



Universidad de las Ciencias Informáticas

Facultad 2

Centro de Informatización de la Gestión Documental

(CIGED)

Módulo Reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor:

Yaidel Luis Araujo Collado

Tutor(s):

MSc. Madelis Pérez Gil

Ing. Wisbel Sánchez de la Noval

Ing. José Javier Hernández Benítez

La Habana, 10 de junio de 2019

“Año 61 de la Revolución”

“La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica.”

Aristóteles



Declaración de autoría

Declaro ser único autor de la presente tesis que tiene por título: Módulo Reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2 y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de junio del año 2019.

Yaidel Luis Araujo Collado

Firma del Autor

MSc. Madelis Pérez Gil

Firma del Tutor

Ing. Wisbel Sánchez de la Noval

Firma del Tutor

Ing. José J. Hernández Benítez

Firma del Tutor

Datos de contacto

MS.c. Madelis Pérez Gil: Ingeniera en Ciencias Informáticas 2008. Máster en Gestión de Información de la Facultad de Economía / Universidad de La Habana / Cátedra UNESCO 2014. Profesora Auxiliar del Dpto. de Ingeniería de Software de la Facultad 2.

Correo: mgil@uci.cu

Universidad de las Ciencia Informáticas, La Habana, Cuba.

Ing. Wisbel Sánchez de la Noval: Graduado en el año 2016 como Ingeniero en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña como especialista B en Ciencias Informáticas en el Departamento de Desarrollo de Aplicaciones del Centro CIGED. Ha desempeñado el rol de desarrollador en el proyecto Arkheia.

Correo: wdelanoval@uci.cu

Universidad de las Ciencia Informáticas, La Habana, Cuba.

Ing. José J. Hernández Benítez: Graduado en el año 2016 como Ingeniero en Ciencias Informáticas, en la Universidad de las Ciencias Informáticas. Se desempeña como especialista B en Ciencias Informáticas en el Departamento de Desarrollo de Aplicaciones del Centro CIGED. Ha desempeñado el rol de desarrollador en el proyecto Repositorio Institucional.

Correo: jjbenitez@uci.cu

Universidad de las Ciencia Informáticas, La Habana, Cuba

Dedicatoria

Quiero dedicar esta tesis a todas las personas que hicieron posible este momento tan importante.

Agradecimientos

Hoy cierra un capítulo en el libro de la vida, uno que ha necesitado muchas horas de sacrificio y dedicación, no solo mías, sino también de aquellas personas que de una forma u otra ayudaron a que este momento que un día fue un sueño hoy se convierta en realidad

Resumen

El Centro de Informatización de la Gestión Documental (CIGED por sus siglas en español), es una organización especializada en el desarrollo de sistemas y servicios informáticos integrales de alta calidad y competitividad en la informatización de los procesos de gestión documental. El mismo cuenta con varios productos entre ellos el Sistema para Repositorios Digitales (XABAL Repxos), que no es más que un sistema web que permite almacenar y difundir la producción intelectual digital generada por una institución, el cual cuenta con un módulo de Reportes. Este módulo permite la creación de reportes de forma dinámica para su posterior análisis por los usuarios del sistema. Debido a la necesidad de creación de una suite de Gestión Documental para el Centro de Informatización de la Gestión Documental, se hace necesario el desarrollo de este módulo con las tecnologías que sean compatibles con otros productos desarrollados en el centro, como por ejemplo XABAL Arkehia. Es por ello que se decide el desarrollo de este módulo utilizando la tecnología del Share Alfresco Community en su versión 5.2 logrando así la integración de los procesos de gestión documental. Para lograr este objetivo se seleccionó una metodología de desarrollo de software para guiar el proceso de creación del módulo seleccionando además lenguajes de programación, herramientas informáticas y tecnologías acordes para el desarrollo. Se definieron además un conjunto de pruebas a realizar para comprobar el correcto desempeño del módulo una vez este sea implementado.

Palabras clave: Alfresco Community, gestión documental, módulo reportes, Repxos.

TABLA DE CONTENIDOS

Introducción	1
CAPÍTULO 1. Fundamentación Teórica	5
1.1. Introducción	5
1.2. Conceptos fundamentales	5
1.3. Metodología	14
1.4. Herramientas informáticas, tecnologías y lenguajes de programación	15
CAPÍTULO 2. Análisis y diseño	24
2.1. Introducción	24
2.2. Descripción de la solución propuesta	24
2.3. Modelo de dominio	24
2.4. Diagrama de paquetes	26
2.5. Especificación de requisitos de software	27
2.6. Definiciones de actores	32
2.7. Diagramas de casos de uso del sistema	32
2.8. Interfaces del sistema	46
2.9. Matrices de trazabilidad	49
2.10. Descripción de la arquitectura	51
CAPÍTULO 3. Implementación y pruebas	55
3.1. Introducción	55
3.2. Implementación	55
3.3. Estándares de codificación	57
3.4. Pruebas de software	57
Conclusiones GeNERALES	74
Recomendaciones	75
Referencias Bibliográficas	76
Bibliografía	79

Anexos.....82

ÍNDICE DE FIGURAS

Figura 1 Distribución del tipo de software empleado para la creación de repositorios en OpenDOAR (SEDIC)	7
Figura 2 Modelo de Dominio del Módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.	25
Figura 3 Diagrama de paquetes.....	27
Figura 4 Diagrama de Caso de Usos del Sistema.....	33
Figura 5 Interfaces del sistema de los casos de uso "Gestionar Servidor Solr " y "Buscar Servidor Solr"	47
Figura 6 Interfaces del sistema de los casos de uso "Gestionar Consulta Solr " y "Buscar Consulta Solr"	47
Figura 7 Interfaces del sistema de los casos de uso "Gestionar Reporte"	48
Figura 8 Interfaces del sistema de los casos de uso "Buscar Reporte"	48
Figura 9 Matriz de trazabilidad Requisito-Requisito	49
Figura 10 Matriz de trazabilidad Caso de Uso-Requisito.....	50
Figura 11 Matriz de trazabilidad Caso de Uso- Caso de Prueba	50
Figura 12 Diagrama de despliegue	56
Figura 13 Fragmento de código desarrollado.....	63
Figura 14 Representación del grafo de flujo de camino básico de aplicar matriz de concurrencia.	63

ÍNDICE DE TABLAS

Tabla 1 Requisitos Funcionales del módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.....	29
Tabla 2 Diagrama de CU Gestionar Consulta Solr.....	34
Tabla 3 Diagrama de CU Buscar Consulta Solr.....	37
Tabla 4 Diagrama de CU Gestionar Reporte.....	38
Tabla 5 Diagrama de CU Buscar Consulta Solr.....	41
Tabla 6 Diagrama de CU Buscar Reporte.....	42
Tabla 7 Diagrama de CU Gestionar Servidor Solr.....	43
Tabla 8 Diagrama de CU Exportar a formato PDF.....	46
Tabla 9 Casos de prueba de caja blanca para los caminos básicos.....	65
Tabla 10 Diseño de caso de prueba del caso de uso “Gestionar Servidor”.....	66
Tabla 11 Diseño de caso de prueba del caso de uso “Gestionar Consulta”.....	67
Tabla 12 Diseño de caso de prueba del caso de uso “Buscar Consulta”.....	69
Tabla 13 Diseño de caso de prueba del caso de uso “Buscar Servidor”.....	70
Tabla 14 Diseño de caso de prueba del caso de uso “Buscar Reporte”.....	70
Tabla 15 Diseño de caso de prueba del caso de uso “Exportar Reporte a .pdf”.....	71
Tabla 16 Diseño de caso de prueba del caso de uso “Gestionar Reporte”.....	71

