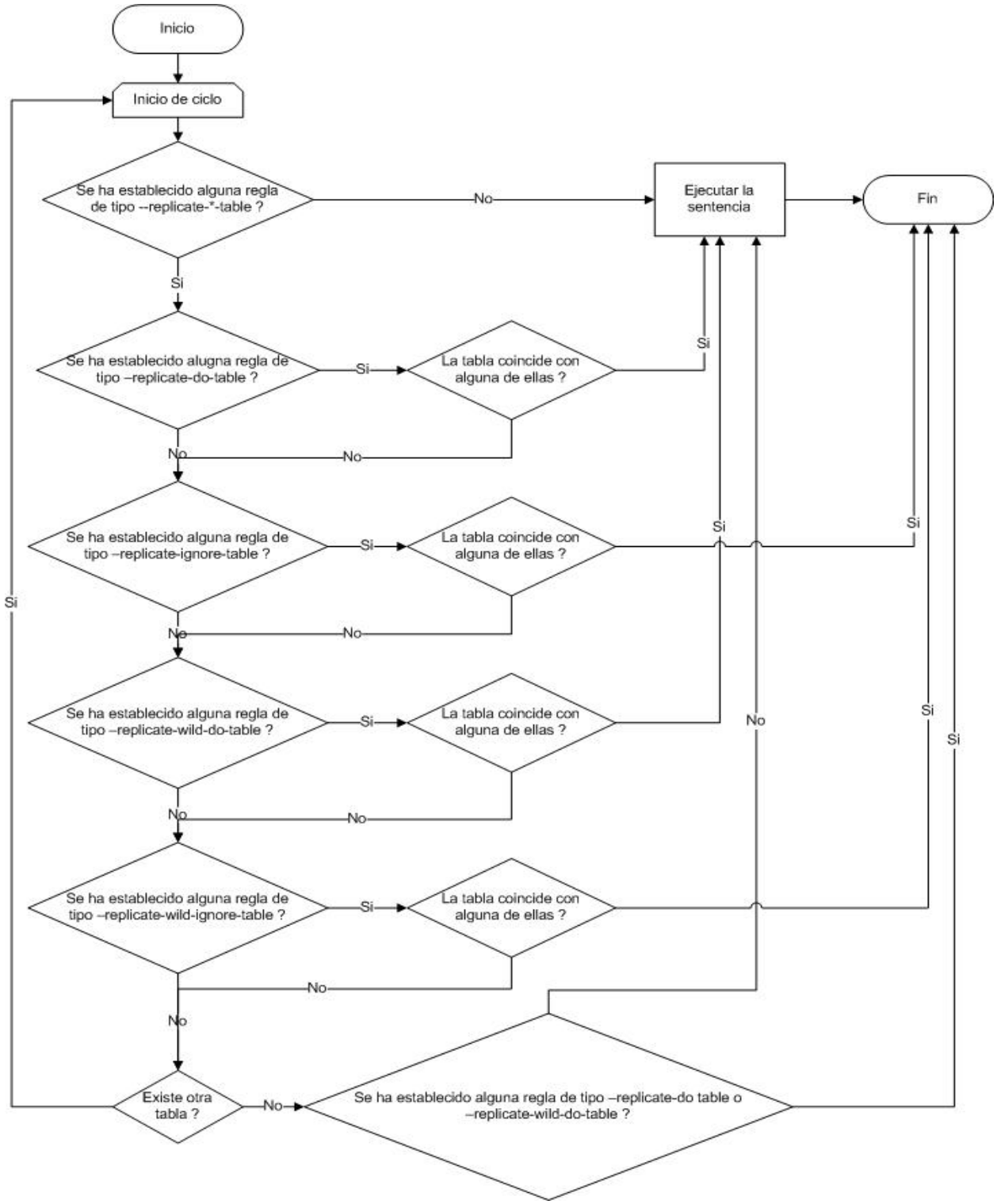

Figura 1.1 Fase 2 del procedimiento para la evaluación de las sentencias a ejecutar en el servidor esclavo. (Fuente: elaboración propia)

La bitácora de MySQL contiene todas las sentencias que actualizan datos, incluso aquellas que potencialmente podrían actualizarlos, por ejemplo una sentencia DELETE que no afecta ninguna fila. Las sentencias se almacenan en forma de eventos que describen las modificaciones. La bitácora, además, contiene información acerca del tiempo de duración de cada consulta que actualizó datos. El formato de cada uno de los eventos almacenados en la bitácora depende del formato especificado. Los tres tipos de formato soportados son: basado en fila (RBL), basado en sentencias (SBL) y mixto (MBL). Los formatos disponibles dependen de la versión de MySQL que se utilice.

- SBL: la capacidad de replicación en MySQL originalmente se basaba en la propagación de las sentencias SQL desde el maestro al esclavo. Esto se conoce como registro basado en sentencias.
- RBL: en el registro basado en fila, el servidor escribe los eventos al registro binario de forma tal que indican cómo se afectan las filas individuales de las tablas. El soporte para RBL se añadió en la versión 5.1.5 de MySQL.
- MBL: desde MySQL 5.1.8 está disponible una tercera opción: registro mixto. Con MBL, SBL se usa por defecto, pero automáticamente se cambia a RBL en determinados casos particulares, por ejemplo cuando hay una llamada a una función definida por un usuario (UDF[6]) o una llamada a una función no determinista.

Cada formato de registro binario tiene ventajas y desventajas:

Ventajas de SBL:

- Tecnología ampliamente probada, existente en MySQL desde su versión 3.23.
- Menor volumen de ficheros de bitácora generados. Significativamente menor cuando sentencias UPDATE o DELETE afectan muchas filas.
- Los archivos de la bitácora contienen todas las sentencias que realizan cambios, por tanto ellos pueden emplearse para auditar la base de datos.

Desventajas de SBL:

[6] UDF: User Defined Function.

- No todas las sentencias pueden ser replicadas pues para sentencias que emplean UDFs no deterministas no es posible replicar empleando SBL, mientras que RBL solamente replicará el valor devuelto por la UDF.
- Las UDFs deterministas deben ser aplicadas en los esclavos.

Ventajas RBL:

- Todo puede ser replicado.

Desventajas de RBL

- Ficheros de bitácora más grandes.
- Cuando se utiliza RBL para replicar sentencias, por ejemplo UPDATE o DELETE, cada fila cambiada debe ser escrita a la bitácora. Por el contrario, cuando se usa SBL solamente se registra la sentencia. Si la sentencia modifica muchas filas, RBL puede escribir significativamente más datos a la bitácora. En estos casos, la bitácora estará bloqueada por más tiempo para escribir los datos, lo cual puede causar problemas de concurrencia.
- No es posible examinar la bitácora para conocer las sentencias que fueron ejecutadas.

Es posible emplear las opciones --binlog-do-db y --binlog-ignore-db para indicar al servidor MySQL cuáles eventos registrar en la bitácora:

- --binlog-do-db=bd_nombre

  Le indica al servidor que restrinja la bitácora a las actualizaciones para las cuales la base de datos predeterminada (aquella seleccionada mediante USE) coincide con bd_nombre. El resto de las bases de datos que no se mencionan explícitamente se ignoran. Para incluir múltiples bases de datos en el proceso de réplica se debe especificar la regla para cada base de datos.
- --binlog-ignore-db=bd_nombre

  Le indica al servidor MySQL no registrar en la bitácora las actualizaciones para las cuales la base de datos predeterminada coincide con bd_nombre. Para ignorar múltiples bases de datos es preciso establecer la regla para cada una de ellas.

El diagrama de flujo de la figura 1.2 describe cómo el servidor evalúa las reglas especificadas para registrar o ignorar las actualizaciones en la bitácora. Existe una excepción para determinar la base de datos predeterminada en el caso de las

sentencias CREATE DATABASE, ALTER DATABASE y DROP DATABASE, en cuyos casos la base de datos que se está creando, modificando o eliminando reemplaza a la base de datos predeterminada.

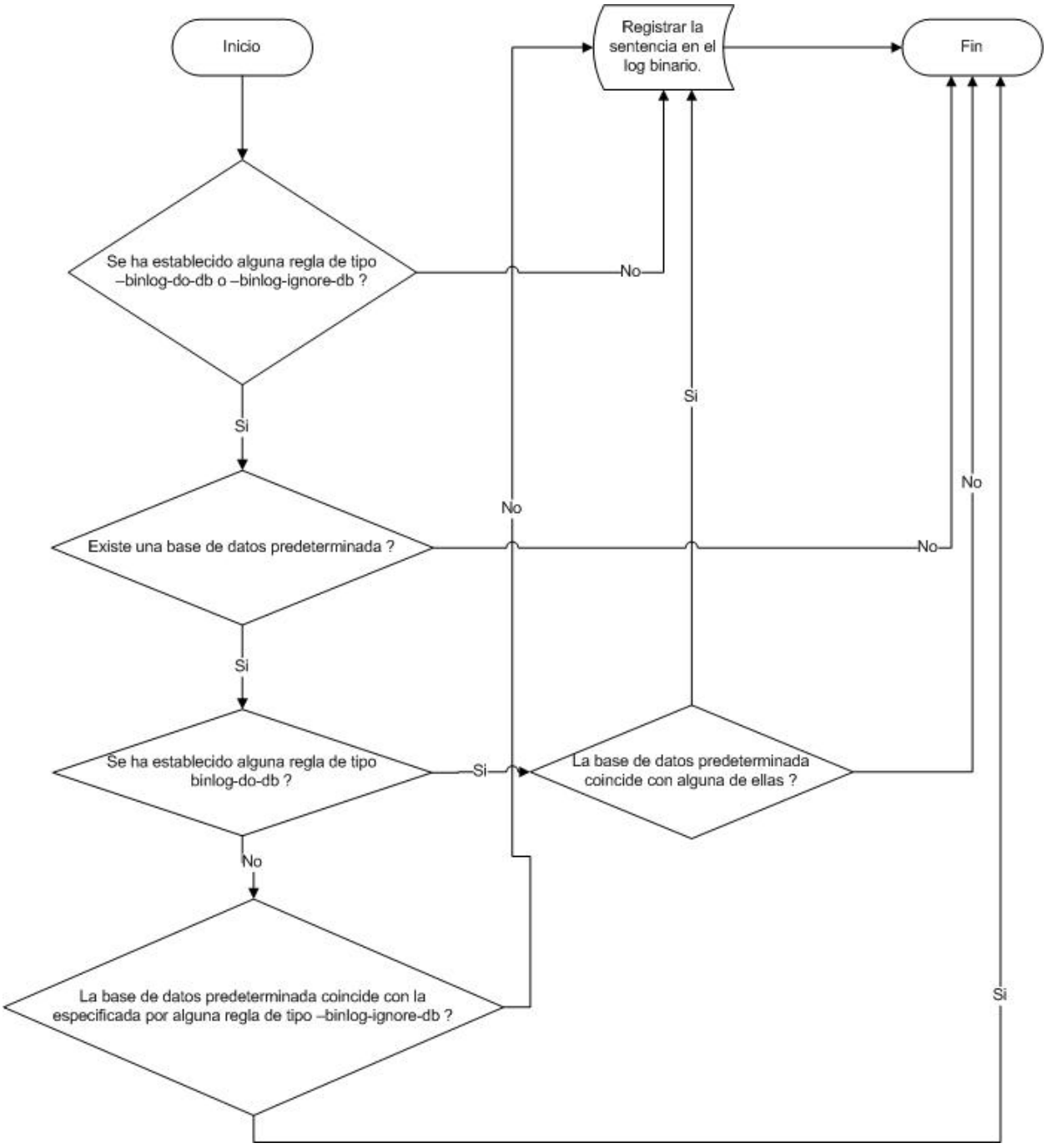

Figura 1.2 Procedimiento del servidor para evaluar las sentencias a almacenar en el log binario. (Fuente: elaboración propia)

1.2.3 Replicación de ficheros utilizando Rsync

Rsync es una herramienta de copiado de ficheros rápida y versátil. Puede copiar localmente, hacia/desde otro sitio de procesamiento sobre cualquier servidor remoto, o hacia/desde un demonio rsync remoto. Ofrece una variedad de opciones que controlan cada aspecto de su comportamiento y permite especificar de forma flexible el conjunto de ficheros a ser copiados. Se destaca por su algoritmo delta-transfer, el cual reduce la cantidad de datos a enviar a través de la red mediante el envío únicamente de las diferencias entre los ficheros fuente y los existentes en el destino. Rsync es ampliamente usado para realizar copias de resguardo y espejos y como un comando de copiado mejorado en el uso diario. Éste encuentra los ficheros que necesitan transferirse utilizando por defecto un algoritmo de chequeo rápido que verifica los archivos que han cambiado su tamaño o la última fecha de modificación.

Hay dos maneras diferentes en las que Rsync puede contactar un servidor remoto: usando un shell remoto como transporte (por ejemplo ssh o rsh) o contactando un demonio rsync directamente a través del protocolo TCP. Se emplea un servidor remoto como transporte cuando la fuente o el destino contienen sólo el carácter dos puntos (:) como separador después de la especificación del sitio. Se contacta directamente un demonio rsync cuando la fuente o el destino contienen dos caracteres dos puntos (::) como separador después de la especificación de sitio o si se especifica una URL (rsync://)

Rsync emplea para la sincronización de ficheros que ya existen en el destino el algoritmo rsync propuesto inicialmente por Andrew Tridgell en su tesis doctoral (7). Las metas de este algoritmo son:

- debe trabajar sobre datos arbitrarios, no sólo texto;
- debe ser rápido para grandes ficheros y colecciones de ficheros;
- el algoritmo no debe asumir ningún conocimiento previo acerca de los dos ficheros, pero debe tomar ventajas de sus similitudes si existieran;
- el tráfico en la red debe mantenerse al mínimo para reducir el efecto de la latencia;
- el algoritmo debe ser computacionalmente tan barato como sea posible.

A continuación se expone una descripción del algoritmo rsync:

Sean A y B dos computadoras conectadas mediante un enlace de bajo ancho de banda y alta latencia. Al iniciar la transferencia se tiene un fichero con a_i bytes en A y uno con b_i bytes en B. El propósito del algoritmo es que B reciba una copia del fichero de A.

La estructura básica del algoritmo es:

1. B envía determinados datos S a A basados en b_i.
2. A compara estos con a_i y envía ciertos datos D a B.
3. B construye el nuevo fichero usando b_i, S y D.

Para la correcta implementación de este algoritmo resulta necesario conocer qué forma tendrá S, cómo la utiliza A para compararla con a_i y cómo B reconstruye a_i. Una forma muy simple para este algoritmo es:

1. B divide b_i en N bloques b'_j del mismo tamaño y calcula la firma S_j de cada bloque. Estas firmas se envían a A.
2. A divide a_i en N bloques a'_k y calcula S'_k para cada uno.
3. A busca coincidencias de S_j con S'_k para todo k.
4. Para cada k, A envía a B el número de bloque j correspondiente al S_j que es igual a S'_k o el bloque a'_k.
5. B construye a_i usando los bloques de b_i o los bloques literales de a_i.

Esta propuesta inicial es muy simple, pero inútil en la práctica. Su problema radica en que A solamente puede encontrar coincidencias que están limitadas por los bloques. Si el fichero en A es idéntico al de B, excepto por algunos bytes que le han sido insertados al inicio, entonces no serán encontradas coincidencias en los bloques y el algoritmo transferirá todo el fichero. Este problema puede ser resuelto mediante la generación de firmas en A que no estén acotadas por los bloques, sino por cada byte. Cuando A compara la firma para cada nuevo bloque que pueda ser generado mediante un desplazamiento de un byte con las firmas S_j, será capaz de encontrar coincidencias. Esto permite manejar inserciones y eliminaciones de tamaño arbitrario entre a_i y b_i.

Esto funcionaría, pero no sería práctico debido al costo computacional de calcular una firma razonable para cada posible bloque. Podría considerarse hacer un algoritmo de firma muy barato pero esto es muy difícil de lograr sin hacer la firma demasiado débil y una firma débil haría el algoritmo inusable.

La solución, y la clave del algoritmo rsync, es no utilizar una firma por bloque sino dos. La primera muy barata de calcular y la segunda con muy baja probabilidad de colisión. La segunda firma, más costosa, solamente necesita ser calculada por A para cada bloque cuya firma barata coincide con una de las firmas baratas de B.

Si se denominan a las dos firmas R y H entonces el algoritmo queda de la siguiente forma:

1. B divide b_i en N bloques b'_j del mismo tamaño y calcula las firmas R_j y H_j para cada uno. Estas firmas se envían a A.
2. Por cada byte i en a_i A calcula R'_i sobre el bloque que comienza en i.
3. A compara R'_i con cada R_j recibido de B.
4. Por cada j donde R_j se igual R'_i A calcula H'_i y la compara con H_j.
5. Si H'_i coincide con H_j entonces a envía un token a B indicando una coincidencia de bloque y cuál bloque coincide. En caso contrario A envía los bytes literales a B.
6. B recibe los bytes literales y tokens de A y los utiliza para construir a_i.

Rsync emplea MD4[7] como algoritmo para la firma fuerte hasta su versión 3.0, a partir de la cual comienza a utilizar MD5[8]. Se plantea que este algoritmo tiene las siguientes propiedades (donde b es la cantidad de bits de la firma):

- La probabilidad de generar aleatoriamente bloques con la misma firma que un bloque determinado es $O(2^{-b})$.
- El costo computacional de encontrar un segundo bloque con la misma firma de un bloque dado es $\Omega(2^b)$.
- Los bits individuales dentro de la firma no están correlacionados y tienen una distribución uniforme.

El algoritmo seleccionado inicialmente para la firma rápida se define por:

$$r_1(k, L) = \left(\sum_{i=0}^{L-1} a_{i+k} \right) mod\ M$$

[7] MD4: Message digest, versión 4.

[8] MD5: Message digest, versión 5.

$$r_2(k,L) = \left(\sum_{i=0}^{L-1}(L-i)a_{i+k}\right) mod\ M$$

$$r(k,L) = r_1(k,L) + Mr_2(k,L)$$

Donde r(k,L) es la firma para un bloque de tamaño L con un desplazamiento de k *bytes*. M es un módulo arbitrario cuyo valor fue establecido a 2^{16}. Nótese que el resultado es una firma de 32 bit. Además la firma puede ser calculada incrementalmente de la siguiente forma:

$$r_1(k+1,L) = (r_1(k,L) - a_k + a_{k+L})\ mod\ M$$

$$r_2(k+1,L) = \left(r_2(k,L) - La_k + r_1(k+1,L)\right) mod\ M$$

$$r(k+1,L) = r_1(k+1,L) + Mr_2(k+1,L)$$

Esto permite el cálculo de los valores sucesivos con tres adiciones, dos sustracciones, una multiplicación y un corrimiento de bit, asumiendo que M es una potencia de dos (7).

1.3 Conclusiones parciales

Es necesario realizar la integración de instancias del sistema ArchiVenHIS para difundir de forma eficiente la información referente al patrimonio documental descrito a través de él y que se encuentra disperso en varios Archivos Históricos.

Tanto OAI-PMH como la replicación MySQL y rsync son alternativas viables de recolección de datos para la implementación de la integración de instancias del sistema gestor de documentos históricos ArchiVenHIS.

Puede establecerse una relación entre los conceptos recurso, componente, registro y conjunto de OAI-PMH y los que maneja el sistema gestor de documentos históricos ArchiVenHIS de la siguiente manera:

- Recurso se refiere a un documento.
- Componente es el conjunto de los metadatos consignados por la norma ISAD(G) que se describen en el sistema ArchiVenHIS para un documento dado y a partir de los cuales es posible generar registros en determinado formato.
- Registro consiste en aquellos metadatos codificados en formato XML según el estándar Dublin Core o EAD[9].

[9] EAD: *Encoded Archival Description*. Estándar XML para la codificación de metadatos archivísticos.

- Conjunto se asocia a cada una de las instancias de los niveles de organización (fondo, subfondo, serie, etc.) que conforman el cuadro de clasificación de cada Archivo Histórico que se integre a la solución.

El estándar OAI-PMH no establece un mecanismo que garantice el acceso al recurso cuando esté disponible en formato digital. En el caso de implementar la propuesta del estándar se requiere siempre descargar completamente la representación digital del documento.

La replicación de las representaciones digitales de los documentos con rsync permitiría realizar un uso más eficiente del canal de comunicación que se establece con cada Archivo integrado a la solución.

Con el mecanismo de replicación que implementa MySQL es posible garantizar el acceso total a las descripciones de los fondos documentales custodiados por los Archivos integrados. Además como valor añadido se puede tener, de manera centralizada en lugar seguro, una copia de sus bases de datos que pueden utilizarse para la recuperación en estas instituciones en casos de fallas o desastres.

Capítulo 2: Propuesta de sistema para la integración de Archivos Históricos que emplean ArchiVenHIS en la descripción del patrimonio documental

En el año 2010, como parte del contrato “CONTRATO SOLUCIÓN INTEGRAL PARA EL SISTEMA NACIONAL DE ARCHIVOS (FASE 1)”, se refleja la intención de extender personalizaciones de ArchiVenHIS a otros Archivos; por lo cual surge la necesidad de desarrollar una aplicación Web (SAHISWEB) que facilite la integración de los Archivos que emplean ArchiVenHIS en la descripción del patrimonio documental que resguardan y que además permita:

- difundirlo vía Internet a través de un punto de acceso único,
- brindar un espacio para el intercambio colaborativo entre investigadores,
- publicar las principales noticias del acontecer archivístico.

En este capítulo se configura un mecanismo apropiado y eficiente para la integración del patrimonio disperso en los Archivos Históricos que emplean ArchiVenHIS, se presentan las tecnologías utilizadas durante el desarrollo, así como los principales artefactos generados durante la implementación de SAHISWEB.

2.1 Características del sistema SAHISWEB

El sistema cuenta con funcionalidades para describir, según la norma ISDIAH[10], cada uno de los Archivos que potencialmente pudieran integrarse a SAHISWEB. Integrado o no el Archivo, el sistema permite acceder a su descripción siempre que esté disponible para los usuarios, lo cual brinda al menos información general de las instituciones descritas (nombre, ubicación, fondos que custodia, teléfono, dirección de correo electrónico, etc.). El administrador del sistema puede habilitar o deshabilitar la visibilidad sobre estas descripciones.

Una vez descritos los Archivos según la norma ISDIAH, es posible integrarlos a SAHISWEB. Con esto se consigue que los fondos documentales descritos en ellos mediante ArchiVenHIS sean accesibles a través de SAHISWEB. El sistema permite al

[10] ISDIAH: Norma Internacional para Describir Instituciones que Custodian Fondos de Archivos.

administrador habilitar o deshabilitar el acceso al patrimonio documental de los Archivos integrados.

La aplicación implementa varias funcionalidades para localizar la documentación de interés en cada uno de los archivos integrados, para cada uno de los cuales se muestra un conjunto de resultados independientemente del método de búsqueda empleado. Los métodos de búsqueda disponibles son:

- Búsqueda general: Se buscan coincidencias sintácticas entre el criterio de búsqueda especificado y los campos título y alcance y contenido descritos según la norma ISAD(G).
- Búsqueda avanzada: Además de incluir el criterio de la búsqueda general, permite especificar otros para refinarla: materia, tipología, onomástico, geográfico, fecha y nivel de organización dentro del cuadro de clasificación.
- Explorar: Permite explorar la documentación existente en el Archivo navegando a través de la estructura jerárquica que representa el cuadro de clasificación.

En todos los casos, los documentos que coincidan con los patrones de búsqueda o se localicen mediante la exploración, si existen en formato digital, serán presentados al usuario para su visualización. Además si el usuario se encuentra autenticado puede guardarlos en su espacio personal.

El módulo Espacio Personal permite a los usuarios guardar referencias de interés, organizadas por temas de investigación. Este módulo brinda las funcionalidades para definir, actualizar y eliminar los mencionados temas, así como para mantener actualizadas las referencias dentro de ellos: eliminarlas, copiarlas y moverlas. Esto facilita la consulta de la documentación que se revisa con frecuencia y evita la necesidad de buscarlas cada vez en el sistema, lo cual contribuye a su rendimiento.

Los usuarios autenticados en SAHISWEB pueden intercambiar sus experiencias e investigaciones mediante el Foro habilitado para este propósito en el sistema.

A través del módulo Noticias el administrador del sistema puede gestionar (registrar, actualizar, establecer rango de fechas de publicación y eliminar) las noticias en SAHISWEB. Una vez publicadas pueden ser accedidas por los usuarios interesados en consultarlas.

Se definen los roles: administrador, gestor de foro, moderador e investigador. Este último lo obtienen los usuarios por defecto una vez que se registran en SAHISWEB. El resto pueden ser asignados a los usuarios por uno que posea el rol de administrador.

2.1.1 Modelado del sistema

Actores del sistema

En la tabla 2.1 se realiza una breve descripción de los actores que intervienen en la Aplicación Web para facilitar el acceso, a través de Internet, a los fondos documentales de los Archivos integrados en la Solución.

Tabla 2.1 Actores del sistema SAHISWEB.

Actor	Descripción
USUARIO	Usuario sin autenticar en el portal. Puede registrarse, autenticarse, realizar búsquedas y visualizar los elementos del portal.
INVESTIGADOR	Usuario autenticado en el portal. Puede almacenar en su espacio personal los resultados de las búsquedas y acceder y participar en los foros de debate de investigación.
ADMINISTRADOR	Es el encargado de administrar el Portal. Puede describir nuevos archivos según la norma ISDIAH, así como definir los parámetros para la integración de un archivo al sistema.
MODERADOR	Usuario moderador de los foros de debate de investigación. Puede tomar medidas con los usuarios que realicen acciones indebidas en los foros, así como reorganizar los temas de debate.
GESTOR FORO	Usuario administrador del Foro. Puede realizar la gestión de los foros, así como gestionar los privilegios para los usuarios y moderadores de los mismos.

Modelo de casos de uso del sistema (CUS)

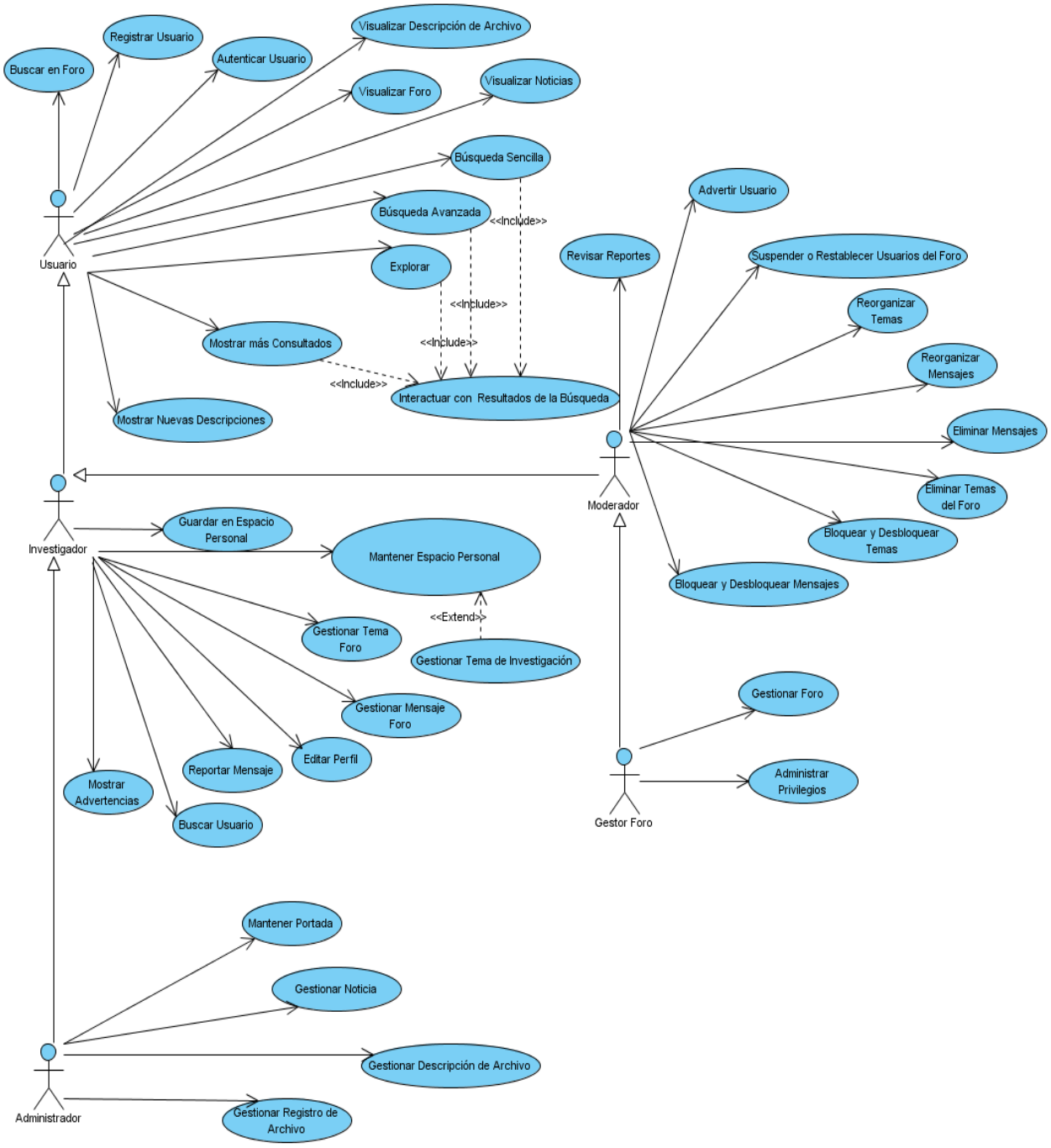

Figura 2.1 Diagrama de casos de uso del sistema

Descripción de los CUS

A continuación se presenta la descripción detallada de los CUS Gestionar Registro de Archivo, Búsqueda Avanzada e Interactuar con los resultados de las búsquedas por ser de interés para los resultados de la investigación.

Tabla 2.2 CUS Gestionar Registro de Archivo.

Caso de Uso	**Gestionar Registro de Archivo**
Actores	**Administrador**
Resumen	El caso de uso inicia cuando el actor accede a las opciones que le permiten registrar o modificar los datos de acceso a las descripciones de los fondos documentales realizadas en cada uno de los Archivos integrados en la solución. Una vez registrados se posibilita la activación o desactivación del acceso. El actor introduce los datos necesarios para registrar, modificar, activar o desactivar los datos de acceso a los Archivos, el sistema ejecuta las acciones correspondientes, finalizando así el caso de uso.
Precondiciones	El actor debe estar autenticado en la aplicación con el rol de Administrador. Para registrar los datos de un nuevo Archivo, el mismo debe haber sido descrito según la norma ISDIAH (*Ver CUS Gestionar Descripción de Archivo*).
Referencias	**RF 7, RF 7.1, RF 7.2, RF 7.3**
Prioridad	**Crítica**
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a la opción para gestionar el acceso a la información de los fondos documentales proporcionada por cada uno de los Archivos que integran la solución.	2. Muestra interfaz que permite registrar los datos de acceso a la información descrita sobre los fondos documentales de un Archivo integrado en la solución. Consulta y muestra el listado de los Archivos cuyos datos de acceso han sido registrados en el sistema, junto a las opciones para modificarlos, desactivarlos o activarlos.

	a) Para registrar los datos de acceso a la información descrita sobre los fondos documentales de un Archivo integrado en la solución, ver Sección “Registrar datos de acceso”. b) Para modificar los datos de acceso a la información de un Archivo, ver Sección “Modificar datos de acceso”. c) Para activar o desactivar el acceso a la información de un Archivo, ver Sección “Desactivar o activar acceso”.
Sección “Registrar datos de acceso”	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. Muestra interfaz que permite especificar los datos de acceso a la información descrita sobre los fondos documentales de un Archivo integrado en la solución: Archivo descrito según la norma ISDIAH, Usuario, Contraseña, nombre de la Base de Datos, IP del servidor, puerto de red y directorio donde se encuentran almacenadas las representaciones digitales de los documentos.
2. Selecciona el Archivo al cuál pertenecen los datos de acceso.	3. Verifica que no existan datos de acceso asociados al Archivo seleccionado.
4. Introduce los datos de acceso y presiona el botón para registrarlos.	5. Verifica que se hayan especificado todos los datos de acceso.
	6. Verifica que la dirección IP tenga un formato válido.
	7. Verifica que el número de puerto especificado sea válido.
	8. Verifica que los datos proporcionados garantizan el acceso a la información de los fondos documentales descritos en el Archivo seleccionado.
	9. Registra los datos de acceso a la

	información de los fondos documentales del Archivo especificado y le notifica al actor, con el mensaje: “Se han registrado los datos de acceso al Archivo satisfactoriamente.”, finalizando así el caso de uso.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	3.1. Permite editar los datos de acceso correspondientes al Archivo seleccionado, ver Sección “Modificar datos de acceso”.
	5.1. Notifica al actor que faltan por introducir datos obligatorios, con el mensaje de error: “El campo *nombre del campo* es obligatorio.”. 5.2. Retorna al paso 2 del flujo normal de los eventos.
	6.1. Notifica al actor sobre el error cometido, con el mensaje: “Debe especificar la dirección IP en un formato válido.”. 6.2. Retorna al paso 2 del flujo normal de los eventos.
	7.1. Notifica al actor sobre el error cometido, con el mensaje: “Debe especificar un número de puerto válido.”. 7.2. Retorna al paso 2 del flujo normal de los eventos.
	8.1. Notifica al actor sobre el error cometido, con el mensaje: “No es posible acceder a la información del Archivo seleccionado con los datos de acceso especificados.”. 8.2. Finaliza el caso de uso.
Sección “Modificar datos de acceso”	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**

1. Selecciona la opción para modificar los datos de acceso a la información descrita sobre los fondos documentales de un Archivo específico integrado en la solución.	2. Consulta y muestra los datos de acceso del Archivo seleccionado: Usuario, Contraseña, nombre de la Base de Datos, IP del servidor puerto de red y directorio donde se encuentran almacenadas las representaciones digitales de los documentos.
3. Edita los datos deseados y presiona el botón para modificarlos.	4. Verifica que se hayan especificado todos los datos de acceso.
	5. Verifica que la dirección IP tenga un formato válido.
	6. Verifica que el número de puerto especificado sea válido.
	7. Verifica que los datos proporcionados garantizan el acceso a la información de los fondos documentales descritos en el Archivo seleccionado.
	8. Modifica los datos de acceso a la información de los fondos documentales del Archivo especificado y le notifica al actor, con el mensaje: “Se han modificado los datos de acceso al Archivo satisfactoriamente.”, finalizando así el caso de uso.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
	4.1. Notifica al actor que faltan por introducir datos obligatorios, con el mensaje de error: “El campo *nombre del campo* es obligatorio.”. 4.2. Retorna al paso 3 del flujo normal de los eventos.
	5.1. Notifica al actor sobre el error cometido, con el mensaje: “Debe especificar la dirección IP en un formato

	válido.”. 5.2. Retorna al paso 3 del flujo normal de los eventos.
	6.1. Notifica al actor sobre el error cometido, con el mensaje: “Debe especificar un número de puerto válido.”. 6.2. Retorna al paso 3 del flujo normal de los eventos.
	7.1. Notifica al actor sobre el error cometido, con el mensaje: “No es posible acceder a la información del Archivo seleccionado con los datos de acceso especificados.”. 7.2. Finaliza el caso de uso.
Sección “Desactivar o activar Acceso”	
Acción del Actor	**Respuesta del Sistema**
1. Selecciona la opción para *Activar/Desactivar* el acceso a la información de los fondos documentales de un Archivo determinado.	2. El sistema solicita confirmación de la acción a realizar, con el mensaje: “¿Está seguro que desea *Activar/Desactivar* el acceso al Archivo seleccionado?”.
3. Confirma la acción a realizar.	4. Activa/Desactiva el acceso al Archivo seleccionado y le notifica al actor, con el mensaje: “El acceso al Archivo ha sido *activado/desactivado* satisfactoriamente.”, finalizando así el caso de uso.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
3.1. Anula la acción, finalizando así el caso de uso.	
Poscondiciones	Se actualizan los datos de acceso a los Archivos integrados en la solución.

Tabla 2.3 CUS Búsqueda Avanzada.

Caso de Uso	**Búsqueda Avanzada**
Actores	**Usuario**

Resumen	El actor del sistema accede a la interfaz de usuario para la búsqueda avanzada, especifica los campos sobre los que desea buscar (“Archivo”, “Nivel de organización de los fondos documentales dentro del Archivo seleccionado“, “Título”, “Fecha”, “Expediente incorporado”, “Geográfico”, “Tipología”, “Institucionales”, “Materia” y “Onomástico”), y el sistema muestra un listado de los niveles de organización que coincidan con los criterios de búsqueda definidos por el actor, terminando así el caso de uso.
Precondiciones	
Referencias	**RF 9, RF 9.1, RF 9.2, RF 9.3**
Prioridad	**Crítica**

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a la opción para realizar una búsqueda avanzada.	2. Muestra interfaz que permite especificar los elementos por los que se puede acotar el resultado de la búsqueda (“Archivo”, “Nivel de organización de los fondos documentales dentro del Archivo seleccionado“, “Título”, “Fecha”, “Expediente incorporado”, “Geográfico”, “Tipología”, “Institucionales”, “Materia” y “Onomástico”).
3. Introduce los criterios por los que desea filtrar y presiona el botón para ejecutar la acción.	4. Verifica que se ha especificado al menos un criterio para la búsqueda.
	5. Verifica el formato del campo fecha (aaaa-mm-dd).
	6. Consulta y muestra los niveles de organización que coinciden con los criterios especificados junto a las opciones para interactuar con los resultados obtenidos. (*Ver CUS Interactuar con Resultados de la Búsqueda*)
7. Consulta los niveles de organización presentados como resultado de la búsqueda.	8. Finaliza el caso de uso.
Flujos Alternos	

Acción del Actor	Respuesta del Sistema
	4.1. Muestra una notificación indicando que no se ha especificado ningún criterio de búsqueda, con el mensaje: “Debe especificar algún criterio de búsqueda.”. 4.2. Retorna al paso 3 del flujo normal de los eventos.
	5.1. Notifica al actor que el formato para el campo fecha es incorrecto, con el mensaje de error: “Debe especificar la fecha en el formato válido (aaaa-mm-dd).”. 5.2. Regresa al paso 3 del flujo normal de los eventos.
	6.1. Notifica al actor que no se han encontrado resultados acorde a los criterios de búsqueda especificados, con el mensaje: “La búsqueda no arrojó resultados para los criterios especificados.”, finalizando así el caso de uso.
7.1. Selecciona un nivel para consultar: su ubicación, su descripción o, en caso que sea un documento, su representación digital.	7.1.1. Permite la interacción con el nivel seleccionado según la información que solicite. Ver *CUS Interactuar con Resultados de la Búsqueda*. 7.1.2. Regresa al paso 7 del flujo normal de los eventos.
Poscondiciones	

Tabla 2.4 CUS Interactuar con Resultados de la Búsqueda.

Caso de Uso	**Interactuar con Resultados de la Búsqueda**
Actores	**Usuario**
Resumen	A partir de los resultados obtenidos a través de una búsqueda o de una exploración de niveles de organización, el actor accede a las opciones de un nivel de organización específico para visualizar su descripción, su ubicación lógica o su representación digital en caso de ser un documento. El sistema responde a la solicitud realizada, finalizando así

	el caso de uso.
Precondiciones	
Referencias	**RF 11, RF 11.1, RF 11.2, RF 11.3**
Prioridad	**Crítica**

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Selecciona alguna de las opciones para interactuar con un nivel de organización determinado.	a) Si accede a la opción para ver la descripción del nivel de organización seleccionado, ver Sección "Visualizar descripción del nivel de organización". b) Si selecciona la opción para visualizar la ubicación lógica del nivel de organización seleccionado, ver Sección "Visualizar ubicación del nivel de organización". c) Si selecciona la opción para visualizar la representación digital del documento seleccionado, ver Sección "Visualizar representación digital del documento".