INTRODUCCIÓN

La información es un pilar importante que incurre en el éxito o fracaso directo de las actividades de los seres humanos. A través de la historia el hombre ha sentido la necesidad de transmitir sus conocimientos a otras personas y que estos quedaran plasmados para futuras generaciones. A partir de ahí y con el desarrollo constante de la sociedad surgen las Tecnologías de la Información y las Comunicaciones (TICs), que corresponden y se refieren a todas las tecnologías que de una u otra forma interfieren y median en los procesos informacionales y comunicativos entre seres humanos. Pueden ser entendidas como un conjunto de recursos tecnológicos integrados entre sí, que proporcionan, por medio de facilidades de hardware, software, y de telecomunicaciones, la semi-automatización y comunicación de procesos relativos a negocios.

El volumen de documentación generado en las instituciones se incrementa debido, entre otros factores, al desarrollo propio de las TICs; por lo que mantenerlo ordenado y al alcance de todos, se torna difícil. Es a través del estudio profundo y la puesta en práctica de la ciencia archivística que se podrán llevar de manera eficiente los procesos de gestión documental. La archivística es la ciencia de los archivos, que como tal ciencia está integrada por un conjunto de conocimientos y de métodos para el tratamiento de los documentos y de los archivos. Mientras que la Gestión Documental (GD) se encarga de proveer un control sistemático sobre los documentos archivísticos para documentar los procesos de negocios, la toma de decisiones y las transacciones. (Rodríguez Cruz, Castellanos Crespo et al. 2016).

La expansión tecnológica también ha sentado sus bases en la gestión archivística y en gran medida se debe a la capacidad de esta integrarse con otras disciplinas como la historia, la administración, el derecho y la informática. Con el uso de las TICs se ha logrado fusionar los procesos tradicionales con las mismas, lográndose preservar documentos, información, y la memoria histórica de las naciones.

Cuba no se encuentra exenta de este ámbito documental tecnológico, pero estudios y diagnósticos llevados a cabo por especialistas en el tema demuestran que aún existen tensiones entre la cultura del papel y los avances tecnológicos. El país se ha dado a la tarea de establecer programas de GD en las organizaciones, como un elemento necesario para el aumento de la eficiencia, la productividad y el cumplimiento de sus objetivos. Alineados a dichos objetivos se encuentra la Universidad de Ciencias Informáticas (UCI) que es una de las instituciones que contribuye al avance de las tecnologías y la informatización del país, formando profesionales altamente calificados en la rama de la informática y desarrollando software para diferentes

entidades. Fue fundada en el año 2002, y ha llevado paralelamente las actividades académicas y productivas. Actualmente cuenta con 14 centros de desarrollo, entre los cuales está el Centro de Informatización de la Gestión Documental (CIGED), que se encarga del desarrollo de sistemas y servicios informáticos dedicados a la informatización o mejora de la gestión documental y el cual tiene entre sus productos, además del Sistema de Gestión de Archivos Históricos (Arkehia) y un Gestor de Documentos Administrativos (eXcriba), un Sistema para Repositorios Digitales (Repxos), que no es más que un sistema web que permite almacenar y difundir la producción intelectual digital generada por una institución. Actualmente en el centro CIGED se están llevando a cabo la evolución de otros software en tecnologías diferentes y como meta del centro, la UCI y el país, se quiere la integración de los mismos en una suit la cual se nombra Plataforma Cubana de Gestión Documental y Archivística la cual cumpla con el enfoque de archivística integrada brindando la posibilidad de gestionar por completo el flujo de información en una entidad, posibilitando la liberación de contenido al público para su consulta a través del uso de un repositorio digital y de esa manera preservar información sensible para la sostenibilidad y desarrollo de la entidad. Por este motivo se hace de vital importancia, además de la inclusión de un repositorio digital en la suit, la migración hacia una misma tecnología, escogiéndose para ello Alfresco Community en su versión 5.2, migrando el módulo de reportes de XABAL Repxos a la tecnología del Share de Alfresco Community logrando así parte de dicha integración y con el fin de obtener, a través de consultas a servidores solr, datos en forma de reportes que puedan ser usados para su posterior análisis.

Por lo que se define como **problema de investigación**: ¿Cómo lograr que la generación de reportes de XABAL Repxos se integre con la tecnología del Share de Alfresco Community 5.2? Se tiene como **objeto de estudio**: El proceso de generación de reportes. Para dar solución al problema se plantea como **objetivo general**: Desarrollar el módulo reportes del sistema XABAL Repxos utilizando la tecnología del Share de Alfresco Community 5.2; teniendo como **campo de acción**: La generación de reportes en el sistema XABAL Repxos, definiendo para la guía y orientación en el desarrollo del proceso investigativo las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos a tener en cuenta para abordar la solución del problema planteado relacionado con el módulo reportes del Sistema para Repositorios Digitales XABAL Repxos de la UCI?
2. ¿Qué propuesta de solución se define para implantar en el módulo reporte en la tecnología del Share

de Alfresco Community 5.2?

3. ¿Cómo desarrollar un módulo reportes en el Share de Alfresco Community 5.2 que permita la generación de reportes de usuarios logrando así una integración de las tecnologías del centro en una suit?
4. ¿Cómo se valida el correcto funcionamiento del módulo reportes del sistema XABAL Repxos en la tecnología del Share de Alfresco Community 5.2?

Se proponen las siguientes **tareas de investigación**:

1. Revisión de la bibliografía para elaborar el marco teórico conceptual en lo referente al desarrollo del módulo reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.
2. Análisis de las herramientas informáticas y metodologías de desarrollo de software asumidas por el proyecto para realizar la implementación del módulo reportes sobre la tecnología del Share de Alfresco Community 5.2.
3. Identificación de las principales funcionalidades del sistema para la posterior implementación del mismo, teniendo en cuenta los requisitos definidos.
4. Identificación de los diferentes tipos de pruebas de software, para su posterior aplicación sobre el sistema desarrollado.

Para dar cumplimiento a las tareas se utilizarán los siguientes **métodos científicos**:

1. **Analítico-Sintético**: este método sirvió para la recopilación de información requerida durante la realización del estudio del estado del arte y para el desarrollo del trabajo mediante la revisión de documentos y artículos, de donde se extrajeron los elementos más significativos relacionados con los módulos de reportes, además del análisis de las diferentes herramientas, metodologías y tecnologías a utilizar en el desarrollo del módulo.
2. **Modelación**: fue utilizado en la representación, mediante el uso de diagramas, de las características del sistema a desarrollar, relaciones entre objetos; y las actividades que intervinieron en el proceso de configuración del entorno colaborativo.