Sección "Visualizar descripción del nivel de organización"	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a la opción para visualizar la descripción del nivel de organización seleccionado.	2. Consulta y muestra la descripción correspondiente al nivel de organización seleccionado, con los campos Título, Fechas, Nivel de ubicación lógica, Expediente incorporado, Geográfico, Tipología, Institucionales, Materia, Onomástico, Productor, Alcance y contenido.
	3. Finaliza el caso de uso.

Sección “Visualizar ubicación del nivel de organización”	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a la opción para visualizar la ubicación lógica del nivel de organización seleccionado.	2. Muestra el árbol de niveles que especifica la ubicación lógica del nivel de organización seleccionado.
	3. Finaliza el caso de uso.
Sección “Visualizar representación digital del documento”	
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
1. Accede a la opción para visualizar la representación digital de un documento seleccionado.	2. Muestra un listado con las representaciones digitales del documento seleccionado.
3. Selecciona un elemento del listado para visualizarlo.	4. Muestra el cuadro de diálogo para la descarga del archivo correspondiente al navegador.
	5. Finaliza el caso de uso.
Flujos Alternos	
Acción del Actor	**Respuesta del Sistema**
3.1. Consulta los elementos del listado, finalizando así el caso de uso.	
Poscondiciones	

2.2 Implementación del sistema SAHISWEB

A continuación se precisa el ambiente de desarrollo, el cual puede describirse a través de las siguientes herramientas y tecnologías libres, todas basadas en estándares abiertos:

- Sistema operativo: Debian 6

- Control de versiones: Subversion
- Gestión del proyecto: Redmine
- Herramienta de modelado para UML[11]: Visual Paradigm 6.4 Edición Comunitaria
- Entorno de desarrollo integrado (IDE): Netbeans 6.9 para PHP
- Lenguajes del lado del cliente: HTML, CSS, Javascript
- Lenguaje de programación del lado del servidor: PHP 5.3
- Servidor Web: Apache 2.2
- Sistema gestor de bases de datos (SGBD): MySQL 5.1
- Marco de trabajo (framework): CMS[12] Drupal 6.20
- Gestor de foros: phpBB 3.0

Se utiliza el CMS Drupal debido a las facilidades que este ofrece para la implementación de varios requisitos funcionales, por ejemplo los referidos a la gestión de usuarios y roles, así como el mantenimiento de la portada. Por otra parte posibilita la integración, de forma sencilla, con el gestor de foros phpBB utilizando el módulo phpBBforum, el cual garantiza que ambos sistemas compartan una autenticación única. Además, la personalización del tema de phpBB acorde al de Drupal, estando ambos integrados a partir de la autenticación compartida, permite dar al usuario la sensación de navegar en un mismo sistema: SAHISWEB.

A pesar de las ventajas que proporciona el CMS Drupal para el desarrollo de aplicaciones Web, fue necesario implementar varios módulos entre los que destacan los siguientes:

- conexión_archivos: Módulo responsable de gestionar los parámetros de conexión a las fuentes de datos de los archivos integrados a la solución previamente descritos según la norma ISDIAH.
- buscar: Este módulo abarca las funcionalidades de búsqueda, tanto sencilla como avanzada, al igual que explorar los fondos documentales de los archivos integrados y la interacción con los resultados de las búsquedas.
- gestionar_tema_investigacion: Implementa las funcionalidades para mantener actualizados los temas de investigación dentro del espacio personal.

[11] UML: Lenguaje unificado de modelado.

[12] CMS: *Content Management System*.

- espacio_personal: Este módulo implementa las funcionalidades necesarias para mantener actualizadas las referencias dentro de los temas de investigación del espacio personal.
- mas_consultados: Implementa las funcionalidades para mostrar un listado de hasta diez de los documentos más consultados.

2.2.1 Persistencia de los datos en el sistema SAHISWEB

Drupal emplea para mantener la persistencia de los datos durante su funcionamiento un total de 150 relaciones. De ellas 62 son intrínsecas de phpBB y 82 de Drupal. El resto, mostradas en la figura 2.2, se corresponden con extensiones a la base de datos necesarias para almacenar la información asociada con el registro de los Archivos integrados, el espacio personal y la consulta del patrimonio documental.

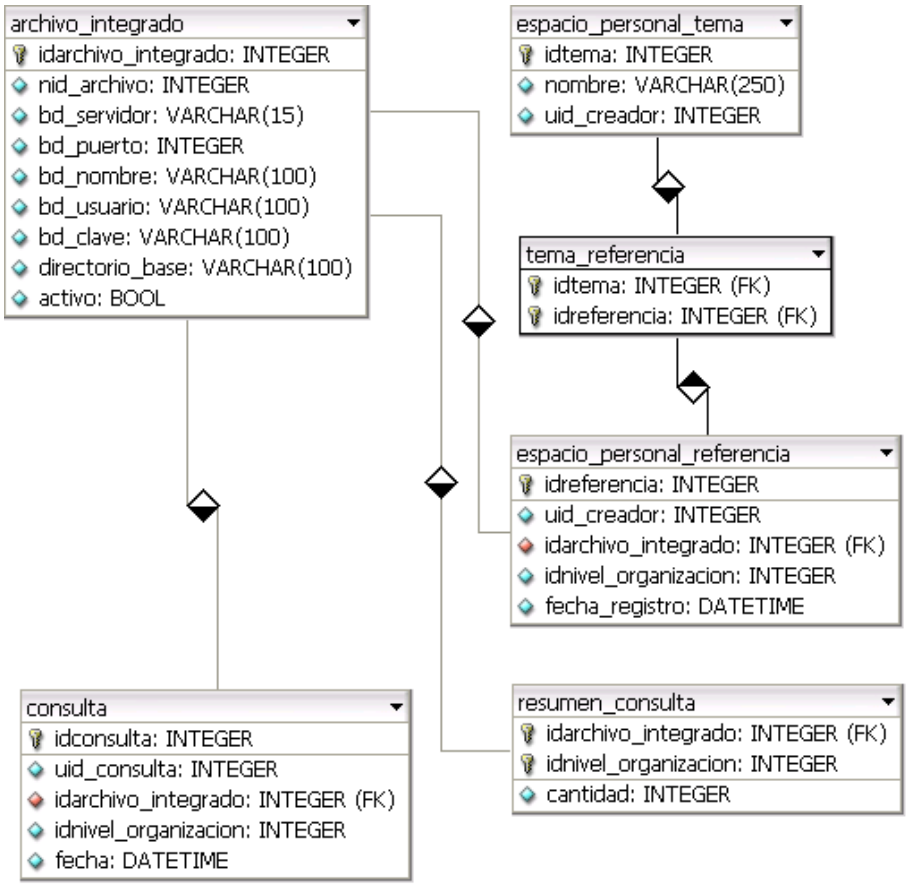

Figura 2.2 Extensión de la base de datos de SAHISWEB.

A continuación se describen cada una de las relaciones expuestas en el esquema de la figura 2.2, para una mejor compresión de su estructura, así como de la necesidad de su

uso. En la relación resumen_consulta se precisa la manera en que se establece el resumen de las consultas sobre el patrimonio documental a través de un trigger.

Tabla 2.5 Relación archivo_integrado.

Nombre: archivo_integrado		
Descripción: Contiene los datos de acceso a la información replicada desde cada uno de los archivos integrados a la solución.		
Atributo	Tipo	Descripción
idarchivo_integrado	int(10)	Identificador de la relación archivo_integrado.
nid_archivo	int(10)	Identificador que hace referencia a la instancia del tipo de contenido Descripción de Archivo (ISDIAH), que almacena la descripción del Archivo integrado.
bd_servidor	varchar(15)	Dirección IP donde se encuentra accesible el servidor de base de datos que contiene la réplica de la base de datos correspondiente a la instancia de ArchiVenHIS empleada por el Archivo integrado a la solución.
bd_puerto	int(10)	Puerto donde el servidor de bases de datos (bd_servidor) acepta conexiones TCP/IP.
bd_nombre	varchar(100)	Nombre de la base de datos replicada.
bd_usuario	varchar(100	Usuario para acceder a la base de datos replicada.
bd_clave	varchar(100)	Contraseña asignada al usuario (bd_usuario) para acceder a la base de datos replicada.
directorio_base	varchar(100)	Nombre del directorio, relativo al repositorio, donde se encuentran las representaciones digitales asociadas a las descripciones de los documentos, que han sido replicadas con rsync desde el Archivo integrado.
activo	tinyint(1)	Representa si la información replicada desde el archivo integrado está disponible para la consulta pública o no.

Tabla 2.6 Relación consulta.

Nombre: consulta		
Descripción: Contiene información relativa a las consultas, realizadas por los usuarios e investigadores, sobre el patrimonio de los Archivos integrados a SAHISWEB.		
Atributo	Tipo	Descripción
idconsulta	int(10)	Identificador de la relación consulta.
uid_consulta	int(10)	Hace referencia al usuario que realiza la consulta.
idarchivo_integrado	int(10)	Llave foránea que hace referencia a la relación archivo_integrado.
idnivel_organizacion	int(10)	Hace referencia al nivel de organización del patrimonio documental consultado el cual se encuentra almacenado en la base de datos replicada desde un Archivo integrado.
fecha	datetime	Fecha en que se realiza la consulta.

Tabla 2.7 Relación resumen_consulta.

Nombre: resumen_consulta		
Descripción: Contiene información resumida de la relación consulta, en aras de hacer más eficiente la implementación del CUS Mostrar más consultados. La relación se consulta a nivel de aplicación y sólo se actualiza a nivel de base de datos mediante un trigger que se dispara después de cada inserción en la relación consulta. El trigger verifica si existe un registro en resumen_consulta correspondiente al nivel de organización y Archivo integrado consultados y, si existe, añade 1 a la cantidad de consultas, de lo contrario crea un nuevo registro con una cantidad de consultas igual a 1.		
Atributo	Tipo	Descripción
idarchivo_integrado	int(10)	Llave foránea que hace referencia a la relación archivo_integrado.
idnivel_organizacion	int(10)	Hace referencia al nivel de organización del patrimonio documental consultado el cual se encuentra almacenado en la base de datos replicada desde un Archivo integrado.
cantidad	int(10)	Total de consultas realizadas sobre un nivel de organización perteneciente a un Archivo integrado determinado.

Tabla 2.8 Relación espacio_personal_tema.

Nombre: espacio_personal_tema		
Descripción: Contiene los temas de investigación del espacio personal de los investigadores.		
Atributo	Tipo	Descripción
idtema	int(10)	Identificador de la relación espacio_personal_tema.
nombre	varchar(250)	Hace referencia al nombre del tema de investigación perteneciente al espacio personal del uid_creador.
uid_creador	int(10)	Hace referencia al investigador que crea el tema en su espacio personal.

Tabla 2.9 Relación espacio_personal_referencia.

Nombre: espacio_personal_referencia		
Descripción: Contiene los temas de investigación del espacio personal de los investigadores.		
Atributo	Tipo	Descripción
idreferencia	int(10)	Identificador de la relación espacio_personal_referencia.
uid_creador	int(10)	Se corresponde con el identificador del investigador que guarda la referencia en su espacio personal.
idarchivo_integrado	int(10)	Llave foránea que hace referencia al identificador del Archivo integrado que contiene el documento cuya referencia se desea almacenar en el espacio personal.
idnivel_organizacion	int(10)	Hace referencia al documento consultado cuya referencia se desea almacenar en el espacio personal.
fecha_registro	datetime	Fecha en que se guarda la referencia en el espacio personal.

Tabla 2.10 Relación tema_referencia.

Nombre: tema_referencia
Descripción: Relación muchos a muchos entre espacio_personal_referencia y espacio_personal_tema que indica los temas de investigación del espacio personal al que

pertenecen las referencias de documentos guardadas.		
Atributo	Tipo	Descripción
idtema	int(10)	Llave foránea que referencia la relación espacio_personal_tema.
idreferencia	int(10)	Llave foránea que referencia la relación espacio_personal_referencia.

Además de las relaciones específicas del CMS Drupal, las relativas a phpBB y las extendidas para SAHISWEB, el sistema requiere consultar información de los Archivos integrados para la ejecución de las búsquedas. Es por ello que es necesario incluir otras relaciones que intervienen en el proceso, de ellas varias son nomencladores, otras mantienen la información de las descripciones realizadas según la norma ISAD(G) a los fondos documentales de los Archivos integrados a la solución y contienen la información relativa a las representaciones digitales de los documentos descritos a través de ArchiVenHIS, así como del cuadro de clasificación (véase el anexo 4 para consultar la descripción detallada de las principales relaciones del sistema ArchiVenHIS que intervienen en el proceso de búsqueda).

2.2.2 Integración de Archivos a la solución

Para integrar un Archivo al sistema SAHISWEB es necesario que exista conectividad TCP/IP[13] entre el Archivo a integrar y el centro de datos donde se hospeda SAHISWEB, de ahora en adelante se denominará nodo central, para dar soporte a la replicación de datos entre ellos. Se requiere además que el Archivo a integrar emplee el sistema ArchiVenHIS para realizar la descripción de los fondos documentales bajo su custodia, los cuales desea publicar en Internet a través de SAHISWEB.

Los datos a replicar desde el Archivo a integrar son:

- La base de datos de ArchiVenHIS, en este caso se tendrá un dúo de servidores maestro-esclavo por cada Archivo a integrar.
- Las representaciones digitales de los documentos, ubicadas en el directorio repositorio de la instancia de ArchiVenHIS instalada en el Archivo. Esta replicación se hará mediante rsync.

[13] TCP/IP: Transmission Control Protocol/ Internet Protocol.

Replicación de la base de datos:

El servidor MySQL del nuevo Archivo actuará como maestro (root@master, IP 10.1.1.101) y se designará un servidor MySQL en el nodo central que funcionará como esclavo (root@slave , IP 10.2.2.202).

Configuración del servidor maestro

Debe habilitarse en el servidor maestro una cuenta de usuario necesaria para que el servidor esclavo pueda establecer una conexión. Para ello, una vez conectados a MySQL con privilegios suficientes para crear una nueva cuenta, se ejecuta el siguiente comando:

```
root@master:/etc/mysql# mysql -u root -p
mysql> GRANT REPLICATION SLAVE ON *.* TO 'user_réplica'@10.2.2.202 IDENTIFIED BY 'clave_réplica';
```

Deben estar activas las opciones *log-bin [=/camino/al/log]*, donde */camino/al/log* define la ubicación y nombre de los archivos de registro binario, y *server-id=1* el cual puede ser cualquier número entre 1 y 2^32 – 1 y debe ser único para el maestro y cada uno de los esclavos. Además se utiliza la opción *binlog-do-db=BD,* donde *BD* especifica el nombre de la base de datos que se desea replicar. Estas opciones deben ser añadidas o modificadas en el fichero /etc/mysql/*my.cnf*.

```
log_bin = /var/log/mysql/mysql-bin.log //Camino a la ubicación de los logs binarios del servidor
 //MySQL
binlog_do_db = archivenhis // Base de datos a replicar
server-id=1 // Identificador único para el servidor en el entorno de replicación
```

Resulta necesario realizar una copia de seguridad de la estructura y datos contenidos en la base de datos a replicar (archivenhis). Esto puede conseguirse con el siguiente comando, donde root representa el usuario mediante el cual se realizará el volcado de los datos y */opt/bdatos_backup.sql* la dirección al fichero que contendrá las sentencias SQL que representan la copia de seguridad.

```
root@master:~# mysqldump --master-data=2 --routines -u root -p archivenhis > /opt/bdatos_backup.sql
```

En este fichero se guarda una copia de todas las sentencias SQL que permiten restaurar la base de datos en los servidores esclavos. Además de la información que

indica el punto de inicio de la replicación, dada por un archivo específico de log binario en el maestro y la posición dentro de este último a partir de la cual se debe obtener la información a replicar. Si se abre el fichero de salva con un editor de textos es posible consultar esta información (véase el anexo 5.1).

Por último se debe copiar hacia el servidor esclavo la salva de seguridad obtenida del maestro en una ubicación determinada e importar el contenido del fichero.

```
root@master:~# scp /opt/bdatos_backup.sql root@10.2.2.202:/opt/
```

Configuración del Servidor esclavo

Es necesario que esté activa la opción server-id=n. En este caso además se emplearán las opciones replicate-do-db=BD para asegurar que sólo se replicarán los datos de la base de datos de interés. Estas opciones deben ser añadidas o modificadas en el fichero my.cnf.

```
server-id = 2
replicate-do-db= archivenhis
#Conectar como root al MySQL
root@slave: ~# mysql -u root –p
mysql> create database archivenhis;       //Crear la base de datos vacía donde se importarán los datos
Query OK, 1 row affected (0.00 sec)
mysql> use archivenhis                    //Para usar la nueva base de datos
Database changed
mysql> source /opt/bdatos_backup.sql //Importar los datos desde el archivo copiado del
 //servidor maestro
```

Después de importados los datos se debe configurar en el servidor esclavo otros parámetros como la dirección IP del servidor Maestro, el usuario que usará para conectarse con el Maestro, la contraseña que usará este usuario, el fichero de logs binario a leer, la posición dentro de este log a leer y el tiempo en segundos que el esclavo esperará entre reintentos de conexión con el Maestro en caso de fallas.

```
mysql> CHANGE MASTER TO
MASTER_HOST='10.1.1.101', //IP del servidor maestro
MASTER_USER='user_replica', //usuario para acceder al maestro para realizar la replicación
MASTER_PASSWORD='clave_replica', //contraseña del mencionado usuario
MASTER_LOG_FILE='mysql-bin.000004', //log en el maestro a leer para replicar, esta información se
```

//puede encontrar en el fichero de la copia de seguridad
MASTER_LOG_POS=106, //posición dentro del log a leer para replicar, esta información
//se puede encontrar en el fichero de la copia de seguridad
MASTER_CONNECT_RETRY = 10; //tiempo en segundos entre reintentos de conexión hacia el
//MASTER

#Iniciar el flujo de replicación en el esclavo
mysql> START SLAVE;

#Para ver el estado de la replicación en el esclavo (véase el anexo 5.2 para consultar la salida del comando)
mysql> show slave status\G

Por último es necesario disponer de un usuario en el servidor esclavo que le permita a SAHISWEB acceder a la base de datos replicada desde el Archivo remoto.

root@slave: ~# mysql -u root –p
#Creación de un usuario con privilegios de acceso a la réplica de la base de datos archivenhis desde el servidor donde se encuentra SAHISWEB. En la sentencia siguiente 10.2.2.200 representa la dirección IP de este servidor.
mysql> GRANT ALL PRIVILEGES ON archivenhis.* TO 'integrado1'@'10.2.2.200' IDENTIFIED BY 'integrado1';

Replicación de las representaciones digitales de los fondos documentales

La replicación de las representaciones digitales de los documentos tiene lugar entre el servidor Web donde se hospeda la instancia de ArchiVenHIS empleada por el Archivo a integrar (root@archivenhis, IP 10.1.1.100) y el servidor Web donde se aloja SAHISWEB en el nodo central (root@sahisweb, IP 10.2.2.200). En el primero las representaciones digitales de los documentos se encuentran en el directorio repositorio dentro de la ubicación donde está instalado ArchiVenHIS. Estas serán replicadas hacia un directorio que debe ser definido dentro del repositorio de SAHISWEB, ubicado en *sites/default/files/repositorio* que es una dirección relativa al directorio donde está instalado SAHISWEB.

Para que la replicación tenga lugar es necesario tener instalado en ambos OpenSSH y rsync, así como la existencia de un usuario en el servidor donde se aloja ArchiVenHIS para que el servidor central se conecte y pueda leer las representaciones digitales de los documentos.

El procedimiento para la creación de nuevos usuarios en el sistema operativo (Debian 6) se describe a continuación. Sólo se debe especificar el nombre del usuario, usualmente alguno relacionado con la función que tendrá.