Como **métodos empíricos** se utilizarán:

- **Entrevista**: este método de investigación fue utilizado para obtener información de forma directa con el cliente para determinar las herramientas que se utilizarían y levantamiento de requisitos

funcionales y no funcionales que presenta el sistema.

El desarrollo está estructurado en tres capítulos que contienen la información siguiente:

Capítulo 1: Fundamentación teórica: Se establecen los principales aspectos teóricos para el objeto de estudio de la investigación. Se realiza el estudio de los módulos reportes existentes. Estudio de los sistemas como Alfresco Community 5.2. Se seleccionan herramientas y tecnologías que se emplean en el desarrollo del módulo reportes para Share Alfresco Community y se caracteriza la metodología para seleccionar aquella que se adecue mejor a las características del producto.

Capítulo 2: Análisis y diseño: Se describe la propuesta de solución. Se diseña el modelo de domino. Se definen los requisitos funcionales y no funcionales. Se diseñan y elaboran diagramas y especificaciones de casos de uso. Se explican y ejemplifican los patrones arquitectónicos y de diseños utilizados.

Capítulo 3: Implementación y pruebas: Se elaboran diagramas de componentes y de despliegue. Se enuncian los estándares de codificación. Se define la estrategia de pruebas de software.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El presente capítulo contiene los resultados del estudio de los módulos generadores de reportes; sus características, así como su importancia en los sistemas de información. Se hace un análisis del módulo de reporte de la herramienta de generación de reportes basados en Apache Solr de Repxos 3.0. También se describen las herramientas y tecnologías usadas para el desarrollo del módulo, así como la metodología de desarrollo y el lenguaje de modelado a utilizar.

1.2. Conceptos fundamentales

1.2.1. Repositorios

Los repositorios constituyen sistemas de información que tienen como finalidad organizar, preservar y difundir en el modo acceso abierto (Open Access) recursos científicos y académicos de las instituciones (Duperet Cabrera, Pérez Martínez et al. 2015).

Son sistemas de información que preservan y organizan materiales científicos y académicos como apoyo a la investigación y el aprendizaje; y garantizan el acceso a la información. Se considera un sistema de gestión de contenidos, que administra la producción científica en formato digital. Utilizan estándares abiertos para garantizar que sus contenidos sean accesibles y puedan ser buscados y recuperados para su uso posterior. Son un medio de publicación científica. Ofrece otros servicios complementarios. Un repositorio permite importar, identificar, almacenar, preservar, recuperar y exportar un conjunto de objetos digitales, desde un portal web.

De forma general, es un sistema de almacenamiento digital donde todos los archivos almacenados son públicos y pueden ser accedidos por cualquier usuario. Las características más notables son sus diferentes formatos de archivo, escalabilidad, extensibilidad; además cuenta con estándares de metadatos, descriptivos, de conservación y administrativos. Dos de sus grandes ventajas es la de búsqueda y visualización de metadatos.

Tipos de repositorio:

Existen varios tipos de repositorios clasificados atendiendo a sus prestaciones y datos que almacenan. A continuación se abordan conceptos de varios tipos de repositorios:

Repositorios temáticos: Los repositorios temáticos se definen como un conjunto de objetos digitales resultantes de la investigación relacionada con un campo disciplinario específico (Soares Guimarães, da Silva et al. 2012).

Repositorios de datos: repositorios que almacenan, conservan y comparten los datos de las investigaciones (Hernández Pérez and García Moreno 2013).

Repositorios institucionales: desarrollado por organismos políticos, sociales y educativos como universidades e institutos o asociaciones, para depositar, usar y preservar la producción científica y académica que generan en formato digital y haciéndola accesible al público. De esta manera la institución ofrece un servicio acorde al movimiento de acceso abierto (Barrueco and García Testal 2009).

Este último tiene como objetivo facilitar el acceso de la comunidad científica internacional a los resultados de la investigación realizada por sus miembros y aumentar la visibilidad de la producción científica de la institución. Así mismo, contribuir a la preservación de los documentos digitales allí depositados. Reunir y difundir al mundo la producción científica y académica de la institución.

Plataformas para sistemas de repositorios:

Existe una amplia comunidad científica involucrada en el desarrollo de plataformas tecnológicas para la creación de repositorios open access, la utilización de una o de otra responderá a las necesidades y recursos de la institución que lo ponga en marcha. Existe una revisión bastante exhaustiva sobre software libre para repositorios publicada por el Open Society Institute (OSI) en la cual se analizan y comparan las más utilizadas como son DSpace, Eprints, Fedora, CDSware, entre otras.

Todas estas plataformas tienen en común que cumplen con el protocolo OAI-PMH (Open Archive Initiative-Protocol for Metadata Harvesting) el cual tiene como misión desarrollar y promover estándares de

interoperabilidad. De las estadísticas sacadas del directorio OpenDOAR, las dos con mayor número de instalaciones son Eprints y Dspace (Figura 1).

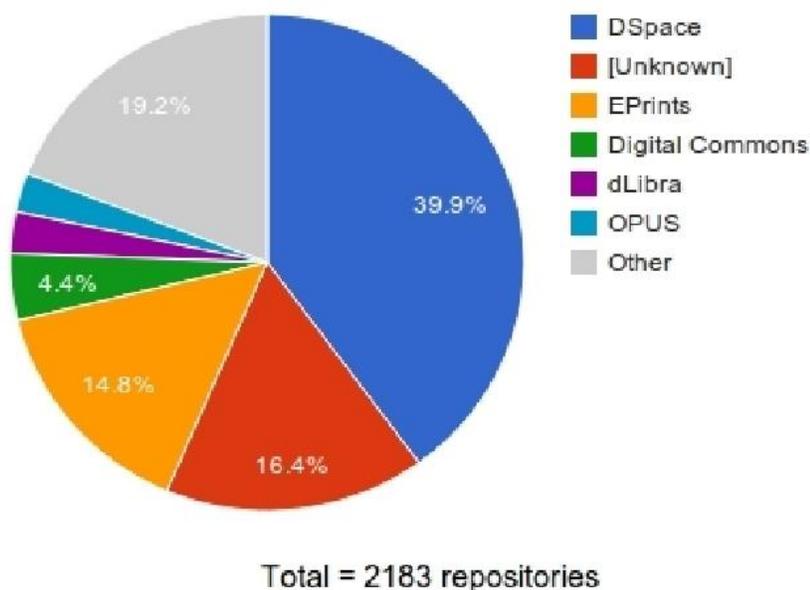


Figura 1 Distribución del tipo de software empleado para la creación de repositorios en OpenDOAR (SEDIC)

Entre las más utilizadas se encuentra DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales y comúnmente es usada como solución de repositorio institucional. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, videos, datos de investigación y otras formas de contenido. DSpace cuenta con algunas características de gran importancia que permiten y facilitan su uso en una gran cantidad de instituciones. Entre ellas se encuentra su distribución bajo Licencia Berkeley Software Distribution (BSD, por sus siglas en inglés); que es multiplataforma, por lo que se adapta a gran número de Sistemas Operativos; es basado en tecnología web y es un sistema adaptable.