Servidor remoto:

```
root@archivenhis:/# adduser rsync_central
Adding user ` rsync_central ' ...
Adding new group ` rsync_central ' (1002) ...
Adding new user ` rsync_central ' (1002) with group ` rsync_central ' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for rsync_central
Enter the new value, or press ENTER for the default
 Full Name []:
 Room Number []:
 Work Phone []:
Home Phone []:
 Other []:
Is the information correct? [Y/n] y
```

Es importante tener en cuenta que en el proceso de creación de los usuarios se debe configurar una contraseña que será utilizada una única vez. Después de creado el usuario en el servidor remoto, a este usuario se le debe deshabilitar la contraseña. Esto debe realizarse sólo después de ejecutar correctamente el paso número cuatro del procedimiento en el servidor central que se describe a continuación:

1. Bloquear la contraseña del usuario creado en el sistema.

```
root@archivenhis:/#passwd rsync_central -l
 passwd: password expiry information changed.
root@archivenhis:/#
```

2. Editar el archivo authorized_keys que se encuentra en el directorio .ssh del usuario creado para el archivo central.

ssh-rsa
AAB3NzaC1yc2AQ6PGQZNn85bLO+e2NQrQsyrxeNhlN398aQJTzsrYRSFrFDSv9xg8Pi
+F5kmRzoOD6ORTk9+soje8WjHngptx7xihdaH9gRa4E3LpLZcAQFlOgAiU2SVeELlo52n
xS8m2jhPfF3eIJ9137jr78SMl4dGVxjwPgffrUHpXe2HLjE66szjoW67eGuFG23d/lbTNJQq
6w0EGTz0u6x/x8eFvma8+VOFJi8spKqodezt/ogfXv79tksdzlvQzYHIFz8IIXpLZBq/SgPieE
FDWLNRQYuMjXYOgo0CZXq7g3iORd2n60tysAoCdX+wNzfxv root@sahisweb

Se añaden los siguientes parámetros delante de “ssh-rsa”:

from="10.2.2.200",no-port-forwarding,no-X11-forwarding,no-agent-forwarding,no-pty

Con los dos pasos anteriores se evita que el usuario pueda autenticarse en el sistema utilizando la contraseña y ejecutar comandos de forma interactiva. Se limita el acceso y sólo es posible desde la dirección IP definida, en este caso 10.2.2.200**.**

En el servidor central se deben ejecutar seis pasos para completar la replicación de los documentos.

1. Crear el directorio donde se alojarán las representaciones digitales replicadas

root@sahisweb:/#mkdir /var/www/sahisweb/sites/default/files/repositorio/nuevoarchivo

2. Generación de las llaves pública y privada con el algoritmo RSA

root@sahisweb:/# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
c6:c7:87:a8:df:fc:28:a1:a9:fe:d7:d7:ce:8d:a9:72 root@sahisweb
The key's randomart image is:
+--[RSA 2048]----+

Una vez generadas las llaves se procede a copiar la llave pública (id_rsa.pub) para el servidor remoto en un fichero llamado authorized_keys dentro del directorio .ssh del home del usuario rsync_central.

3. Copiar la llave pública del usuario que se utilice para la sincronización hacia el servidor remoto

root@sahisweb:/# ssh-copy-id -i /root/.ssh/id_rsa.pub rsync_central@10.1.1.100

rsync_central@10.1.1.100's password:

Now try logging into the machine, with "ssh rsync_central@10.1.1.100", and check in:

.ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.

root@sahisweb:/#

Es bueno destacar que en el proceso se debe especificar la localización de la llave pública, el usuario con que se accederá al servidor remoto y la dirección IP de dicho servidor.

A partir de este paso es posible la autenticación y acceso al servidor remoto sin utilizar contraseña explícitamente, como se muestra en el paso cuatro.

4. Comprobar que el usuario y la llave copiada funcionan correctamente

root@sahisweb:/# ssh rsync_central@10.1.1.100

Linux dns 2.6.32-5-686 #1 SMP Tue Mar 8 21:36:00 UTC 2011 i686

The programs included with the Debian GNU/Linux system are free software;

the exact distribution terms for each program are described in the

individual files in /usr/share/doc//copyright.*

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent

permitted by applicable law.

rsync_central@archivenhis:~#

5. Realizar las copias de los ficheros utilizando rsync y ssh

root@sahisweb:/# rsync -avz --delete-after -e ssh

rsync_central@10.1.1.100:/var/www/archivenhis/repositorio/

/var/www/sahisweb/sites/default/files/repositorio/nuevoarchivo/

6. Añadir la sincronización de los directorios del nuevo Archivo integrado al sistema en las tareas programadas

root@sahisweb:/# crontab –e

Insertar

*/5 * * * * /usr/bin/rsync -avz --delete-after -e ssh rsync_central@10.1.1.100:/var/www/archivenhis/repositorio/ /var/www/sahisweb/sites/default/files/repositorio/nuevoarchivo/ >> /var/log/sincronizacion.log

Una vez que se ha configurado y comprobado el correcto funcionamiento de la replicación, tanto de la base de datos como de las representaciones digitales de los

documentos, desde el Archivo a integrar hacia el centro de datos donde se ha desplegado SAHISWEB, sólo resta indicarle a este último cómo acceder a los datos replicados.

Para ello se deben seguir los pasos especificados en el CUS Gestionar Registro de Archivo, en su sección Registrar datos de acceso, el cual ha sido implementado en SAHISWEB.

En la figura 2.3 se muestra un ejemplo en correspondencia con los datos de configuración utilizados en el epígrafe.

Figura 2.3 Captura de interfaz de usuario de SAHISWEB que muestra cómo proporcionar los datos de acceso a un Archivo integrado.

2.2.3 Modelo de despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo (8).

En la figura 2.4 se muestra el diagrama de despliegue de SAHISWEB, donde las áreas denotadas por Archivo 1, Archivo 2 y Archivo 3 representan Archivos Históricos que describen el patrimonio documental bajo su custodia empleando ArchiVenHIS y que pueden ser integrados a SAHISWEB. Para el despliegue de ArchiVenHIS se precisa en cada Archivo de un servidor de bases de datos MySQL y un servidor Web Apache. El primero alojará la base de datos del sistema y el segundo la aplicación Web que garantiza el mantenimiento y acceso a la información desde la red local del propio Archivo.

Por otro lado, el área denotada por AGN representa el nodo central que incluye los servidores Web y de bases de datos MySQL para hospedar la aplicación SAHISWEB, su base de datos, así como cada una de las réplicas de las bases de datos de las instancias de ArchiVenHIS de cada uno de los Archivos integrados. Lo anterior permite el acceso desde Internet, a través de SAHISWEB, al patrimonio documental disperso en cada uno de los Archivos integrados, una vez centralizados a partir de los mecanismos reflejados en la figura 2.4 y configurados según se ha descrito en secciones anteriores.

Es conveniente resaltar que no es necesario que cada uno de los Archivos integrados sea accesible desde Internet para difundir el patrimonio documental bajo su custodia y que conservan su autonomía aún si fallara el enlace entre alguno de ellos y el nodo central. Además, mientras dure la falla del enlace es posible acceder a la información previamente replicada en el nodo central. Este último puede actualizarse de los cambios sucedidos durante el tiempo que dure la falla, una vez que se restablezca el canal de comunicación.

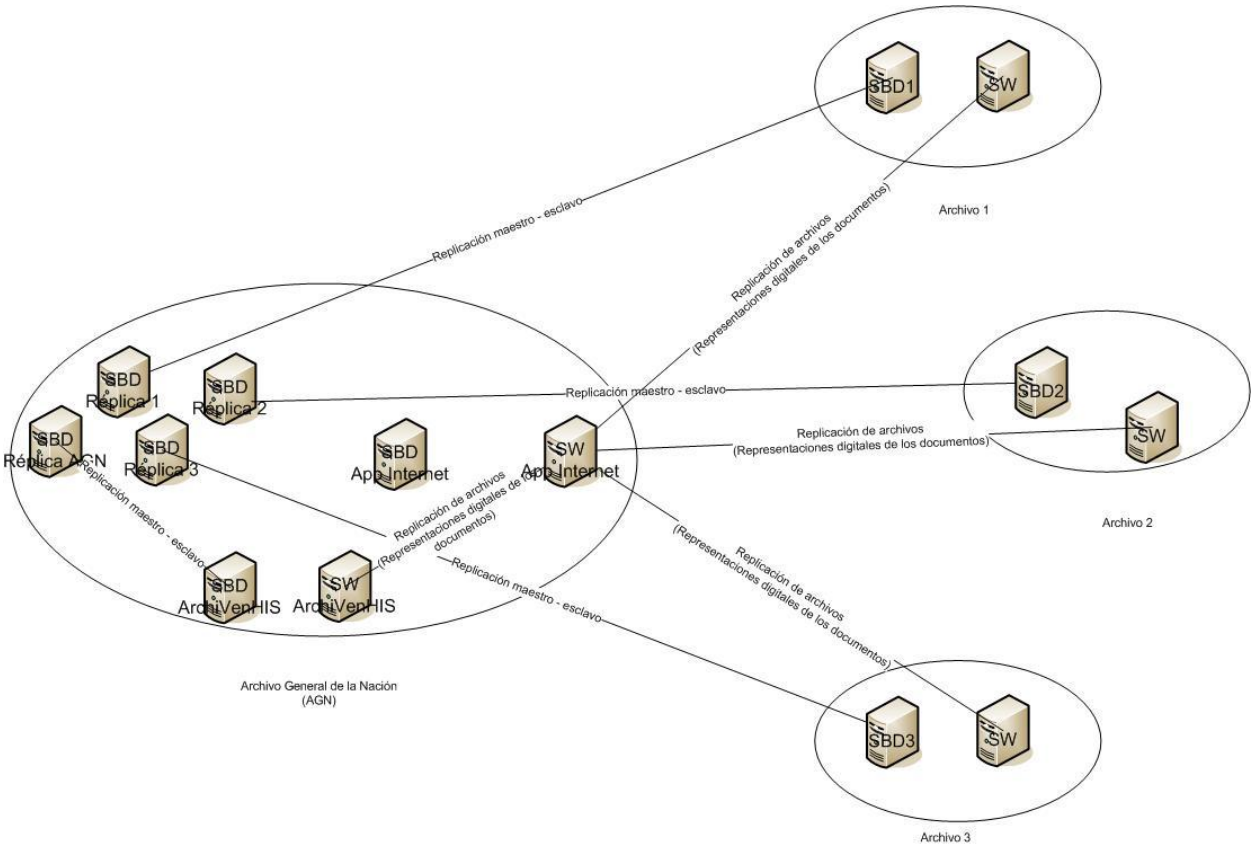

Figura 2.4 Diagrama de despliegue del sistema SAHISWEB.

2.3 Conclusiones parciales

A partir de la configuración de los mecanismos de replicación de MySQL y rsync se garantiza la integración de los datos relativos a los fondos documentales descritos a través de ArchiVenHIS y que se encuentren dispersos en varios Archivos Históricos.

La herramienta phpBB satisface todos los requerimientos funcionales correspondientes a la creación de un espacio de intercambio colaborativo entre investigadores.

La selección del CMS Drupal y phpBB como herramienta para la gestión de foros agiliza el desarrollo de SAHISWEB y contribuye con la soberanía tecnológica de quien lo utilice.

SAHISWEB ofrece al administrador los mecanismos necesarios para indicarle al propio sistema cómo acceder a los datos centralizados desde los Archivos integrados. Una vez hecho esto permite a los usuarios el acceso a los mismos, facilitando la difusión del patrimonio documental que estos custodian.

Capítulo 3: Valoración de la propuesta

En el presente capítulo se describen las pruebas realizadas para verificar la eficiencia del método propuesto para la integración de los datos, así como el análisis de sus resultados. Además se expone un resumen de las pruebas ejecutadas por el Centro Nacional de Calidad del Software (CALISOFT) para certificar la liberación del producto SAHISWEB, que se sirve de los datos centralizados para prestar servicios de localización y consulta de los documentos custodiados por los archivos integrados a la solución y descritos mediante ArchiVenHIS.

3.1 Pruebas de funcionalidad

La Prueba de software se puede definir como una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas (configuración de la prueba), registrándose los resultados obtenidos. Seguidamente se realiza un proceso de Evaluación en el que los resultados obtenidos se comparan con los resultados esperados para localizar fallos en el software. Estos fallos conducen a un proceso de Depuración en el que es necesario identificar la falta asociada con cada fallo y corregirla, pudiendo dar lugar a una nueva prueba (9).

Según Pressman (2001), las pruebas constituyen un elemento crítico para la garantía de la calidad del software y se clasifican en dos tipos principales, las de caja blanca (que son realizadas sobre el código de la aplicación) y las de caja negra (que son realizadas sobre la interfaz del software con el objetivo fundamental de descubrir si la entrada de datos es validada, así como si los resultados obtenidos son los esperados).

En el caso del sistema SAHISWEB se realizaron pruebas de caja negra basadas en el método partición de equivalencia. La partición de equivalencia divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar (9).

Las pruebas se realizaron a partir de la estrategia de pruebas diseñada la cual consta de tres pasos: diseño, implementación del método y análisis de los resultados.

Diseño: El método partición de equivalencia consta de tres pasos, teniendo como artefacto de entrada los CUS:

- Para cada caso de uso, generar un sistema completo de escenarios.
- Identificar los casos de prueba.
- Identificar los valores de datos para las pruebas.

Implementación: Una vez diseñados los casos de prueba se procede a probar contra la aplicación y verificar si se cumplieron los requisitos establecidos en los casos de usos. En este caso el nivel de las pruebas a realizar por parte del equipo de CALISOFT es de liberación.

Las pruebas de liberación se encargan de la verificación, es decir, permiten determinar que los requisitos estén completos y correctos. Para realizarlas es preciso definir y montar el escenario (entorno) de pruebas teniendo en cuenta:

- Despliegue del sistema
- Recursos del sistema (Servidores, PC Cliente)

Análisis de los resultados: Se realizaron tres iteraciones por parte de CALISOFT quedando liberado el sistema en la tercera iteración. En la figura 3.1 se muestra una gráfica que ilustra la cantidad de no conformidades detectadas en cada una de las iteraciones.

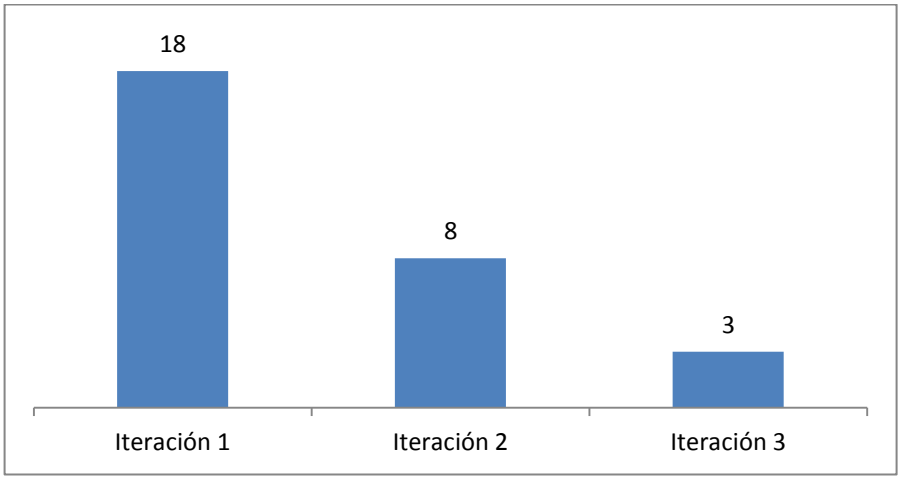

Figura 3.1 Cantidad de no conformidades detectadas en las iteraciones realizadas durante el proceso de prueba.

Es válido aclarar que en la gráfica sólo se tienen en cuenta aquellas no conformidades que tuvieron incidencia en la aplicación, aunque se detectaron otras que por su naturaleza no procedieron, es decir, no fue necesaria una respuesta por parte del equipo de desarrollo.

Las expuestas en la figura 3.1 tuvieron un impacto de nivel medio a partir de la detección de errores en el formato de varias interfaces, así como en las dimensiones de algunos componentes, principalmente de los mensajes al usuario. Además se encontró un error ortográfico y varios mensajes que no se mostraban en el momento preciso. Para resolver las no conformidades la medida tomada por la dirección del proyecto en todos los casos fue regresar al flujo de implementación.

De manera general, los resultados obtenidos con las pruebas realizadas fueron satisfactorios y se pudo constatar el correcto funcionamiento de las funcionalidades implementadas.

Además de las pruebas funcionales, CALISOFT realizó pruebas de liberación sobre los manuales de usuario del sistema y el manual de instalación, los cuales también fueron liberados en la tercera iteración. No se considera necesario establecer un análisis de estas no conformidades pues todas giraron en torno a errores ortográficos o algunas incongruencias entre las interfaces del sistema incluidas en los manuales y las reales implementadas.

3.2 Replicación como alternativa para la centralización de los datos

Los entornos de réplica empleados en las pruebas se han inspirado en lo propuesto en (10) para medir la rapidez de la replicación en MySQL.

Para estudiar el comportamiento de la replicación se observa el tiempo requerido para replicar cada tupla de una determinada relación R. Cada registro de R será estampado, al momento de su inserción, con una marca de tiempo generada mediante una llamada a una UDF no determinista (10). Se tomará el tiempo requerido para replicar cada registro como la diferencia entre sus marcas de tiempo en el esclavo y el maestro.

La estructura de la relación R se especifica en el siguiente fragmento de código:

```
CREATE TABLE r(
id SERIAL PRIMARY KEY,
marca_tiempo VARCHAR(100) NOT NULL,
fecha TIMESTAMP NOT NULL
)ENGINE=INNODB;
```

Se implementa un procedimiento almacenado que puebla R en el maestro con una determinada cantidad de tuplas especificada como parámetro.

```
CREATE PROCEDURE poblar(cantidad INTEGER)
BEGIN
 DECLARE i INTEGER DEFAULT 1;
 WHILE i<=cantidad DO
 INSERT INTO r(marca_tiempo) VALUES(now_usec());
 SET i=i+1;
 END WHILE;
END; $$
```

Una primera prueba se realiza a partir de configurar ambas instancias de MySQL, maestro y esclavo, en un mismo servidor. Una segunda prueba se hace a partir de la configuración del maestro y el esclavo en máquinas independientes enlazadas mediante una red a 100Mbps y sincronizando sus relojes mediante el protocolo NTP[14].
En ambos casos, una vez concluido el proceso de replicación para registrar las diferencias de tiempo de cada tupla, se emplea una tabla federada para obtener los datos del maestro (federated_r) y otra para guardar los resultados para su posterior análisis (diferencias).

[14] NTP: *Network Time Protocol.*

```
CREATE TABLE federated_r(
id SERIAL PRIMARY KEY,
marca_tiempo VARCHAR(100) NOT NULL,
fecha TIMESTAMP NOT NULL)
ENGINE=FEDERATED
CONNECTION='mysql://usuario:clave@masterIP/db/r';
```

Donde los comodines representan:
masterIP: Nombre o dirección IP del servidor maestro.
usuario: Nombre dado a un usuario MySQL, previamente definido en el maestro con privilegios suficientes para el acceso desde el servidor esclavo y la consulta de la relación R.
clave: Contraseña asignada a usuario.
db: Nombre de la base de datos replicada que contiene a la relación R.

```
CREATE TABLE diferencias(
id INTEGER PRIMARY KEY,
diferencia INTEGER NOT NULL
)ENGINE=INNODB;
```

Esta última se puebla a partir de los datos generados por la siguiente consulta:

```
INSERT INTO diferencias(id, diferencia) SELECT r.id, TIMESTAMPDIFF(FRAC_SECOND, federated_r.marca_tiempo, r.marca_tiempo) AS diferencia FROM federated_r JOIN r USING(id);
```

Análisis de los resultados de la prueba de replicación con ambas instancias, maestro y esclavo, sobre un mismo servidor

El análisis de los resultados comienza a partir de la interpretación de algunos estadígrafos descriptivos (de tendencia central, posición y dispersión) resumidos en la tabla 3.1.

Tabla 3.1 Estadísticos descriptivos de la variable que representa el tiempo utilizado para replicar cada tupla.