DSpace a contando con múltiples versiones desde su lanzamiento incluyendo en cada un notable cambio en sus prestaciones. Antes de 2012, las versiones se numeraban como 1.x, 1.6, 1.7 y 1.8. Cada año aproximadamente se lograba terminar una versión nueva y se liberaba. En ese momento, 2008 -2010, se tenía en mente la versión “definitiva”, DSpace 2, pero esa versión 2 nunca llegó, así que en 2013 se decidió comenzar un nuevo esquema de numeración de versiones en forma [major].[minor] (incrementar el primer

número, [mayor], significa que se está ante una versión principal de DSpace y las versiones menores [menor] son las que incrementan el segundo número que sólo incluyen parches y resolución de bugs (bugs fixes) de la versión principal.). DSpace desde entonces se numera como 3.0 (la primera tras el cambio de criterio) 3.1, 3.2.... 4.0, 4.1. Nunca hubo una versión 2 de DSpace, siendo la versión 3.0 su primera oficialmente lanzada.

DSpace 3.0

DSpace en su versión 3.0 fue lanzado oficialmente al público el 30 de noviembre de 2012 y cuenta con una serie de características relativamente significativas. Ciertas funciones se habilitan automáticamente de forma predeterminada, mientras que otras requieren una activación deliberada. La siguiente lista contiene las fundamentales funciones nuevas en 3.0 que están habilitadas de forma predeterminada:

- 1) Interfaz OAI-PMH completamente reescrita:
 - Controlador y Open-AIRE compatibles.
 - 2 formatos de exportación de metadatos predeterminados y una forma fácil de escribir nuevos usando XSLT.
 - Permite múltiples contextos (puntos finales de URL), cada uno con una configuración diferente.
- 2) Mejoras a las estadísticas basadas en Solr:
 - Estadísticas de flujo de trabajo.
 - Estadísticas de consultas de búsqueda.
 - Actualización de la versión Solr y optimización del rendimiento.
- 3) Soporte de Google Analytics para JSPUI:
 - Soporte para la recopilación de estadísticas al ingresar la clave GA en dspace.cfg.

La siguiente lista contiene ejemplos de las funciones que se incluyen en la versión DSpace 3.0, pero deben habilitarse manualmente:

- 1) Versiones a nivel de artículo:
 - Crea y conserva diferentes versiones de artículos.
 - Identificadores mejorados.
 - Solo XMLUI.
- 2) Estadísticas de uso basadas en ElasticSearch:

- Backend de Elasticsearch escalable, se ejecuta en un nodo incorporado de forma predeterminada.
 - La visualización puede ser pública o restringida.
- 3) Mejoras en la autenticación LDAP:
- Autenticación LDAPHierarchical sustituida por la autenticación LDAPA.
 - Nueva opción para asignar la pertenencia a un grupo LDAP a grupos internos de DSpace.
- 4) Descubrimiento: Búsqueda y navegación:
- Mejoras para XMLUI (Fragmentos de búsqueda, hit resaltado, oculta resultados restringidos, artículos relacionados, etc.).

DSpace 6.0

DSpace en su versión 6.0 fue lanzado oficialmente al público el 24 de octubre de 2016 y cuenta con una serie de características muy significativamente superiores a sus versiones precedentes (3.x, 4.x y 5.x); y al igual que sus anteriores versionados cuenta con funciones de activación automáticas y manuales. La siguiente lista contiene las fundamentales funciones actualizadas o mejoradas que 6.0 ofrece con respecto a su versión anterior 3.0:

- 1) La propia API de Java ahora se divide en tres "capas" que ahora se inicializan a través de Spring:
- Capa de servicio, que contiene la lógica de negocio DSpace.
 - Capa del objeto de acceso a la base de datos (DAO), que contiene consultas específicas de la base de datos para localizar o encontrar objetos individuales.
 - Capa de objeto de base de datos, que proporciona acceso a columnas de base de datos específicas y propiedades de objetos.
- 2) Sistema de configuración mejorado (recargable):
- Ofrece una gestión más sencilla de las configuraciones locales a través de un nuevo archivo local.cfg. Cualquier configuración (de los archivos .cfg) se puede anular en DSpace simplemente copiándolos en su local.cfg y cambiando el valor.
 - Muchas configuraciones ahora se vuelven a cargar automáticamente tan pronto como guarda su archivo local.cfg. Esto significa que no necesita reiniciar Tomcat cada vez que necesite cambiar una configuración.

- 3) Complementos de almacenamiento de archivos (flujo de bits) mejorado, incluida la compatibilidad con el almacenamiento de archivos de Amazon S3.
- 4) Informes configurables del estado del sitio (es decir, estado del repositorio) a través de correo electrónico.
- 5) Nuevas funcionalidades de XMLUI:
 - Nuevo marco para la importación de metadatos desde fuentes externas:
 - ✓ Marco para admitir la importación de metadatos de fuentes externas: DS-2876.
 - ✓ Plugin de importación de metadatos PubMed: DS-2880.
 - Este concepto anteriormente solo era compatible con JSPUI.
 - Panel de control administrativo extensible, que permite a las instituciones anular o personalizar las pestañas en el "panel de control" administrativo de XMLUI (DS-2625).
 - Exportación de resultados de búsqueda XMLUI a CSV para la edición de metadatos por lotes. Anteriormente, esta característica solo era compatible con JSPUI.
- 6) Nuevas características de la API REST:
 - Informes de control de calidad basados en REST.
 - Soporte para todos los métodos de autenticación a través de REST (por ejemplo, Shibboleth, LDAP, etc.).
 - La API REST ahora proporciona una versión a través del punto final de estado (hereda la versión de la API de DSpace).
- 7) Mejoras de búsqueda y descubrimiento:
 - Todas las búsquedas ahora por defecto son booleanas AND.
 - La nueva faceta "Tiene archivo", que le permite filtrar fácilmente por elementos que tienen uno o más archivos.
 - Indexación de texto completo de hojas de cálculo de Excel, de modo que ahora se pueden buscar.
 - El texto de derecha a izquierda en archivos PDF ahora está indexado para la búsqueda.
- 8) Otras mejoras:
 - OAI-PMH se actualizó para cumplir con las pautas de OpenAIRE 3.0 para repositorios de publicaciones.
- 9) Características eliminadas o reemplazadas en 6.0:

- El archivo de configuración build.properties ha sido reemplazado por un archivo de configuración local.cfg mejorado. El nuevo local.cfg le permite anular fácilmente cualquier configuración (desde dspace.cfg o modules / *. Cfg files) simplemente copiándolo en su local.cfg y especificando un nuevo valor. También proporciona opciones de configuración mejoradas como se detalla en la documentación de referencia de configuración. El antiguo archivo build.properties ya no se usa ni se admite. Como parte de la adición de este nuevo esquema de configuración, muchos de los ajustes de configuración en DSpace (principalmente aquellos en los módulos / *. Cfg files) tuvieron que ser renombrados o añadidos con el nombre del módulo. Esto significa que ya no se garantiza que las configuraciones 5.x (o inferiores) sean compatibles con 6.x.
- Se ha cambiado el nombre del archivo de configuración de la página de portada de citas de PDF.
- Se han eliminado el motor de búsqueda heredado (basado en Apache Lucene) y el sistema de navegación heredado (basado en tablas de base de datos) desde DSpace 6.0 o superior. En su lugar, DSpace ahora solo usa Discovery (basado en Apache Solr) para todas las capacidades de búsqueda / navegación.
- La Interfaz de red ligera DSpace (LNI), compatible con la API WebDAV / SOAP / RPC, se ha eliminado de DSpace 6.0 o superior. En su lugar se utilizará REST o SWORD (v1 o v2) como reemplazo.
- La compatibilidad con el almacenamiento de archivos SRB (Storage Resource Broker) se ha eliminado de DSpace 6.0 o superior. Como no se mantuvo (y aparentemente no se usó) durante muchos años, esta característica se eliminó junto con sus configuraciones. Como reemplazo, se agregó un nuevo sistema de complemento de almacenamiento de archivos, que incluye una opción de almacenamiento de archivos local tradicional (predeterminado) y una opción de almacenamiento de archivos de Amazon S3.
- Los grupos de usuarios Administrador y Anónimo no pueden ser renombrados o eliminados. Si los ha renombrado, se les cambiará el nombre a los nombres de inventario durante la actualización. DSpace ahora depende de estos nombres específicos debido a cambios internos.

- La función de estadísticas de uso de ElasticSearch está en desuso en la versión 6.0. Si bien aún funcionan y puede continuar usándolos, en una futura versión de DSpace se eliminarán. Se está buscando una opción de estadísticas de uso, y la más recomendada a utilizar es el motor de estadísticas SOLR predeterminado y / o la integración de DSpace con Google Analytics.

Conclusiones de la comparación entre versiones 3.0 y 6.0 de DSpace

La versión 6.0 de DSpace ofrece notables ventajas, mejoras y correcciones sobre sus anteriores versiones, incluyendo por supuesto 3.0. Entre las más notables e influyentes para el desarrollo de la propuesta de solución planteada se encuentran, entre otros, la redefinición de la numeración de los tipos de objetos (ObjectTypes) a obtener en las consultas siguiendo la siguiente asignación: Bitstream = 0, Item = 2, Collection = 3, Community = 4, Site = 5, EPerson = 7; una mejor compatibilidad con el servicio API incluyendo como sustituto de WebDAV a REST, mejor definición de roles esta vez no editables con en algunas versiones anteriores, todas las búsquedas serán por defecto booleanas y utilizan operadores como AND para múltiples parámetros de consultas, remplazo de motores de búsqueda ahora utilizando los basados en Apache Solr, entre otras que dejan clara la superioridad de la versión 6.0 y las ventajas que esta ofrece en el desarrollo del módulo.

1.2.2. Reportes

Un reporte es una noticia o informe que brinda información con algún propósito. En el ámbito de la informática es un informe que organiza y exhibe la información contenida en una base de datos. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar por los usuarios. A través de los reportes se refleja el comportamiento de los diferentes componentes que integran cualquier sistema de información de una empresa o institución, permitiendo el reconocimiento, control y monitoreo de las diversas problemáticas existentes en la misma. De esta forma se apoya en la correcta toma de decisiones (Definicion.de, 2014). Los reportes en la informática como informes de cierta actividad o proceso, posibilitan presentar las estadísticas de cualquier ámbito o entorno. La manera fácil para descifrar o analizar problemas descritos en diversos medios permite indagar en respuestas y soluciones.

Reportes dinámicos

En la actualidad la forma de generar los diferentes tipos de reportes se restringe a las facilidades que presten las herramientas propias de las bases de datos o herramientas creadas con este fin. Se presenta como problema, que el soporte para la adaptación de los reportes a las necesidades de los usuarios sea solo por personal técnico capacitado, lo que implica gastos a las empresas en tiempo y costos (PRÓMETA, 2009). Además, en toda organización existe demasiada información, la cual necesita ser procesada y manipulada para que el personal de toma de decisiones pueda abstraer los datos que necesitan en el momento que lo requieran. Esta es la causa por la que necesitarían un reportador eficiente y fácil de manejar, sin necesidad de un mayor conocimiento técnico, es decir un reportador dinámico (Definicion.de, 2014).

En resumen, un reportador dinámico es una herramienta informática, que basada en las estadísticas que desea el usuario, es capaz de transformar o cambiar la estructura del reporte o la consulta de los datos a obtener según las necesidades que tiene el usuario.

Sistemas Generadores de Reportes

En todo sistema de información es fundamental contar con una herramienta adicional cuyo objetivo sea la de generar reportes. Los sistemas generadores de reportes tienen en las bases de datos su principal fuente de alimentación, pues a partir de la información almacenada en la misma se realizan consultas para obtener información en forma de reporte (PRÓMETA, 2009).

A diferencia de las consultas tradicionales, con los generadores de reportes se puede definir el diseño y la forma en que la información será visualizada. Este se compone por un diseñador de reportes y por un motor de generación de reportes. El primero se encarga de brindar las herramientas para diseñar la apariencia del informe. El segundo accede a la fuente de datos, obtiene los necesarios y los introduce en el diseño de la plantilla del reporte con la que luego se puede realizar ciertas operaciones como: imprimir, enviar por correo electrónico o guardar en un archivo. Los sistemas de generación de reportes incluyen además varias funcionalidades que permiten la introducción de gráficos en los reportes y la definición del formato final en el que son generados (PRÓMETA, 2009).

En la Universidad de las Ciencias Informáticas en el centro CIGED existe una herramienta de generación de reportes, la cual se analizó para desarrollar la propuesta de solución:

Herramienta de generación de reportes basados en Apache Solr de Repxos 3.0

Esta es una herramienta informática creada en el 2015 como solución para el sistema de repositorio XABAL Repxos 3.0. Este sistema está desarrollado sobre la plataforma DSpace 3.0 que utiliza un servidor Solr para indexar los logs de uso de la aplicación, permitiendo el acceso a sus estadísticas mediante consultas Solr.

Características principales:

- Se parametrizan las diferentes consultas sobre la cual se van a basar los reportes, la gestión, diseño y configuración de los reportes.
- Permite la exportación a formato físico PDF.
- Permite la obtención de reportes del repositorio.

1.3. Metodología

El nuevo módulo de reportes a desarrollar esta pensado sobre un conjunto de tecnologías, metodologías, herramientas y lenguajes que se analizaran e investigaran para dar solución a los objetivos propuestos. A continuación se explican las razones de selección de varios de ellos.

1.3.1. Metodología de desarrollo de software

Las metodologías de ingeniería de software pueden considerarse como una base necesaria para la ejecución de cualquier proyecto de desarrollo de software que se considere serio, y que necesite sustentarse en algo más que la experiencia y capacidades de sus programadores y equipo.

Estas metodologías son necesarias para poder realizar un proyecto profesional, tanto para poder desarrollar efectiva y eficientemente el software, como para que sirvan de documentación y se puedan rendir cuentas de los resultados obtenidos. (Maida & Pacienza, 2015).