N	Válidos	100077
	Perdidos	0
Media		1138.45
Mediana		903.00
Desv. típ.		2945.177
Asimetría		28.664
Error típ. de asimetría		.008
Curtosis		1267.142
Error típ. de curtosis		.015
Percentiles	10	798.00
	20	837.00
	25	852.00
	30	863.00
	40	884.00
	50	903.00
	60	923.00
	70	949.00
	75	965.00
	80	985.00
	90	1076.00

A partir de estos se evidencia que el 90% de los registros se replica en un tiempo inferior a los 1076 microsegundos y el 50% requiere entre 852 y 949 microsegundos para replicarse. Un registro tarda como promedio 1138 microsegundos en ser replicado, aunque este valor de conjunto con el análisis anterior y una desviación típica de aproximadamente 2945 microsegundos indican la presencia de algunos valores extremos.

Se aprecia además que la distribución no es simétrica respecto a la media, nótese que más del 90% de las observaciones está por debajo de esta. Esto además se corrobora a partir del cálculo de la asimetría, cuyo valor positivo refleja el hecho de que las mayores frecuencias tienen lugar a la izquierda de la media. Por otra parte el valor elevado de la curtosis refleja la forma leptocúrtica de la distribución. Lo anterior puede ser apreciado gráficamente en el histograma de la figura 3.2:

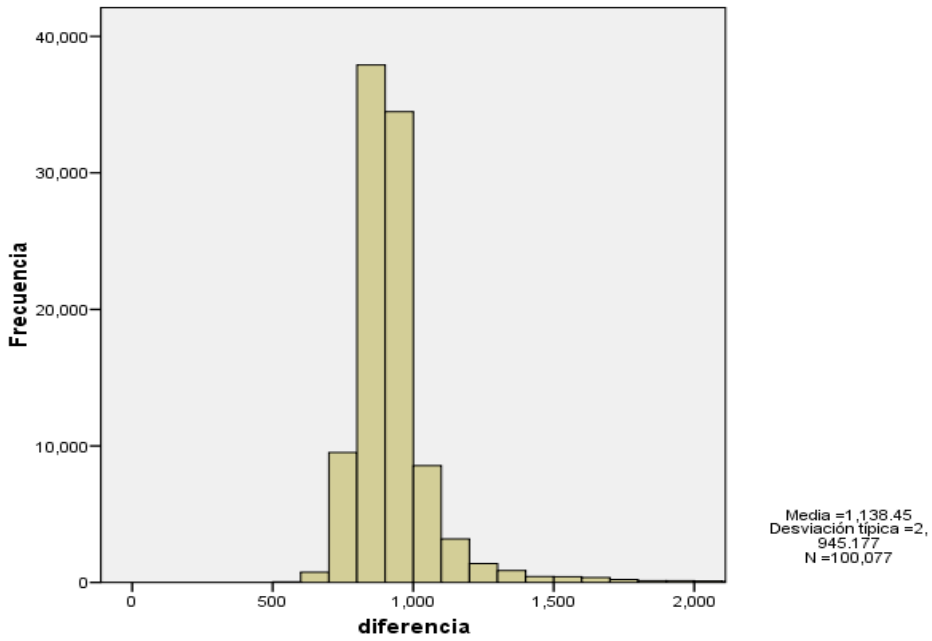

Figura 3.2 Histograma de frecuencias de la variable que representa el tiempo empleado para replicar cada tupla.

Como parte del análisis se observa si existe relación entre el orden de los registros a replicar y el tiempo necesario para replicarlos. Una primera aproximación para cumplir este objetivo es inspeccionar visualmente un gráfico de dispersión del tiempo respecto al número de los registros. La figura 3.3 muestra como la nube de puntos se concentra alrededor de 1ms independientemente del orden de los registros, con excepción de algunos valores atípicos, por tanto no indica la existencia de alguna relación entre las variables.

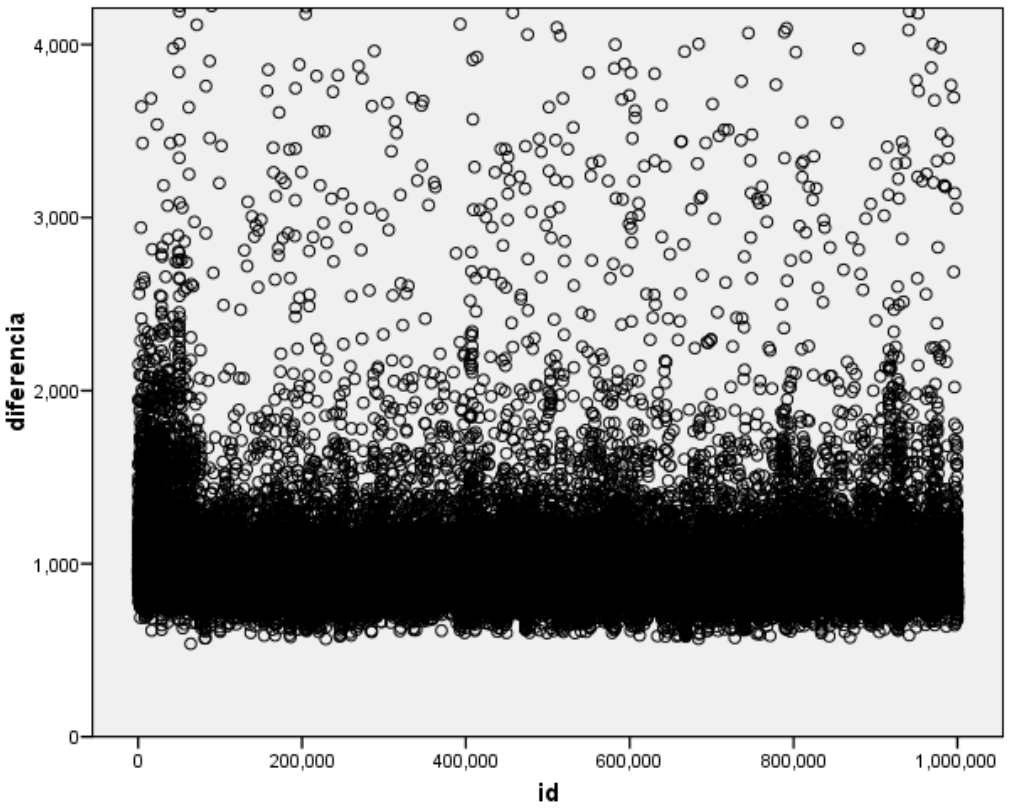

Figura 3.3 Diagrama de dispersión del tiempo requerido para replicar cada tupla en función del orden de esta.

Finalmente el estadígrafo Tau-b de Kendall muestra que no hay evidencia de una correlación entre los rangos de ambas variables (véase la tabla 3.2).

Tabla 3.2 Resumen del estadígrafo Tau-b de Kendal aplicado a las variables que representa el tiempo de replicación de cada tupla y su orden.

			id	diferencia
Tau_b de Kendall	id	Coeficiente de correlación	1.000	-.040(**)
		Sig. (bilateral)	.	.000
		N	100077	100077
	diferencia	Coeficiente de correlación	-.040(**)	1.000
		Sig. (bilateral)	.000	.
		N	100077	100077

** La correlación es significativa al nivel 0,01 (bilateral).

Análisis de los resultados de la prueba de replicación a través de una LAN con una velocidad de 100 Mbps

Durante el desarrollo de la prueba se monitoriza la diferencia entre los relojes de ambos servidores, que se sincronizan mediante el protocolo NTP. Primeramente se analiza el comportamiento de esta variable con el propósito de evaluar su influencia en los resultados de la prueba.

Tabla 3.3 Estadísticos descriptivos de la variable que representa las diferencias entre los relojes de los servidores maestro y esclavo.

N	Válidos	1468
	Perdidos	0
Media		.00000831
Mediana		.00000950
Desv. típ.		.000418094
Asimetría		-.326
Error típ. de asimetría		.064
Curtosis		70.190
Error típ. de curtosis		.128
Percentiles	25	-.00006175
	50	.00000950
	75	.00008000

Los estadígrafos resumidos en la tabla 3.3 reflejan que el sincronismo entre los relojes de los servidores, maestro y esclavo, no impacta significativamente en los resultados de la prueba. Nótese que la media tiende a cero (aproximadamente 8 microsegundos) y su cercanía (aproximadamente 1 microsegundo) con la mediana (aproximadamente 9 microsegundos). Además el 50% de las diferencias muestreadas durante la prueba están entre -62 y 80 microsegundos. La simetría y la curtosis muestran un grado aceptable de simetría y un alto valor de las frecuencias de valores próximos a cero. Esto puede apreciarse gráficamente en el histograma de la figura 3.4.

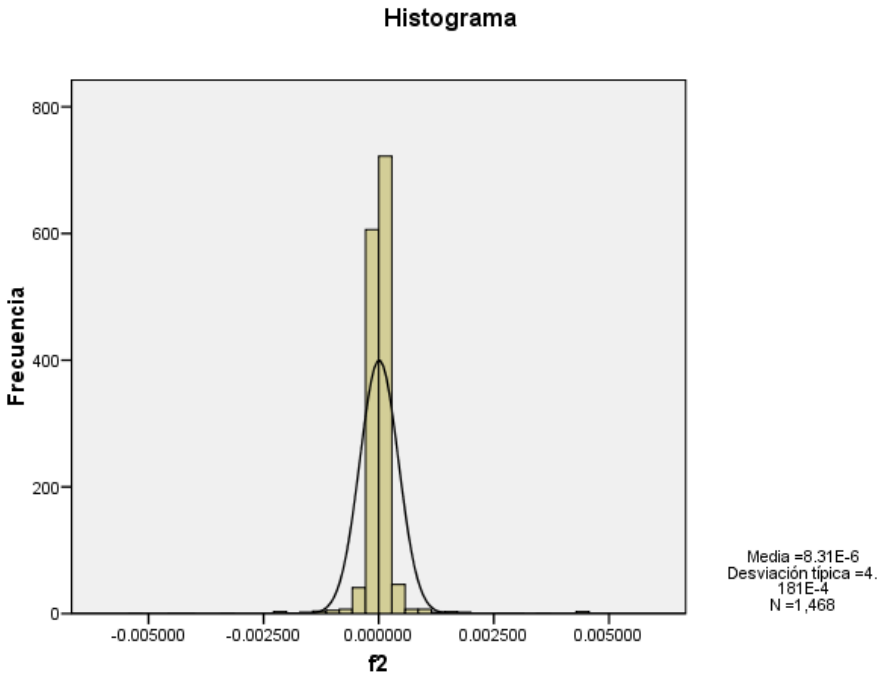

Figura 3.4 Histograma de frecuencias de la variable que representa las diferencias entre los relojes de los servidores maestro y esclavo.

Se realiza entonces el análisis del comportamiento de la replicación durante la prueba sin tener en cuenta la influencia del sincronismo entre los relojes de los servidores involucrados en las variables registradas durante su desarrollo.

Tabla 3.4 Estadísticos de la variable que representa el tiempo de replicación de cada tupla,

N	Válidos	100077
	Perdidos	0
Media		4221.95
Mediana		1907.00
Desv. típ.		22106.099
Asimetría		21.348
Error típ. de asimetría		.008
Curtosis		597.782
Error típ. de curtosis		.015
Percentiles	10	1536.00
	20	1653.00
	25	1698.00
	30	1740.00

40	1821.00
50	1907.00
60	2005.00
70	2138.00
75	2240.00
80	2409.00
90	3337.00

Los estadígrafos descriptivos de posición, dispersión y forma resumidos en la tabla 3.4 reflejan una distribución con características similares a las observadas al evaluar los resultados de la prueba realizada con ambas instancias de MySQL, maestro y esclavo, en un mismo servidor. El 90% de los registros replicados en un tiempo inferior a la media, está última distante de la moda y un alto valor de la asimetría positiva reflejan la ocurrencia de las frecuencias más altas a la izquierda de la media, además un alto valor de curtosis indicando la forma leptocúrtica de la distribución. Esto puede apreciarse gráficamente en el histograma de la figura 3.5:

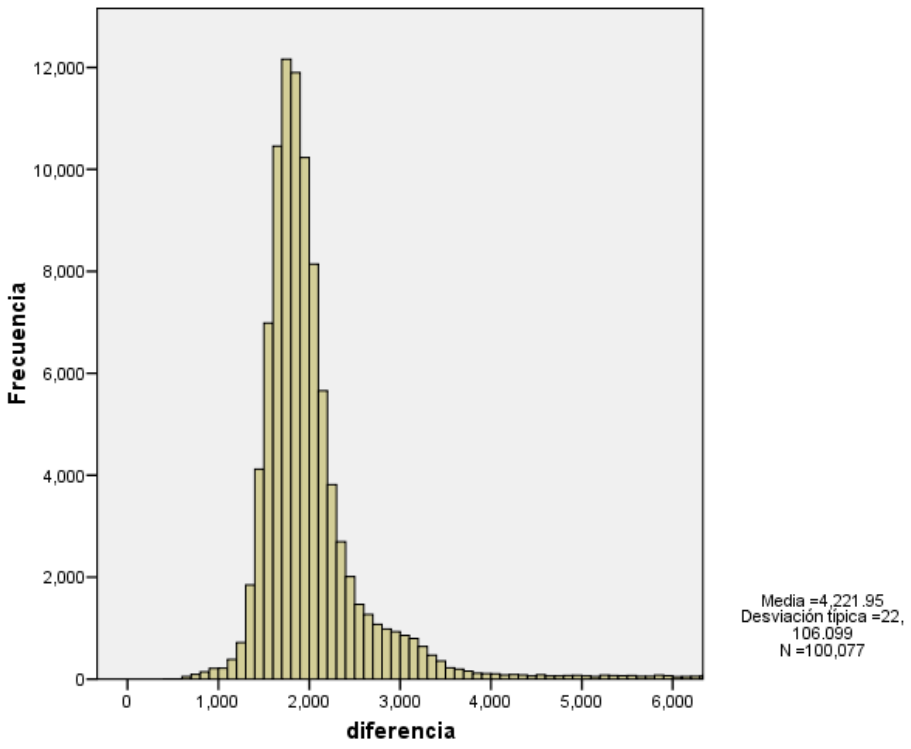

Figura 3.5 Histograma de frecuencias de la variable que representa el tiempo invertido en la replicación de cada tupla.

De forma análoga se inspecciona el gráfico de dispersión (véase la figura 3.6), observándose en este caso como la nube de puntos se concentra alrededor de los 2ms aproximadamente y no sugiere la existencia de una correlación entre el orden del registro replicado y el tiempo necesario para replicarlo.

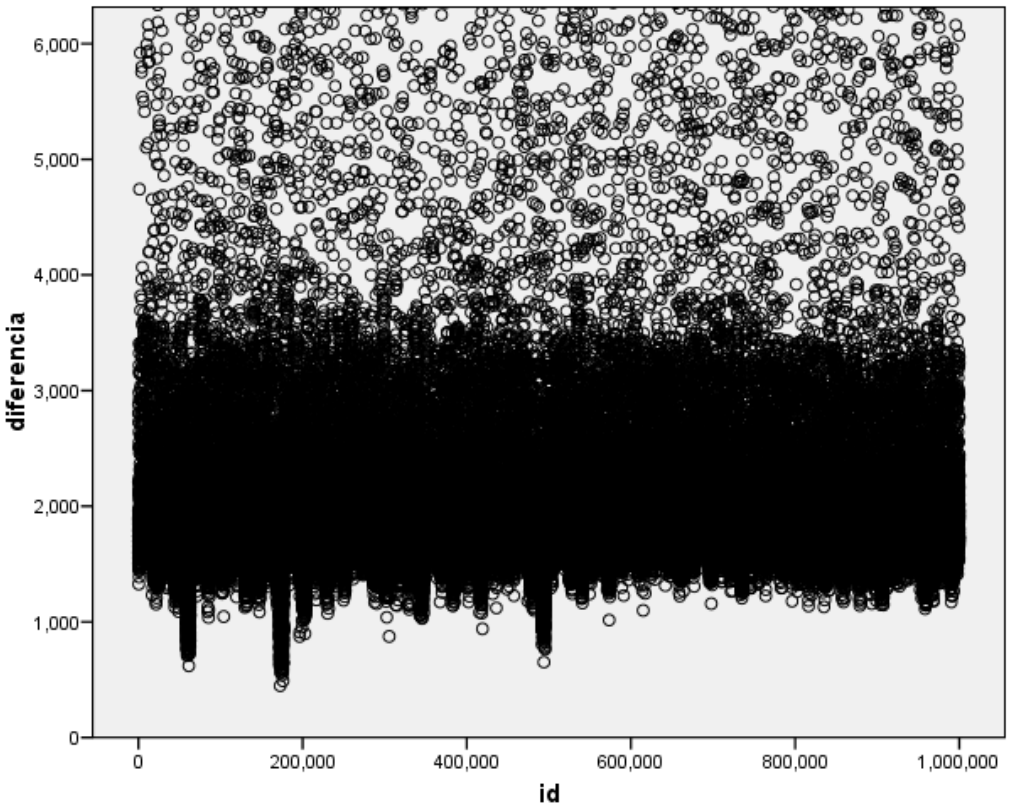

Figura 3.6 Diagrama de dispersión del tiempo requerido para replicar cada tupla en función del orden de esta.

Se emplea una vez más la Tau-b de Kendall para corroborarlo (véase la tabla 3.5).

Tabla 3.5 Resumen del estadígrafo Tau-b de Kendall aplicado a las variables que representan el tiempo de replicación de cada tupla y su orden.

			id	diferencia
Tau_b de Kendall	id	Coeficiente de correlación	1.000	-.035(**)
		Sig. (bilateral)	.	.000
		N	100077	100077

diferencia	Coeficiente de correlación	-.035(**)	1.000
	Sig. (bilateral)	.000	.
	N	100077	100077

** La correlación es significativa al nivel 0,01 (bilateral).

3.3 OAI-PMH como alternativa para la recolección de los datos

Estudio del comportamiento de la recolección de sub-listas sucesivas para completar una solicitud de tipo lista del protocolo OAI-PMH

Para observar el posible comportamiento del protocolo OAI-PMH se realiza una prueba que consiste en recuperar sucesivamente K registros de una determinada relación R. Sea R:

```
CREATE TABLE r(
 id INTEGER AUTO_INCREMENT PRIMARY KEY,
 f1 VARCHAR(100) NOT NULL,
 fecha TIMESTAMP NOT NULL
)ENGINE=INNODB;
```

Para el desarrollo de la misma se empleó la herramienta JMeter. Desarrollada completamente en java y diseñada para implementar, cargar y ejecutar pruebas funcionales y medir el rendimiento de una variada gama de servidores (Web, Bases de datos vía JDBC[15], Correo electrónico: SMTP(S), POP3(S) e IMAP(S), etc.).

El plan de prueba elaborado empleando la herramienta JMeter consta de cinco componentes: Thread Group, JDBC Connection Configuration, Counter, JDBC Request y Graph Results. A continuación se presenta la configuración de cada uno de ellos mediante una breve descripción.

El componente de tipo Thread Group es el punto de inicio del plan y permite indicarle a JMeter el número de usuarios que se desea simular, con qué frecuencia envían solicitudes y cuantas deben enviar. En este caso se ha indicado sólo un usuario y se

[15] JDBC: Java Databases Connection.

especifica que debe ejecutar las solicitudes 10000 veces, dado que para la prueba se ha tomado k=100 y la cardinalidad de R es 10^6 (véase la figura 6.1 del anexo 6 para la configuración del componente Thread Group en la herramienta JMeter).

El componente JDBC Connection Configuration permite especificar los parámetros de conexión a una base de datos a partir de los cuales se crea una conexión JDBC que será empleada por el componente JDBC Request (véase la figura 6.2 del anexo 6 para la configuración del componente JDBC Connection Configuration en la herramienta JMeter).

El componente Counter permite definir un contador que puede ser referenciado por otros componentes dentro de Thread Group. En este caso se emplea este componente para definir el punto de inicio a partir del cual se recuperan los siguientes K registros en cada solicitud (véase la figura 6.3 del anexo 6 para la configuración del componente Counter en la herramienta JMeter). El componente JDBC Request permite enviar una consulta a la base de datos mediante una solicitud JDBC (véase la figura 6.4 del anexo 6 para la configuración del componente Request en la herramienta JMeter).

El componente Graph Results permite apreciar gráficamente el comportamiento del tiempo consumido por las solicitudes. Otros datos de interés se muestran en la parte inferior del gráfico: la cantidad total de solicitudes, el tiempo empleado en resolver la solicitud actual, el promedio de tiempo de las solicitudes, así como la desviación estándar en milisegundos y la cantidad de solicitudes por minuto atendidas por el servidor. Es posible además indicar un fichero de salida donde almacenar determinados elementos de interés para un análisis posterior como el tiempo, latencia, cantidad de bytes de la respuesta, entre otros datos relativos a la solicitud. Algunos de estos datos precisan que el formato del fichero de salida sea XML, como en este caso (véase la figura 6.5 del anexo 6 para la configuración del componente Graph Results en la herramienta JMeter).

Como se ha explicado los resultados arrojados por la herramienta se encuentran disponibles en un archivo XML, por lo que se decide transformar su formato para su posterior análisis con SPSS.

Para esto se utiliza la herramienta PDI (Pentaho Data Integration). Desarrollada en Java, es ampliamente usada por las posibilidades que ofrece para los procesos ETL

(Extraction, Transformation and Loading), así como la migración de datos entre bases de datos y aplicaciones, entre otros usos; a través de un ambiente de diseño con una interfaz gráfica intuitiva. La figura 3.7 representa el proceso diseñado empleando la herramienta PDI para transformar los datos en formato XML, obtenidos como resultado de la ejecución de la prueba antes descrita, a una tabla en una base de datos relacional.

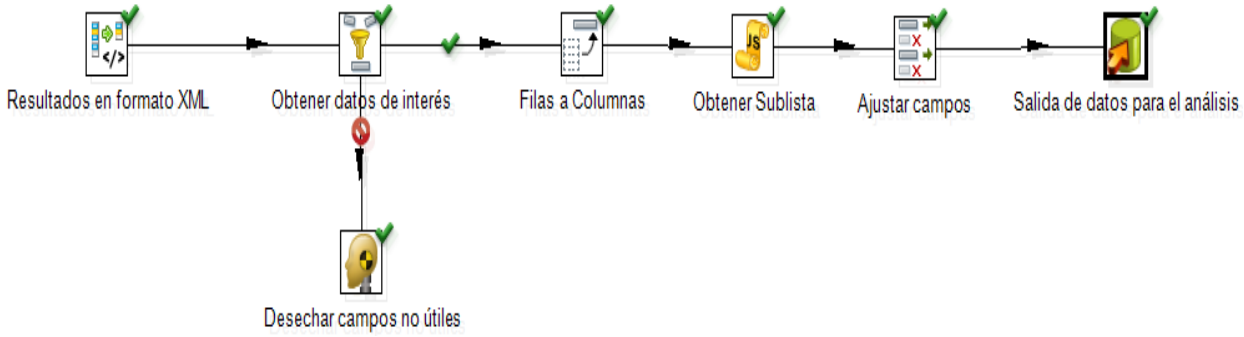

Figura 3.7 Transformación desarrollada empleando PDI para disponer de los datos objeto de análisis en un formato accesible desde SPSS.

El análisis de los resultados se inicia a partir de varios estadígrafos descriptivos tales como media, mediana, desviación estándar, percentiles, asimetría y curtosis.

Tabla 3.6 Estadísticos descriptivos de la variable que representa el tiempo invertido para recuperar cada sublista.

N	Válidos	10000
	Perdidos	0
Media		4926.40
Mediana		5092.00
Desv. típ.		1150.787
Asimetría		-1.091
Error típ. de asimetría		.024
Curtosis		1.811
Error típ. de curtosis		.049
Percentiles	10	3450.20
	20	4241.20
	25	4457.00
	30	4607.00
	40	4878.00
	50	5092.00

60	5298.60
70	5537.00
75	5668.00
80	5815.00
90	6167.00

A partir de lo anterior se infiere que, para k=100, el 50% de las sub-listas requieren entre 4457 y 5668 milisegundos para su recuperación. Para recuperar una sub-lista se necesita un tiempo medio de aproximadamente 4926 milisegundos con una desviación de 1150 milisegundos.

El valor de la asimetría refleja un leve corrimiento de las mayores frecuencias a la derecha de la media. Lo anterior puede ser apreciado de manera gráfica en el histograma de la figura 3.8.

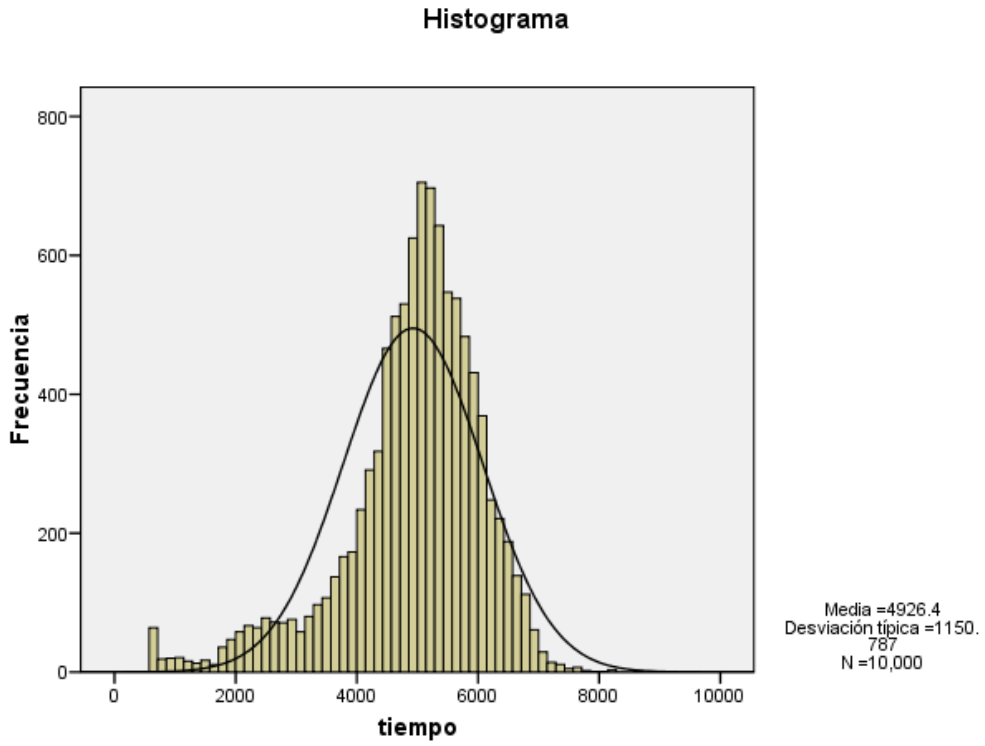

Figura 3.8 Histograma de frecuencias de la variable que representa el tiempo de recuperación de cada sub-lista.

Como parte del análisis se observa si existe relación entre el orden de las sub-listas y el tiempo necesario para replicarlas. Una primera aproximación para cumplir este objetivo es inspeccionar visualmente un gráfico de dispersión del tiempo respecto al número de orden de la sub-lista. La nube de puntos de la figura 3.9 sugiere un aumento del tiempo necesario para recuperar una sub-lista en la medida que aumenta su número de orden.

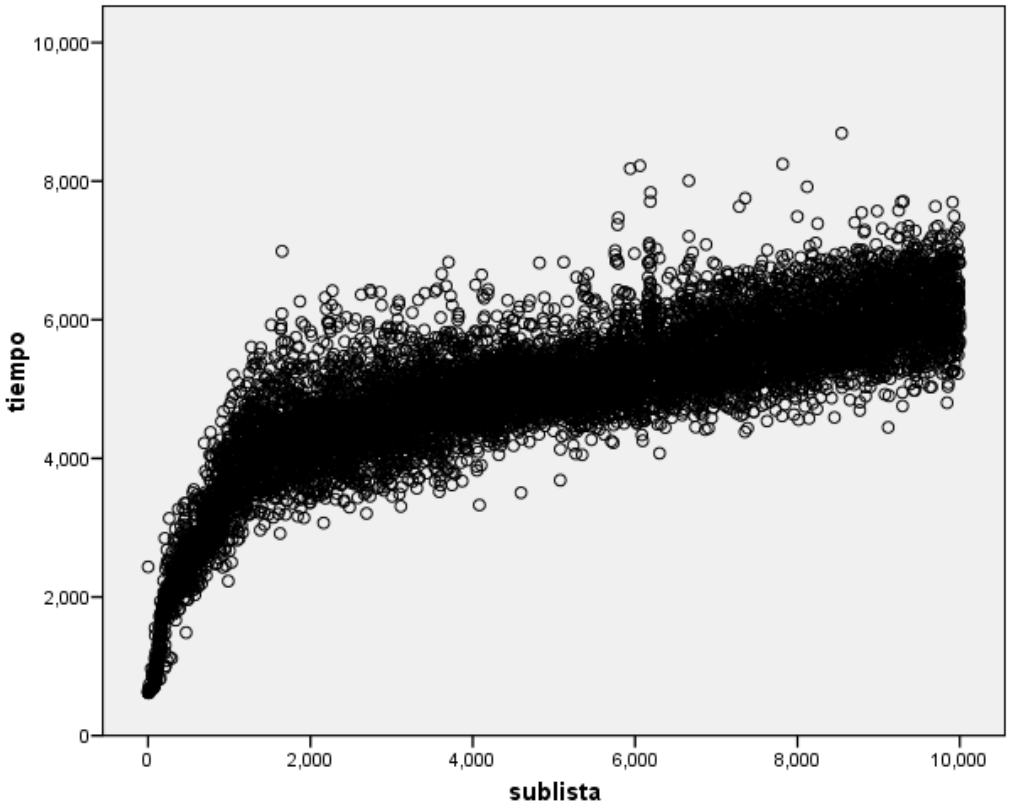

Figura 3.9 Diagrama de dispersión de la variable que representa el tiempo de recuperación de cada sub-lista en función de su orden.

Se realiza entonces un análisis de regresión. El gráfico de la figura 3.10 muestra cómo se ajustan las curvas logarítmica y de potencia. En la tabla 3.7 los valores de R^2 muestran una correlación fuerte entre las variables analizadas, confirmándose la dependencia del tiempo necesario para recuperar una sub-lista respecto a su orden.

Tabla 3.7 Resumen del modelo regresión y estimaciones de los parámetros.

Variable dependiente: tiempo

Ecuación	Resumen del modelo					Estimaciones de los parámetros	
	R cuadrado	F	gl1	gl2	Sig.	Constante	b1
Logarítmica	.807	41875.867	1	9998	.000	-3584.316	1036.516
Potencia	.842	53258.871	1	9998	.000	398.815	.301

La variable independiente es: sublista.

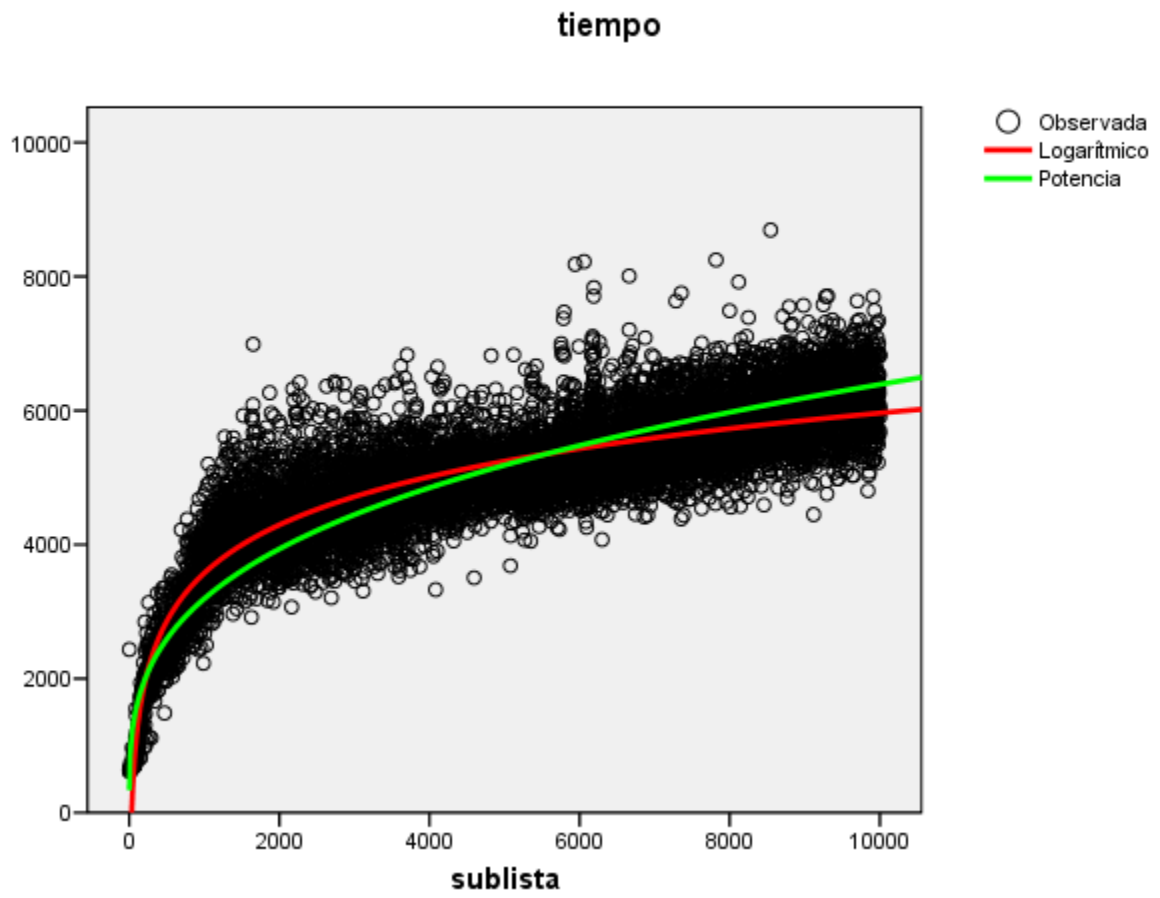

Figura 3.10 Curvas obtenidas mediante modelos de regresión.

3.4 Conclusiones parciales

Las pruebas realizadas reflejan que el tiempo para recuperar cada registro mediante la replicación en MySQL está en el orden de un milisegundo y esta velocidad está limitada por la velocidad de la red que conecta al servidor esclavo con el maestro.

A lo descrito en el análisis de los resultados de las pruebas puede sumarse el hecho de que la duración de las mismas fue de unas decenas de minutos (entre 30 y 40 minutos). Además este tiempo se reduce a 15 o 20 minutos si la replicación tiene lugar en un período de poca actividad en el servidor maestro.

Por otra parte, en la recopilación sucesiva de sub-listas para completar solicitudes OAI-PMH de los tipos ListIdentifier, ListRecords, ListSets, el tiempo empleado para la recuperación de cada sub-lista aumenta con el orden de la misma, debido al incremento del volumen de registros que es necesario consultar y ordenar. La duración de estas pruebas estuvo entre 13 y 14 horas aun cuando estas sólo responden a la parte del proceso de recolección en que el proveedor de datos genera las sub-listas necesarias y no incluyen el tiempo de transferirlas al recolector y procesarlas en este último para que los metadatos contenidos en ellas estén disponibles por el proveedor de servicios.

Conclusiones

Los objetivos trazados para la presente investigación fueron cumplidos satisfactoriamente y se arribó a las siguientes conclusiones:

- Aunque tanto OAI-PMH como la replicación MySQL y rsync son alternativas viables de recolección de datos para la implementación de la integración de instancias del sistema de gestión de documentos históricos ArchiVenHIS, la replicación de los datos garantizó el acceso total a las descripciones de los fondos documentales custodiados por los Archivos integrados, así como un uso más eficiente del canal de comunicación que se establece con cada Archivo integrado a la solución.
- Con la identificación y configuración de los mecanismos de replicación de MySQL y rsync se realizó de forma eficiente la integración de los datos de los fondos documentales dispersos en Archivos Históricos que emplean el sistema ArchiVenHIS para describirlos.
- La aplicación Web SAHISWEB facilitó la difusión a través de un punto de acceso único del patrimonio documental que custodian los Archivos Históricos que fueron integrados a partir del mecanismo de replicación.
- Con la integración del CMS Drupal y phpBB como herramienta de gestión de foros se logró implementar un espacio para el intercambio entre los investigadores, consumiendo un menor tiempo de desarrollo.
- Las pruebas de liberación realizadas por CALISOFT verificaron la aplicabilidad del software al comprobar que los CUS fueron implementados satisfactoriamente.

Recomendaciones

Se recomienda:

- Registrar en el CENDA el sistema SAHISWEB.
- Implementar el protocolo OAI-PMH como variante para la recolección de datos, en aras de integrar Archivos Históricos que no utilicen ArchiVenHIS para describir el patrimonio documental que resguardan.
- Generalizar el sistema SAHISWEB para su despliegue en Cuba integrando aquellas instituciones que manejen Archivos Históricos y que utilicen personalizaciones de ArchiVenHIS para describir los fondos documentales que custodian.

Referencias Bibliográficas y Bibliografía

1. **Rodríguez, Reynier Pernía y otros.** "Sistema Gestor de Documentos Históricos". *Memorias del XII Congreso Internacional de Información, Info 2012. Foro Instituciones de Información del Siglo XXI.* La Habana : s.n., 2012.

2. Archivo General de la Nación "Francisco de Miranda". [En línea] [Citado el: 13 de 11 de 2012.] http://www.agn.gob.ve/.

3. **Cid Almaguer, Adrián, Pernía Rodríguez, Reynier y Surós Vicente, Annia.** "Solución para Sistema Nacional de Archivos Históricos basada en Software Libre". *Memorias de la XIII Convención y Feria Internacional, INFORMÁTICA. IV Taller Internacional de Software Libre y estándares abiertos.* La Habana : s.n., 2009.

4. **Consejo Internacional de Archivos.** *ISAD (G): Norma Internacional General para la Descripción Archivística*. Madrid : s.n., 2000.

5. **OAI.** Open Archives Initiative - Protocol for Metadata Harvesting - v.2.0. [En línea] 2008. [Citado el: 07 de 03 de 2013.] http://www.openarchives.org/OAI/openarchivesprotocol.html.

6. **Oracle Corporation.** MySQL :: MySQL 5.1 Reference Manual. [En línea] 2013. [Citado el: 07 de 05 de 2013.] http://dev.mysql.com/doc/refman/5.1/en/.

7. **Tridgell, Andrew.** "Efficient Algorithms for Sorting and Synchronization". Tesis de Doctorado. Universidad Nacional de Australia, 1999.

8. **Jacobson, Ivar, Boosh, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.* 1ra. Madrid : Addison Wesley, 2000.

9. **Juristo, Natalia, Moreno, Ana M. y Vegas, Sira.** Técnicas de evaluación de software. [En línea] 2006. [Citado el: 13 de 06 de 2013.] http://wotan.liu.edu/docis/dbl/ijseke/index.html.

10. **Schuartz, Baron y otros.** *High Performance MySQL.* 2da. s.l. : O'Reilly, 2008.

11. **Pressman, Roger S.** *Ingeniería del software. Un enfoque práctico.* [ed.] Darrel Ince. V. s.l. : McGraw-Hill, 2001.

12. **Cruz Mundet, José Ramón.** *Manual de archivística.* 2da. s.l. : Ediciones Pirámide, 1996.

13. **OAI.** Open Archives Initiative. [En línea] [Citado el: 07 de 07 de 2013.] http://www.openarchives.org/.

14. **PHP Group.** PHP: Hypertext Preprocessor. [En línea] [Citado el: 07 de 07 de 2013.] http://php.net/.

15. —. PHP: PHP Manual - Manual. [En línea] 05 de 07 de 2013. [Citado el: 07 de 07 de 2013.] http://php.net/manual/en/index.php.

16. **W3C.** Date and Time Formats. [En línea] [Citado el: 07 de 07 de 2013.] http://www.w3.org/TR/NOTE-datetime.