AUP

El Proceso Unificado Ágil de Scott Ambler o Agile Unified Process (AUP) en inglés es una versión simplificada del Proceso Unificado de Rational (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo Desarrollo Dirigido por Pruebas (test driven development – TDD), Modelado Ágil, Gestión de Cambios Ágil, y Refactorización de Base de Datos para mejorar la productividad.

AUP-UCI

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) es una versión simplificada del Proceso Unificado de *Rational* (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles conceptos que aún se mantienen válidos en RUP. AUP se preocupa especialmente de la gestión de riesgos. Propone que aquellos elementos con alto riesgo obtengan prioridad en el proceso de desarrollo y sean abordados en etapas tempranas del mismo.

En la Universidad de la Ciencias Informáticas (UCI) se desarrolló una versión de esta metodología ágil con el objetivo de crear un estándar en el proceso productivo de manera que se adapte al ciclo de vida de los proyectos en la UCI. En esta variación de AUP para la UCI se mantiene la primera fase de inicio, la segunda fase de ejecución en esta variación para la UCI resume las fases restantes de elaboración, construcción y transición. Finalmente se encuentra la fase de cierre donde se analiza los resultados del proyecto y se realizan las actividades formales de cierre de proyecto (Sanchez, 2015).

AUP-UCI propone para el ciclo de vida de los proyectos en la UCI tener siete disciplinas, los flujos de trabajo son los siguientes: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Además, está compuesta por cuatro escenarios:

El escenario 1 propone a los proyectos que modelan el negocio con casos de usos del negocio que solo pueden modelar el sistema con casos de usos del sistema. El escenario 2 propone modelar el negocio con un modelo conceptual y el sistema con casos de uso del sistema. El escenario 3 propone modelar el negocio con descripción de proceso de negocio, junto al modelo conceptual y el sistema mediante la descripción de requisitos por proceso. El escenario 4 propone no modelar el negocio y describir el sistema mediante historias de usuario.

Se selecciona para esta investigación el escenario número 2 ya que este aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan que no es necesario incluir las responsabilidades de las personas que ejecutan las actividades, de esta forma modelarían exclusivamente los conceptos fundamentales del negocio. Se recomienda este escenario para proyectos donde el objetivo primario es la gestión y presentación de información.

1.4. Lenguajes de programación, tecnologías y herramientas informáticas.

1.4.1. Lenguajes de programación

Los lenguajes de programación son las herramientas que permiten la interacción entre el ser humano y las máquinas. Existen diversos lenguajes de programación y se pueden clasificar como, lenguajes de programación del lado de servidor y lenguajes de programación del lado del cliente.

CSS v3.0

 El lenguaje CSS permite presentar, de manera estructurada, un documento que fue escrito en un lenguaje de marcado. Se usa especialmente en el diseño visual de un sitio web cuando las páginas se hallan escritas en XML o HTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas (GeoDir, 2017).

Características

- Atributo gradiente de colores en borde: posibilidad de definir un gradiente de color en el borde de los elementos, en un atributo no estándar de Firefox.
- Bordos redondeados: define la curvatura del borde del elemento con el atributo border-radius.
- Crea sombras con el atributo box-shadow, aplica sombras a los elementos de la página, sin usar imágenes y javascript de forma extra.

HTML5

 El Lenguaje de Marcado de Hipertexto HTML (HyperText Markup Language, por sus siglas en inglés) es una colección de estándares para el diseño y desarrollo de páginas web. Esta colección representa la manera en que se presenta la información en el explorador de internet y la manera de interactuar con ella. HTML5 permite una mayor interacción entre las páginas web y contenido media, así como una mayor facilidad a la hora de codificar el diseño básico (GeoDir, 2017).

UML

 UML (Unified Modeling Language) o Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos y describe la semántica esencial de estos diagramas y símbolos en ellos utilizados (Lamarca, 2013).

Este lenguaje resultó de gran utilidad durante el desarrollo del entorno colaborativo, pues con él se realizó el diseño de los diagramas necesarios.

XML



XML es un lenguaje de marcas para los documentos que contienen información estructurada. La información estructurada contiene tanto el contenido (palabras, imágenes, entre otras) y alguna indicación de lo que desempeña el papel de contenido (por ejemplo, contenido en un encabezado de sección tiene un significado diferente del contenido de una nota al pie, que significa algo diferente de lo contenido en una figura leyenda o el contenido de una tabla de base de datos, entre otras). Casi todos los documentos tienen cierta estructura. Un lenguaje de marcado es un mecanismo para identificar las estructuras de un documento. La especificación XML define una forma estándar de añadir marcadores para documentos (O'Reilly Media, 2016).

Entre sus principales características se encuentran:

- Extensibilidad.
- Estructura.
- Validación.
- Basado en texto.
- Orientado a los contenidos no a la presentación.
- Las etiquetas se definen para crear los documentos, no tienen un significado preestablecido.

Para la implementación de este módulo se hace necesaria la utilización de este lenguaje en la creación de los modelos de contenido y los servicios de datos además de realizar todos los ficheros de configuración y para desplegar los procesos en Alfresco Community.

JavaScript



Es un lenguaje de programación interpretado, o sea, no requiere compilación. Es utilizado especialmente en páginas web embebido en el código HTML o similares. La mayoría de los navegadores pueden interpretar los códigos JavaScript incluidos en las páginas web (Alegsa L. D., 2016).

Por existencia de distintas versiones de JavaScript incompatibles, el World Wide Web Consortium (W3C) diseñó un estándar llamado DOM (Document Object Model) que incorpora Internet Explorer 6 en adelante, Opera versión 7 en adelante y Mozilla con el objetivo de lograr mejor compatibilidad entre navegador y página web.

Con JavaScript se pueden extender las posibilidades de las páginas web como, por ejemplo, evitar que se pueda copiar el texto de una página, botones para agregar automáticamente una página a favoritos, crear

barras de scroll, abrir popups, cambiar el puntero del mouse, rotar banners, validar formularios, entre otras. En la solución se pone de manifiesto este lenguaje en la implementación de las páginas web que serán vistas por el cliente, además de los formularios y las tablas.

Características:

- Tipado dinámico: como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable.
- Objetual: está formado casi en su totalidad por objetos.
- Evaluación en tiempo de ejecución: incluye la función “eval” que permite evaluar expresiones como cadenas en tiempo de ejecución.
- Prototípico: usa prototipos en vez de clases para el uso de herencia.

1.4.2. Herramientas informáticas

Las herramientas de desarrollo de software (HIS) han desempeñado desde sus inicios un importante papel en el desarrollo de aplicaciones. Como parte de la ingeniería de software (IS), las HIS han experimentado también continuos cambios, consecuencia del creciente avance tecnológico propio de los últimos años. Este cambio ha sido un factor con especial influencia sobre la IS, así como también sobre otras disciplinas relacionadas, e impulsa una alta tasa de cambio en las HIS. Actualmente éstas son numerosas y apoyan en múltiples formas diferentes dimensiones del desarrollo de software en general (Lornel Rivas, 2014).

Entorno de Desarrollo Integrado (IDE por sus siglas en Ingles): Visual Studio Code



Visual Studio Code es una versión reducida del entorno de desarrollo oficial de Microsoft centrado exclusivamente en el editor de código. Es multiplataforma y soporta la sintaxis de una gran cantidad de lenguajes de programación. La herramienta proporciona soporte y asistencia a lenguajes de diversos ámbitos como HTML, CSS, JavaScript, diversas variantes de C, Java, SQL, PHP, Ruby, Visual Basic o JSON entre muchos otros, soportando resaltado, sangriado automático, snippets y autocompletado.

Sistema Gestor de Base de Datos: PostgreSQL V 9.4



PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Esta herramienta utiliza un modelo

cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Sus características técnicas la hacen una de las bases de datos más robustas y potentes del mercado. La estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema.

Características principales

- Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de bits, etc. También permite la creación de tipos propios.
- Incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Multiplataforma: está disponible en casi cualquier Unix (34 plataformas en la última versión estable).

Contenedor de aplicaciones web: Apache Tomcat V 7.0



Tomcat es un servidor web. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era solo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

Herramienta CASE para el modelado: Visual Paradigm V 8.0



El Visual Paradigm para UML es una herramienta para desarrollo de aplicaciones utilizando modelado UML ideal para Ingenieros de Software, Analistas de Sistemas y Arquitectos de Sistemas que están interesados en construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos.

Visual Paradigm for UML Enterprise Edition (VP-UML EE): Es la edición top de la línea de productos, lo que representa lo más moderno y agrega valor en términos de modelado de datos orientado a objetos, hace

posible la documentación del proyecto, mapeo relacional de objetos para Java, .NET y PHP, reduciendo costos y aumentando su productividad (Paradigm, 2013).

Posibilita la representación gráfica de los diagramas permitiendo ver el sistema desde diferentes perspectivas como: componentes, despliegue, secuencia casos de uso; clase, actividad, estado, entre otros. Además, identifica requisitos y comunica información, se centra en cómo los componentes del sistema interactúan entre ellos, sin entrar en detalles excesivos, además permite ver las relaciones entre los componentes del diseño y mejora la comunicación entre los miembros del equipo usando un lenguaje gráfico. Facilita licencias especiales para fines académicos.

Herramienta para desarrollo de prototipos de interfaces: Axure RP Pro V 7.0



Axure RP Pro es una herramienta rápida de creación de diagramas, wireframes, prototipos y especificaciones para Websites. Axure RP se distingue principalmente por su facilidad de uso y riqueza de su diseño. Permite desarrollar una interfaz final del producto y exportar en resultado permitiendo su visualización en forma de sitio web o incluso como imagen. El software está disponible en el mercado en dos versiones: Standard y Axure RP Pro (Scientec, 2016).

Software proveedor de herramientas para colecciones digitales: DSpace V 6.0



DSpace es un software de código abierto que provee herramientas para la administración de colecciones digitales, y comúnmente es usada como solución de repositorio bibliográfico institucional. Soporta una gran variedad de datos, incluyendo libros, tesis, fotografías, filmes, video, datos de investigación y otras formas de contenido. Los datos son organizados como ítems que pertenecen a una colección; cada colección pertenece a una comunidad. DSpace está escrito en Java. Usa una base de datos relacional, y soporta el uso de PostgreSQL y Oracle. Tiene dos interfaces, una clásica (JSPUI) que usa JSP y Java Servlet API, y una nueva (XMLUI) basada en Apache Cocoon que usa XML y XSLT. DSpace es totalmente compatible con el protocolo OAI-PMH, y es capaz de exportar paquetes de software METS (Metadata Encoding and Transmission Standard).

Software para la gestión documental: Alfresco Community 5.2



Alfresco es un sistema de administración de contenidos que le permite al usuario la gestión de contenido documental, tareas y recursos. Alfresco permite almacenar, compartir, administrar

contenidos y documentos de la organización (Bermúdez and Fardy 2014).

Se distribuye en tres variantes diferentes:

- Alfresco Community Edition: Es software libre, con licencia LGPL de código abierto y estándares abiertos.
- Alfresco Enterprise Edition: Se distribuye bajo licencia de código abierto y estándares abiertos, con la posibilidad de soporte comercial y propietario a escala empresarial.
- Alfresco Cloud Edition (Alfresco in the cloud) que es la versión SaaS o Software como Servicio de Alfresco.

Alfresco incluye un repositorio de contenidos, un framework de portal web para administrar y usar contenido estándar en portales, una interfaz CIFS (el antiguo SMB) que provee compatibilidad de sistemas de archivos en Windows y sistemas operativos tipo Unix, un sistema de administración de contenido web, capacidad de virtualizar aplicaciones web y sitios estáticos vía Apache Tomcat, búsquedas vía el motor Apache Solr-Lucene y flujo de trabajo en jBPM.

Características

- Gestión de documentos.
- Gestión de contenido web (incluyendo aplicaciones web y virtualización de sesiones).
- Versionado a nivel de repositorio.
- Gestión de registros.
- Gestión de imágenes.
- XForms autogenerados con soporte AJAX.
- Publicación integrada.
- Acceso al repositorio vía CIFS/SMB, FTP y WebDAV.
- Flujo de trabajo basado en BPM Activiti.
- Búsquedas implementadas con el motor Lucene.
- Servidores descentralizados.
- Soporte de varios idiomas.
- Empaquetamiento de aplicación portable.
- Soporte multiplataforma (oficialmente Windows, GNU/Linux y Solaris).

- Interfaz gráfica basada en navegadores de Internet (oficialmente Internet Explorer y Mozilla Firefox).
- Soporte de clustering (despliegue en varios servidores).

1.4.3. Tecnologías

JQuery V 1.10.3



Es un framework Javascript que ofrece una infraestructura para la creación de aplicaciones complejas del lado del cliente. Con jQuery se obtiene ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax. JQuery permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM (Document Object Model, por sus siglas en inglés), manejar eventos y desarrollar animaciones (Alvarez 2010).

Características

- Selección de elementos DOM.
- Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3.
- Permite la realización de eventos.
- Permite manipulación de la hoja de estilos CSS.
- Permite la realización de efectos y animaciones.
- Proporciona animaciones personalizadas.
- Utilidades varias como obtener información del navegador, operar con objetos y funciones para rutinas comunes.

JQuery permitió ahorrar tiempo y recursos en el desarrollo del software. Posee complementos con funcionalidades de uso común en las vistas de los sistemas web, lo hacen un framework flexible y rápido para el desarrollo web, dando la posibilidad de que los errores sean resueltos rápidamente apoyándose en una comunidad de soporte.

Bootstrap v2.3.2



Es un framework desarrollado por Twitter que simplifica el proceso de creación de diseños web combinando CSS y JavaScript. La mayor ventaja es que se puede crear interfaces que se adapten a los distintos navegadores con la propiedad denominada diseño adaptativo. Los diseños creados con

Bootstrap son simples, limpios e intuitivos, dándole agilidad a la hora de cargar y adaptarse a otros dispositivos (Solis 2014).