17. **Apache Software Foundation .** Apache JMeter - User's Manual. [En línea] 2013. [Citado el: 07 de 06 de 2013.] http://jmeter.apache.org/usermanual/index.html.

18. —. Apache JMeter - Apache JMeter™. [En línea] 2013. [Citado el: 07 de 06 de 2013.] http://jmeter.apache.org/.

19. **rsync Community.** rsync(1) - Linux man page. [En línea] [Citado el: 07 de 07 de 2013.] http://linux.die.net/man/1/rsync.

20. **Batchelor, Marc y Bleuel, Jens.** Latest Pentaho Data Integration (aka Kettle) Documentation - Pentaho Wiki. [En línea] 26 de 06 de 2013. [Citado el: 07 de 07 de 2013.] http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation;jsessionid=5711EA68A0B118E2B94DCB493590C721.

21. **Baker, Marina y Huddleston, Dan.** Spoon User Guide - Pentaho Wiki. [En línea] 11 de 02 de 2009. [Citado el: 07 de 07 de 2013.] http://wiki.pentaho.com/display/EAI/Spoon+User+Guide.

22. **DCMI.** DCMI Home: Dublin Core® Metadata Initiative (DCMI). [En línea] [Citado el: 07 de 03 de 2013.] http://dublincore.org/.

23. —. Dublin Core Metadata Element Set, Version 1.1. [En línea] [Citado el: 07 de 07 de 2013.] http://dublincore.org/documents/dces/.

24. **Drupal Community.** Documentation | drupal.org. [En línea] [Citado el: 08 de 07 de 2012.] https://drupal.org/documentation.

25. —. Drupal - Open Source CMS | drupal.org. [En línea] [Citado el: 02 de 05 de 2013.] https://drupal.org/.

26. —. Drupal 6 | API reference | Drupal API. [En línea] [Citado el: 07 de 06 de 2013.] https://api.drupal.org/api/drupal/6.

27. **Freund, John E., Miller, Irwin R. y Johnson, Richard.** *Probabilidades y estadísticas para ingenieros.* s.l. : Felix Varela, 2006. Vol. II.

28. **Freund, John E., Miller, Irwin R. y Johnson, Richard.** *Probabilidades y estadísticas para ingenieros.* s.l. : Felix Varela, 2006. Vol. I.

29. **Heredia Herrera, Antonia.** *Archivística General. Teoría y práctica.* 5ta. s.l. : Diputación Provincial de Sevilla, 1991.

30. **ICA.** ICArchives : Standards : ISAD(G): General International Standard Archival Description - Second edition. [En línea] [Citado el: 07 de 06 de 2013.] http://www.ica.org/?lid=10207.

31. —. ICArchives : Standards : ISDIAH: International Standard for Describing Institutions with Archival Holdings. [En línea] [Citado el: 07 de 06 de 2013.] http://www.ica.org/?lid=10198.

32. **Network Time Foundation.** ntp.org: Home of the Network Time Protocol. [En línea] 16 de 05 de 2013. [Citado el: 07 de 06 de 2013.] http://www.ntp.org/.

33. **NTP Community.** WebHome < Support < NTP. [En línea] 2013. [Citado el: 07 de 07 de 2013.] http://support.ntp.org/bin/view/Support/WebHome.

34. **OAI.** OAI-PMH Implementation Guidelines. [En línea] 05 de 05 de 2005. [Citado el: 07 de 01 de 2013.] http://www.openarchives.org/OAI/implementationguidelines.html.

35. **OAI, Accompanying Measures project (IST- 2001-320015).** Open Archives Forum - OAI-PMH Online Tutorial. [En línea] [Citado el: 07 de 01 de 2013.] http://www.oaforum.org/tutorial/.

36. **Oracle Corporation.** MySQL :: MySQL 5.0 Reference Manual. [En línea] [Citado el: 05 de 01 de 2013.] http://dev.mysql.com/doc/refman/5.0/en/.

37. —. MySQL :: MySQL 5.5 Reference Manual. [En línea] 2013. [Citado el: 07 de 06 de 2013.] http://dev.mysql.com/doc/refman/5.5/en/.

38. —. MySQL :: MySQL 5.6 Reference Manual. [En línea] 2013. [Citado el: 07 de 06 de 2013.] http://dev.mysql.com/doc/refman/5.6/en/.

39. —. MySQL :: The world's most popular open source database. [En línea] 2013. [Citado el: 17 de 05 de 2013.] http://www.mysql.com/.

40. **Pentaho Kettle Project.** Open Source ETL designed to bridge the gap between business and IT. | Kettle Project: Pentaho Data Integration. [En línea] [Citado el: 05 de 07 de 2013.] http://kettle.pentaho.com/.

41. **phpBB Group.** phpBB • Free and Open Source Forum Software. [En línea] 25 de 08 de 2012. [Citado el: 15 de 02 de 2013.] https://www.phpbb.com/.

42. —. phpBB • phpBB 3.0 Olympus Documentation. [En línea] 2008. [Citado el: 07 de 03 de 2013.] https://www.phpbb.com/support/documentation/3.0/.

43. **RAE.** Diccionario de la lengua española - Vigésima segunda edición. [En línea] [Citado el: 07 de 07 de 2013.] http://lema.rae.es/drae/.

44. **rsync Community.** rsync. [En línea] [Citado el: 07 de 05 de 2013.] http://rsync.samba.org/.

45. —. rsync documentation. [En línea] [Citado el: 07 de 06 de 2013.] http://rsync.samba.org/documentation.html.

46. **Elmasri, Ramez y Navathe, Shamkant B.** *Fundamentos de Sistemas de Bases de Datos.* 5ta. s.l. : Addison-Wesley, 2007.

47. **Grau, Ricardo, Correa, Cecilia y Rojas, Mauricio.** *Metodología de la investigación.* 2da. 2004. pág. 105.

48. **Hernández Sampieri, Roberto, Fernández-Collado, Carlos y Baptista Lucio, Pilar.** *Metodología de la investigación.* 4ta. s.l. : McGraw-Hill, 2006. pág. 882.

49. **Witten, Ian H., Bainbridge, David y Nichols, David M.** Interoperability: Protocols and services. [aut. libro] Ian H. Witten, David Bainbridge y David M. Nichols. *How to Build a Digital Library .* 2nd. Waikato : s.n., 2010, págs. 343-370.

50. *Collaborative knowledge management—A construction case study.* **Bhargav, Dave y Koskela, Lauri.** Manchester : s.n., 2009, Automation in Construction, Vol. 18, págs. 894-902.

51. **Consejo Internacional de Archivos.** ISDIAH Norma Internacional para Describir Instituciones que Custodian Fondos de Archivos. Madrid : s.n., 2008.

Anexos

Anexo 1. Descripción detallada del elemento resumptionToken

El formato del resumptionToken no está definido por OAI-PMH y debe ser transparente para los recolectores.

Los siguientes atributos opcionales pueden incluirse como parte del elemento resumptionToken:

- expirationDate: Indica cuando el resumptionToken deja de ser válido.
- completeListSize: Un entero indicando la cardinalidad de la lista completa.
- cursor: La cantidad de elementos de la lista completa retornados hasta el momento.

El siguiente ejemplo muestra una serie de solicitudes de tipo ListRecords en el cual la lista completa consta de 175 registros y el repositorio sólo devuelve 100 registros en cada respuesta:

- El recolector emite una petición de tipo ListRecords.
- El repositorio responde con una lista incompleta de 100 registros. Marca esta lista como incompleta mediante la inclusión en la respuesta de un elemento no vacío de tipo resumptionToken con dos atributos: completeListSize igual a 175 y cursor igual a cero.
- El recolector emite una siguiente solicitud de tipo ListRecords que incluye el resumptionToken recibido en la respuesta anterior.
- El repositorio responde con una lista incompleta de 75 registros. Marca esta lista como la última mediante la inclusión en la respuesta de un elemento de tipo resumptionToken vacío con dos atributos: completeListSize igual a 175 y cursor igual a 100.

Los repositorios que implementen resumptionToken deben hacerlo de manera tal que permita a los recolectores reanudar una secuencia de solicitudes de listas incompletas mediante el reenvío de una solicitud de lista con el resumptionToken más reciente. El propósito de esto es permitir a los recolectores recuperarse de algunos errores de red o de otro tipo que de otra forma implicarían que la solicitud de la lista debería comenzar nuevamente. El reenvío de una solicitud de lista con resumptionToken ocurre en dos contextos:

- Cuando no hay cambios en el repositorio: No hay cambios a la lista completa devuelta por la secuencia de solicitudes. En este caso el repositorio debe retornar la misma lista incompleta cuando se vuelva a emitir la solicitud más reciente, es decir, aquella con el último resumptionToken que no haya caducado.
- Cuando hay cambios en el repositorio: Estos cambios ocurren cuando los registros diseminados en la lista se mueven dentro o fuera del rango de fechas de la solicitud debido a cambios, modificaciones o borrados en el repositorio. En este caso no se requiere la idempotencia estricta de la solicitud empleando el resumptionToken. Por el contrario, la lista incompleta devuelta en respuesta al reenvío de solicitud debe incluir todos los registros con datestamps que no cambiaron dentro del rango de la petición original. La lista incompleta retornada en respuesta a una solicitud reemitida puede contener registros con datestamps que bien fueron movidos dentro o fuera del rango de la solicitud inicial. En casos de cambios sustanciales en el repositorio, puede resultar apropiado retornar un error del tipo badResumptionToken indicando al recolector que debe reiniciar la secuencia de solicitudes.

Anexo 2. Descripción de los códigos de error y los verbos a los que aplican

Código de error (*code*)	Descripción	Verbos a los que aplica
badArgument	La solicitud incluye argumentos ilegales, faltan argumentos requeridos, incluye argumentos repetidos o los valores de los argumentos tienen error de sintaxis.	Todos los verbos
badResumptionToken	El valor del argumento *resumptionToken* es inválido o ha expirado.	*ListIdentifiers* *ListRecords* *ListSets*
badVerb	El valor del argumento *verb* no es un verbo OAI-PMH legal, falta el argumento *verb*, o el argumento *verb* se repite.	No aplica
cannotDisseminateFormat	El formato de metadatos identificado por el valor del argumento *metadataPrefix* no es soportado por el *item* o por el repositorio.	*GetRecord* *ListIdentifiers* *ListRecords*
idDoesNotExist	El valor del argumento *identifier* es desconocido o ilegal en el repositorio.	*GetRecord* *ListMetadataFormats*
noRecordsMatch	La combinación de los valores de los argumentos *from*, *until*, *set* y *metadataPrefix* dan como resultado una lista vacía.	*ListIdentifiers* *ListRecords*
noMetadataFormats	No hay formatos de metadatos disponibles para el *item* especificado.	*ListMetadataFormats*
noSetHierarchy	El repositorio no soporta *sets*.	*ListSets* *ListIdentifiers* *ListRecords*

Anexo 3. Solicitudes y respuestas del protocolo OAI-PMH

En esta sección se describen los tipos de solicitudes, o verbos, definidos en OAI-PMH, haciendo referencia en cada caso a:

- Una sección título correspondiente al token empleado para especificar el tipo de la solicitud mediante el argumento requerido verb para la solicitud HTTP.
- Un breve resumen del significado del verbo y notas sobre su uso.
- Una lista de los argumentos adicionales para la solicitud. Estos son de tres tipos:
    - requerido, el argumento debe incluirse en la solicitud.
    - opcional, el argumento puede incluirse en la solicitud.
    - exclusivo, el argumento puede incluirse en la solicitud, pero debe ser el único, además del argumento verb.
- Condiciones de error o excepción específicas del tipo de solicitud.

GetRecord

Resumen y notas de uso

Este verbo se emplea para recuperar los metadatos de un registro individual del repositorio. Los argumentos requeridos especifican el identificador del ítem del cual se solicita el registro y el formato de metadatos que debe incluirse en el registro.

Argumentos

- identifier, argumento requerido que especifica el identificador único del ítem dentro del repositorio a partir del cual se disemina el *record*.
- metadataPrefix, argumento requerido que indica el formato de los metadatos que deben ser incluidos en la sección de metadatos del registro retornado. El registro sólo se retorna si el formato especificado es soportado por el repositorio para el ítem.

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales o faltan argumentos requeridos.
- cannotDisseminateFormat, el valor del argumento *metadataPrefix* no es soportado por el ítem identificado por el valor del argumento identifier.
- idDoesNotExists, el valor del argumento identifier es desconocido o ilegal dentro del repositorio.

Identify

Resumen y notas de uso

Este verbo se utiliza para recuperar información acerca de un repositorio. Parte de la información que se devuelve es obligatoria dentro del protocolo OAI-PMH. Los repositorios también pueden emplear el verbo Identify para devolver información descriptiva adicional.

Argumentos

- Ninguno

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales.

ListIdentifiers

Resumen y notas de uso

Este verbo es una forma abreviada de ListRecords, recuperando solamente los encabezados en lugar de los registros. Los argumentos opcionales permiten la recolección selectiva de encabezados basada en la membresía de los conjuntos y/o las marcas de tiempo.

Argumentos

- from, argumento opcional que especifica el límite inferior para la recolección selectiva basada en marcas de tiempo.
- until, argumento opcional que especifica el límite superior para la recolección selectiva basada en marcas de tiempo.
- metadataPrefix, argumento requerido que especifica que los encabezados sólo deben ser retornados si está disponible el formato de metadatos especificado.
- set, argumento opcional que especifica un conjunto como criterio para la recolección selectiva.
- resumptionToken, argumento exclusivo con un valor igual al token de control de flujo devuelto por una solicitud ListIdentifiers anterior que emitió una lista incompleta.

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales o faltan argumentos requeridos.

- badResumptionToken, el valor del argumento resumptionToken es inválido o ha caducado.
- cannotDisseminateFormat, el valor del argumento metadataPrefix no es soportado por el repositorio.
- noRecordsMatch, la combinación de los valores de los argumentos from, until y set dan como resultado en una lista vacía.
- noSetHierarchy, el repositorio no soporta sets.

ListMetadataFormats

Resumen y notas de uso

Este verbo se utiliza para recuperar los formatos de metadatos soportados por el repositorio. Un argumento opcional restringe la solicitud a los formatos disponibles para determinado ítem.

Argumentos

- identifier, argumento opcional que especifica el identificador único del componente del cual se han solicitado los formatos de metadatos disponibles. Si este argumento se omite entonces la respuesta incluye todos los formatos de metadatos soportados por el repositorio. Nótese que el hecho de que determinado formato de metadato sea soportado por el repositorio no significa que deba ser diseminado para todos sus componentes.

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales o faltan argumentos obligatorios.
- idDoesNotExists, el valor del argumento identifier es desconocido o ilegal en el repositorio.
- noMetadataFormats, no existen formatos de metadatos disponibles para el ítem especificado.

ListRecords

Resumen y notas de uso

Este verbo se emplea para recolectar registros del repositorio. Los argumentos opcionales permiten la recolección selectiva de registros basada en la pertenencia a conjuntos y/o marcas de tiempo.

Argumentos

- from, argumento opcional con un valor que especifica el límite inferior para la recolección selectiva basada en marcas de tiempo.
- until, argumento opcional con un valor que especifica el límite superior para la recolección selectiva basada en marcas de tiempo.
- set, argumento opcional con un valor setSpec que especifica un *set* como criterio para la recolección selectiva.
- resumptionToken, argumento exclusivo con un valor que representa un token de control de flujo devuelto por una solicitud previa de tipo ListRecords que ha retornado una lista incompleta.
- metadataPrefix, argumento requerido (a menos que el argumento exclusivo resumptionToken sea utilizado).

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales o faltan argumentos obligatorios.
- badResumptionToken, el valor del argumento resumptionToken es inválido o ha caducado.
- cannotDisseminateFormat, el valor del argumento metadataPrefix no es soportado por el repositorio.
- *noRecordMatch*, la combinación de los valores de los argumentos from, until, set y metadataPrefix dan como resultado una lista vacía.
- noSetHierarchy, el repositorio no soporta conjuntos.

ListSets

Resumen y notas de uso

Este verbo se emplea para recuperar la estructura de conjuntos de un repositorio siendo de utilidad para la recolección selectiva.

Argumentos

- resumptionToken, argumento exclusivo con un valor que representa el token para el control de flujo retornado por una solicitud previa de tipo ListSets que ha emitido una lista incompleta.

Condiciones de error o excepción

- badArgument, la solicitud incluye argumentos ilegales o faltan argumentos obligatorios.
- badResumptionToken, el valor del argumento resumptionToken es inválido o ha caducado.
- noSetHierarchy, el repositorio no soporta conjuntos.

Anexo 4 Descripción detallada de las principales relaciones del sistema ArchiVenHIS que intervienen en la ejecución del proceso de búsqueda

Tabla 5.1 Relación expediente.

Nombre: expediente		
Descripción: Nomenclador que regula los tipos de expediente para la descripción de las agrupaciones documentales.		
Atributo	Tipo	Descripción
idexpediente	int(10)	Identificador de la relación expediente
nombre	varchar(50)	Nombre del tipo de expediente.

Tabla 5.2 Relación materia.

Nombre: materia		
Descripción: Nomenclador que regula las materias para la descripción de las agrupaciones documentales.		
Atributo	Tipo	Descripción
idmateria	int(10)	Identificador de la relación materia.
nombre	varchar(50)	Nombre de la materia.

Tabla 5.3 Relación tipología.

Nombre: tipologia		
Descripción: Nomenclador que regula las tipologías para la descripción de las agrupaciones documentales.		
Atributo	Tipo	Descripción
idtipologia	int(10)	Identificador de la relación tipología.
nombre	varchar(50)	Nombre de la tipología.

Tabla 5.4 Relación descripción.

Nombre: descripcion		
Descripción: Contiene la descripción, según la norma ISAD(G) de los fondos documentales.		
Atributo	Tipo	Descripción
iddescripcion	int(10)	Identificador de la relación descripcion.
idusuario_describe	int(10)	Identificador del usuario que elabora la descripción.
titulo12	varchar(250)	Contiene el valor relativo al campo “Título” del área “Identificación” de la norma ISAD(G) para denominar la unidad de descripción.
fi13	date	Contiene información relativa al campo “Fecha(s)” del área “Identificación” de la norma ISAD(G) para identificar y consignar la fecha de inicio de la unidad de descripción.
dfi13	tinyint(1)	Indica si la fecha de inicio es aproximada.
ff13	date	Contiene información relativa al campo “Fecha(s)” del área “Identificación” de la norma ISAD(G) para identificar y consignar la fecha final de la unidad de descripción.
dff13	tinyint(1)	Indica si la fecha final es aproximada.
soporte15	varchar(200)	Contiene información relativa al campo “Volumen y soporte de la unidad de descripción” del área “Identificación” de la norma ISAD(G) para identificar y describir la extensión física o lógica y el soporte de la unidad de descripción.
historia_archivistica23	varchar	Contiene información relativa al campo “Historia archivística” del área “Contexto” de la norma ISAD(G) para proporcionar información sobre la historia de la unidad de descripción.
formaingreso24	varchar	Contiene información relativa al campo “Forma de ingreso” del área “Contexto” de

		la norma ISAD(G) para identificar la forma de adquisición o transferencia.
alcance31	varchar	Contiene información relativa al campo “Alcance y contenido” del área “Contenido y estructura” de la norma ISAD(G) para proporcionar a los usuarios la información necesaria para apreciar el valor potencial de la unidad de descripción.
valoracion32	varchar	Contiene información relativa al campo “Valoración, selección y eliminación” del área “Contenido y estructura” de la norma ISAD(G) para proporcionar información sobre cualquier acción de valoración, selección y eliminación efectuada.
nuevo_ingreso33	tinyint(1)	Indica si se esperan nuevos ingresos relativos a la unidad de descripción.
nuevo_ing_desc33	varchar(200)	Contiene información relativa al campo “Nuevos ingresos” del área “Contenido y estructura” de la norma ISAD(G) para informar al usuario de los ingresos complementarios previstos relativos a la unidad de descripción.
organizacion34	varchar(1000)	Contiene información relativa al campo “Organización” del área “Contenido y estructura” de la norma ISAD(G) para informar sobre la estructura interna de la unidad de descripción.
idexpediente35	int(10)	Llave foránea que hace referencia al nomenclador expediente.
onomastico36	varchar(1000)	Descriptor onomástico adicional solicitado como requisito por el AGN.
geografico36	varchar(1000)	Descriptor geográfico adicional solicitado como requisito por el AGN.
institucional36	varchar(1000)	Descriptor institucional adicional solicitado como requisito por el AGN.
idtipologia36	int(10)	Llave foránea que hace referencia al

		nomenclador tipología.
idmateria36	int(10)	Llave foránea que hace referencia al nomenclador materia.
acceso41	tinyint(1)	Indica si la agrupación documental será de acceso público.
cond_acceso41	varchar(200)	Contiene información relativa al campo “Condiciones de acceso” del área “Condiciones de acceso y utilización” de la norma ISAD(G) para informar sobre la situación jurídica y cualquier otra normativa que restrinja o afecte el acceso a la unidad de descripción.
reproduccion42	tinyint(1)	Indica si es posible reproducir libremente la unidad de descripción.
cond_reproduccion42	varchar(200)	Contiene información relativa al campo “Condiciones de reproducción” del área “Condiciones de acceso y utilización” de la norma ISAD(G) para identificar cualquier tipo de restricción relativa a la reproducción de la unidad de descripción.
escritura43	varchar(200)	Contiene información relativa al campo “Lengua/escritura(s) de los documentos” del área “Condiciones de acceso y utilización” de la norma ISAD(G) para identificar la(s) lengua(s), escritura(s) y sistemas de símbolos utilizados en la unidad de descripción.
requisitos44	varchar(200)	Contiene información relativa al campo “Características físicas y requisitos técnicos” del área “Condiciones de acceso y utilización” de la norma ISAD(G) para informar sobre cualquier característica física o requisito técnico de importancia que afecte al uso de la unidad de descripción.
inst_descrip45	varchar(200)	Contiene información relativa al campo “Instrumentos de descripción” del área

		“Condiciones de acceso y utilización” de la norma ISAD(G) para indicar cualquier tipo de instrumento de descripción relativo a la unidad de descripción.
originales51	varchar(200)	Contiene información relativa al campo “Existencia y localización de los documentos originales” del área “Documentación asociada” de la norma ISAD(G) para, en el caso de que la unidad de descripción esté formada por copias, indicar la existencia, localización, disponibilidad y/o eliminación de los originales.
copias52	varchar(200)	Contiene información relativa al campo “Existencia y localización de copias” del área “Documentación asociada” de la norma ISAD(G) para indicar la existencia, localización y disponibilidad de copias de la unidad de descripción.
relacionadas53	varchar(500)	Contiene información relativa al campo “Unidades de descripción relacionadas” del área “Documentación asociada” de la norma ISAD(G) para identificar las unidades de descripción relacionadas.
notas_publicacion54	varchar(500)	Contiene información relativa al campo “Nota de publicaciones” del área “Documentación asociada” de la norma ISAD(G) para identificar cualquier tipo de publicación que trate o esté basada en el uso, estudio o análisis de la unidad de descripción.
notas61	varchar(1000)	Contiene información relativa al campo “Notas” del área “Notas” de la norma ISAD(G) para dar información que no haya podido ser incluida en ninguna de las otras áreas.
nota_archivero71	varchar(500)	Contiene información relativa al campo

		“Nota del archivero” del área “Control de la descripción” de la norma ISAD(G) para explicar quién y cómo ha preparado la descripción.
normas_desc72	varchar(150)	Contiene información relativa al campo “Reglas o normas” del área “Control de la descripción” de la norma ISAD(G) para identificar la normativa en la que está basada la descripción.
fecha_desc73	date	Contiene información relativa al campo “Fecha(s) de la(s) descripción(es)” del área “Control de la descripción” de la norma ISAD(G) que indica cuándo se ha elaborado y/o revisado la descripción.

Tabla 5.5 Relación tipo_nivel.

Nombre: tipo_nivel		
Descripción: Nomenclador que regula los tipos de nivel de organización (fondo, subfondo, serie, etc) del patrimonio documental resguardado por el Archivo.		
Atributo	Tipo	Descripción
idtnivel	int(10)	Identificador de la relación tipo_nivel
nivel	varchar(50)	Nombre del tipo de nivel.

Tabla 5.6 Relación nivel_organizacion.

Nombre: nivel_organizacion		
Descripción: Forma genérica de representar las instancias de las agrupaciones documentales sin distinción de tipo.		
Atributo	Tipo	Descripción
idnivel	int(10)	Identificador de la relación nivel_organizacion
iddescripcion	int(10)	Llave foránea que hace referencia a la relación descripción asociada a este nivel de organización.
idtnivel	int(10)	Llave foránea que hace referencia al nomenclador tipo_nivel.
idnivel_padre	int(10)	Llave foránea que hace referencia a la

		propia relación nivel_organizacion, indicando el padre, en la jerarquía de niveles, de este nivel de organización.

Tabla 5.7 Relación nivel_contenedor.

Nombre: nivel_contenedor		
Descripción: Especialización de la relación nivel_organizacion que contiene los nombres y acrónimos de las agrupaciones documentales que conforman la estructura del cuadro de clasificación.		
Atributo	Tipo	Descripción
idnivel	int(10)	Identificador de la relación nivel_contenedor y además llave foránea que hace referencia a la relación nivel_organizacion.
nombre	varchar(100)	Nombre del nivel contenedor.
acronimo	varchar(15)	Acrónimo del nivel contenedor.

Tabla 5.8 Relación archivo_digital.

Nombre: archivo_digital		
Descripción: Contiene la información correspondiente a las representaciones digitales de los documentos que permite determinar su ubicación dentro del sistema de archivos, relativa al repositorio de ArchiVenHIS y conocer si están disponibles para la consulta pública.		
Atributo	Tipo	Descripción
idarchivo_digital	int(10)	Identificador de la relación archivo_digital
idusuario_sube	int(10)	Llave foránea que hace referencia al digitalizador que establece el vínculo entre la representación digital y un documento previamente descrito en el sistema.
iddocumento	int(10)	Llave foránea que hace referencia al documento al que se asocia la representación digital.
idusuario_aprueba	int(10)	Llave foránea que hace referencia al usuario que aprueba la disponibilidad de la representación digital para la consulta pública.
nombre_original	varchar(100)	Indica el nombre del fichero que contiene la representación digital antes de incluirse en

		el sistema.
extension	char(4)	Extensión del fichero que contiene la representación digital.
fecha_subido	datetime	Fecha en la que el fichero que contiene la representación digital se incorpora al sistema
fecha_aprobado	datetime	Fecha en la que se aprueba la consulta pública de la representación digital.
representa_imagen	tinyint(1)	Indica si la representación digital se corresponde con el documento en sí o con su transcripción paleográfica.

Anexo 5

5.1 Fichero específico de log binario en el maestro y la posición dentro de este último a partir de la cual se debe obtener la información a replicar

--

-- Position to start replication or point-in-time recovery from

--

-- CHANGE MASTER TO MASTER_LOG_FILE='mysql-bin.000004', MASTER_LOG_POS=106;

5.2 Salida del comando mysql> show slave status\G

************************** 1. row ***************************

Slave_IO_State: Waiting for master to send event

Master_Host: 10.1.1.101

Master_User: user_replica

Master_Port: 3306

Connect_Retry: 10

Master_Log_File: mysql-bin.000006

Read_Master_Log_Pos: 22008

Relay_Log_File: mysqld-relay-bin.000011

Relay_Log_Pos: 22145

Relay_Master_Log_File: mysql-bin.000006

Slave_IO_Running: Yes

Slave_SQL_Running: Yes

Replicate_Do_DB: archivenhis

Replicate_Ignore_DB:

Replicate_Do_Table:

Replicate_Ignore_Table:

Replicate_Wild_Do_Table:

Replicate_Wild_Ignore_Table:

Last_Errno: 0

Last_Error:

Skip_Counter: 0

Exec_Master_Log_Pos: 22008

Relay_Log_Space: 22145

Until_Condition: None

Until_Log_File:

Until_Log_Pos: 0

Master_SSL_Allowed: No

Master_SSL_CA_File:

Master_SSL_CA_Path:

Master_SSL_Cert:

Master_SSL_Cipher:

Master_SSL_Key:

Seconds_Behind_Master: 0

1 row in set (0.00 sec)

Anexo 6. Configuración de los componentes utilizados en el plan de pruebas desarrollado empleando JMeter

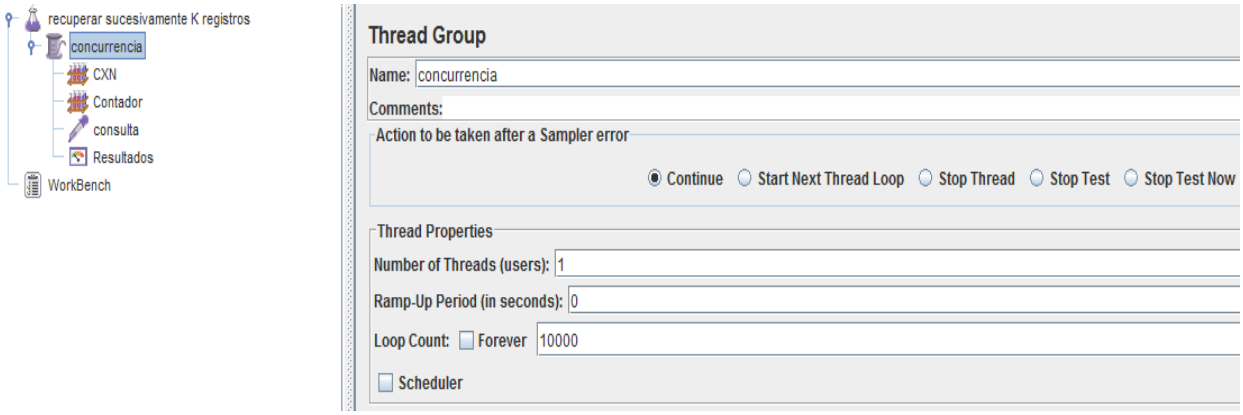

Figura 6.1 Configuración del componente Thread Group en el plan de prueba desarrollado empleando JMeter.

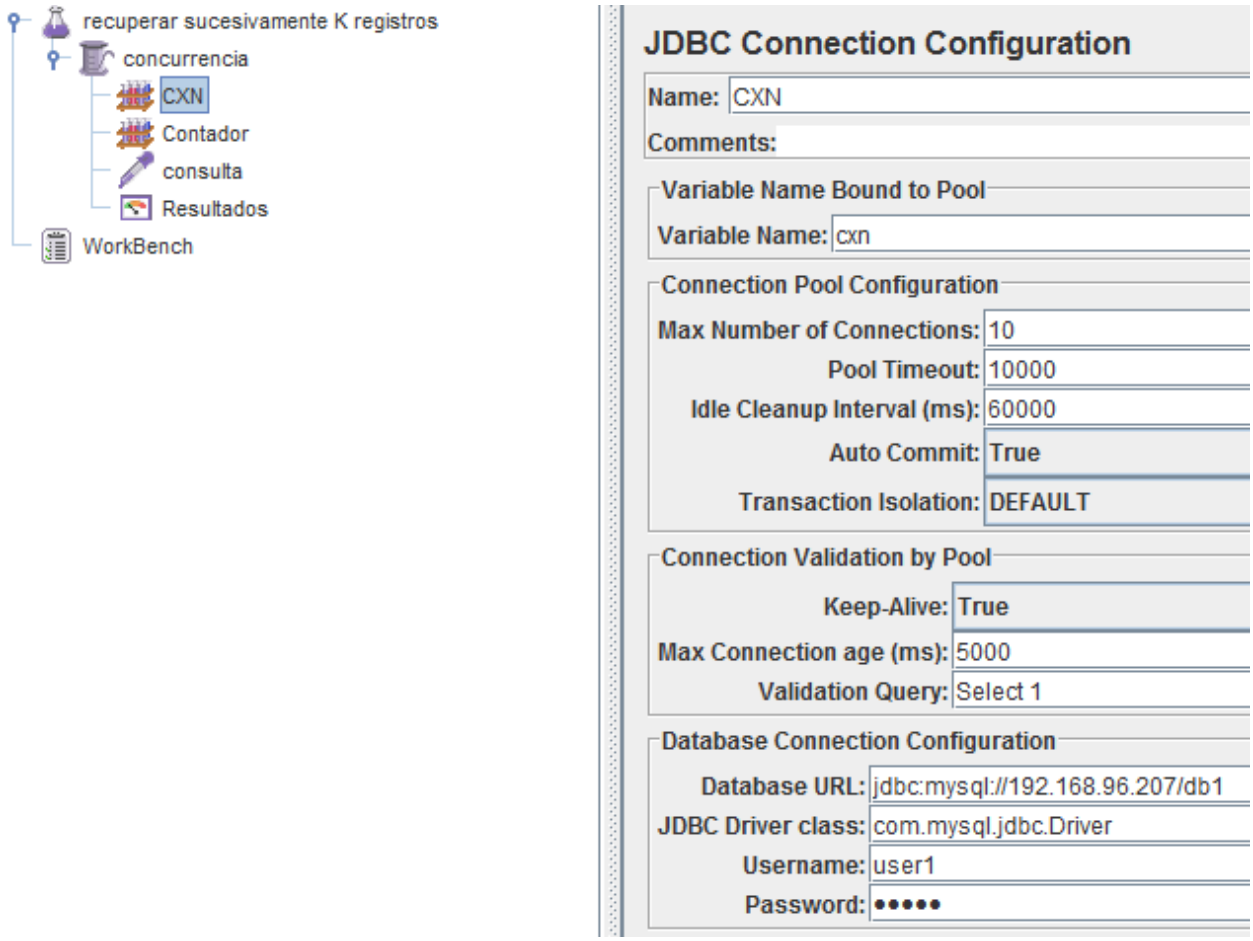

Figura 6.2 Configuración del componente JDBC Connection Configuration en el plan de prueba desarrollado empleando JMeter.

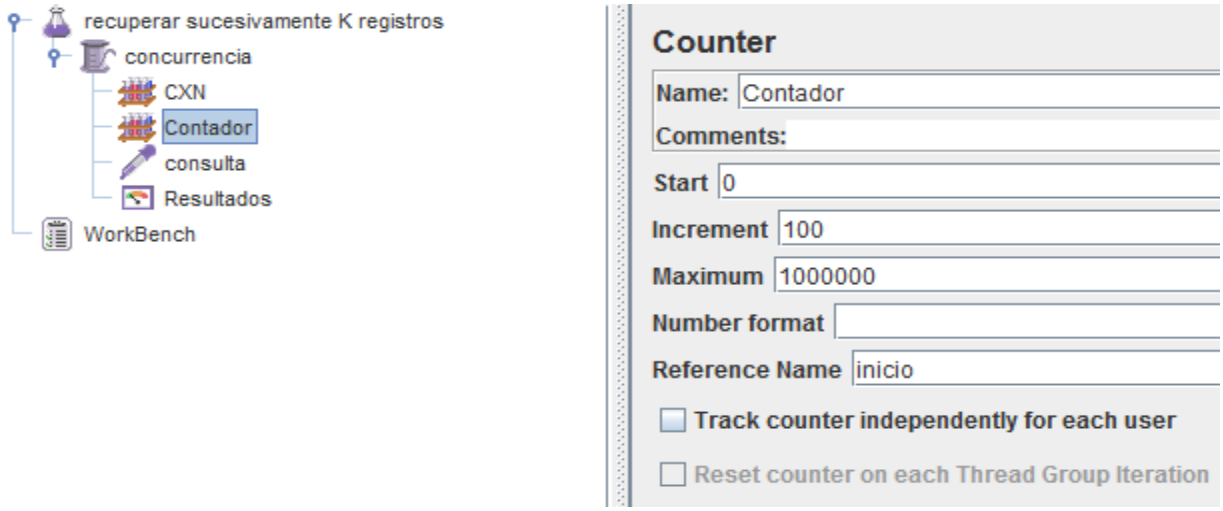

Figura 6.3 Configuración del componente Counter en el plan de prueba desarrollado empleando JMeter.

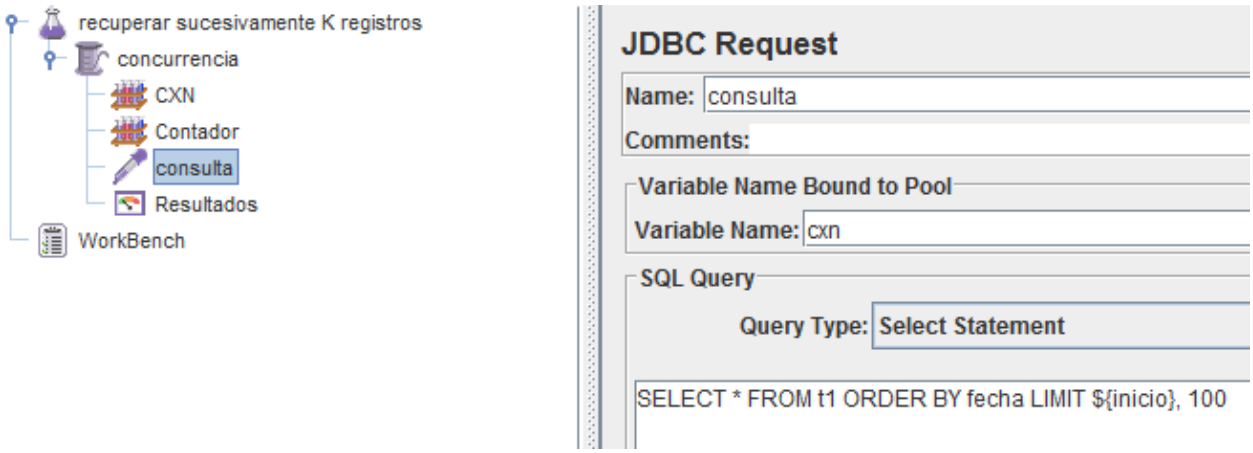

Figura 6.4 Configuración del componente JDBC Request en el plan de prueba desarrollado empleando JMeter.

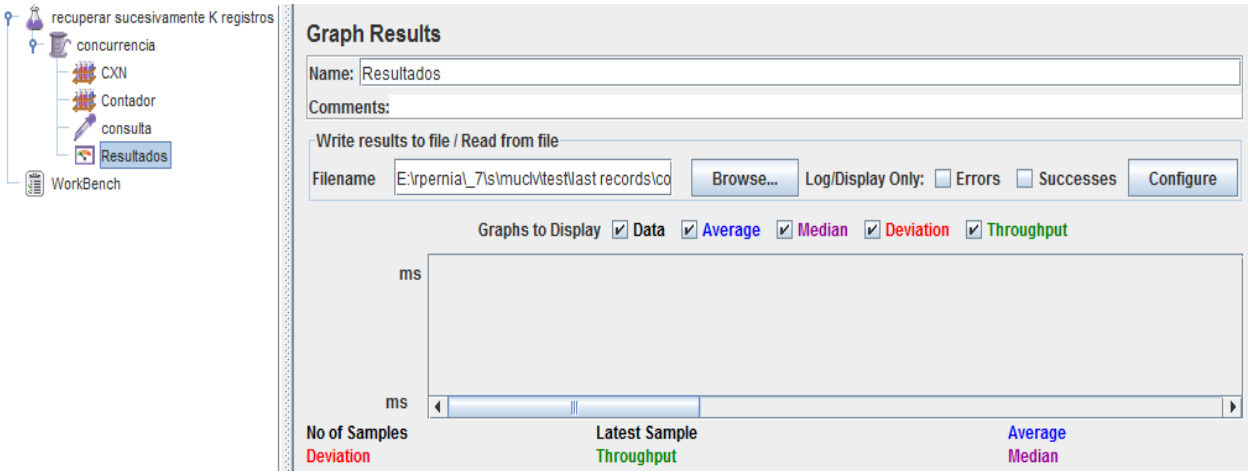

Figura 6.5 Configuración del componente Graph Results en el plan de prueba desarrollado empleando JMeter.