Características

- Se integra perfectamente con las principales librerías JavaScript, por ejemplo JQuery.
- Ofrece un diseño sólido usando LESS y estándares como CSS3 y HTML5.
- Es un framework ligero que se integra de forma limpia en el proyecto actual.
- Dispone de distintos diseños predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.
- Todas las clases CSS están contenidos en un solo archivo y permite una mejor personalización mediante mejoras, herencias y especificidad.

Bootstrap en su versión 2.3.2 permitió agilizar el diseño de las vistas de la herramienta. Esto se debe a la utilización de componentes y servicios creados por la comunidad web, tales como: HTML5, CSS3, jQuery UI, LESS entre otros y considerada como una herramienta sencilla y ágil en el desarrollo de interfaces.

Conclusiones del Capítulo

El estudio de los conceptos asociados a repositorios y reportes permitió conocer sus principales características y beneficios. La caracterización de sistemas generadores de reportes dinámicos en la actualidad, proporcionó un conjunto de conocimientos acerca de sus rasgos y propiedades fundamentales. La descripción de las herramientas, lenguajes y tecnologías definidas permitió la familiarización de los elementos del ambiente de desarrollo, además de adquirir los conocimientos esenciales para poder utilizarlos en el desarrollo de la propuesta de solución.

CAPÍTULO 2. ANÁLISIS Y DISEÑO

2.1. Introducción

En el presente capítulo se describe el módulo de reportes y se realiza el modelo de dominio definiendo los elementos manejados en el sistema junto a las relaciones establecidas entre estos. Se especifican los requerimientos funcionales y no funcionales del módulo. Se representa el diagrama de casos de uso del sistema unido a las descripciones de los casos de uso del mismo.

2.2. Descripción de la solución propuesta

La propuesta de solución es un módulo de reportes desarrollado para la tecnología del Share de Alfresco Community en su versión 5.2, el cual permite una completa gestión de servidores solr, consultas solr y reportes. Para ello dicha herramienta cuenta con tres áreas de trabajo (gestión de servidores solr, gestión de consultas solr y gestión de reportes) las cuales permiten la creación, visualización, modificación, eliminación y búsqueda de cada uno de los componentes del módulo (servidores, consultas y reportes). Permite además la exportación de reportes en documentos con formatos .pdf y cuenta con una interfaz sencilla de usar formada por un menú de navegación con accesos a los espacios de trabajo.

2.3. Modelo de dominio

El modelo de dominio tiene como objetivo comprender y describir solamente las clases más importantes dentro del contexto en el cual se desempeña el software, con el propósito de sentar las bases del entendimiento del desarrollo y no para definirlo completamente (LARMAN, 2003).

El sistema no tiene los procesos de negocio bien definidos, lo que ha propiciado un modelado del dominio en lugar del negocio. El objetivo del modelado del dominio es capturar, comprender y describir las clases más importantes dentro del contexto del sistema. El modelo del dominio, junto al diccionario de clases del dominio, ayuda a los usuarios, clientes, desarrolladores y otros interesados a utilizar un vocabulario común. En la figura 2 se muestra el diagrama de clases del dominio. Por su parte, el diccionario de clases del dominio describe textualmente las clases identificadas durante el modelado del dominio del problema. Este diccionario se muestra a continuación:

- **Administrador:** Hace referencia a la persona que posee el rol de administrador, este tiene todos los permisos para realizar cualquier cambio dentro del módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.
- **Consulta solr:** representa la consulta que se realiza al servidor solr para realizar la petición de los datos en la generación del reporte.
- **Reporte:** tipo de contenido que representa un informe que organiza y exhibe la información necesitada por el usuario. Su función es aplicar un formato determinado a los datos para mostrarlos por medio de un diseño atractivo y que sea fácil de interpretar.
- **Servidor solr:** hace referencia a todos los servidores a los cuales se conecta el sistema Herramienta informática para la gestión de reportes dinámicos basado en consulta solr, poseen un conjunto de datos necesarios para representarlos.
- **Usuario:** representa a todas las personas autenticadas en el sistema y que acceden al sistema con el fin de crear o acceder a ver los reportes.

Modelo de Dominio

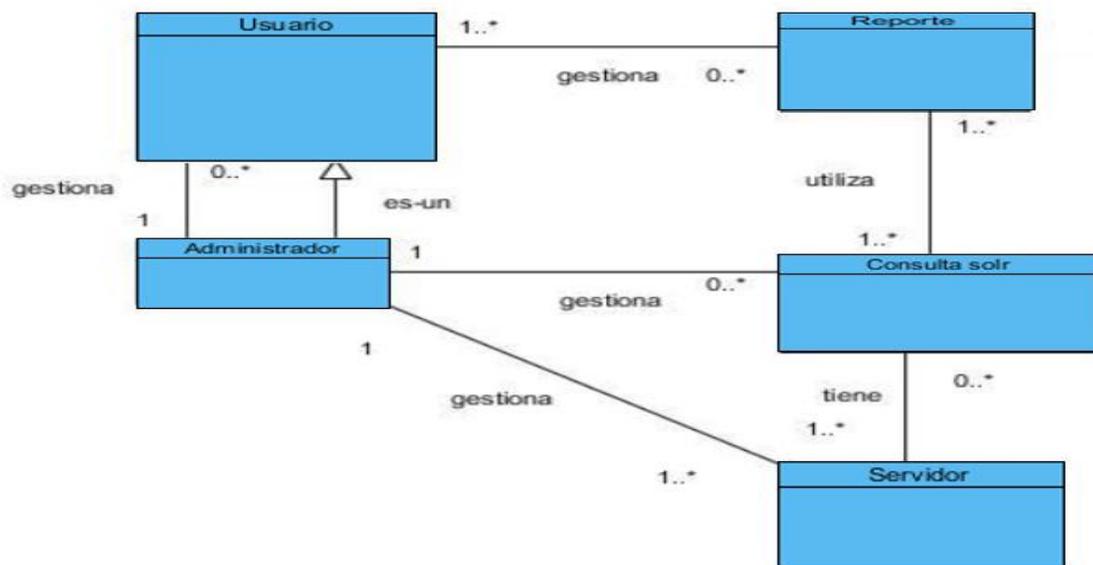


Figura 2 Modelo de Dominio del Módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.

2.3.1. Descripción del modelo de dominio

El administrador al acceder al sistema puede gestionar todos los elementos: consulta solr, reportes, servidores y usuarios (este último aprovechando la propia gestión de usuarios de Alfresco Community). Todos estos pueden ser creados, modificados y eliminados. Los reportes se realizan mediante la interfaz, insertando el nombre del reporte y los campos que se requieran para las estadísticas solicitadas. El usuario deberá llenar los campos requeridos para la obtención del reporte, teniendo como permiso además de su gestión, la opción de visualizarlo. El reporte quedará guardado en la base de datos (si así lo decide el usuario) permaneciendo accesible; y brindando la opción de exportarlo a formato PDF una vez se obtenga el reporte. Para la obtención de las estadísticas de los reportes, cada reporte realizado será una consulta solr. Esta será enviada al servidor seleccionado al cual se debe conectar el sistema. En este caso las estadísticas que consumirá el módulo de reportes, será las del XABAL Repxos implementado sobre el DSpace. Por lo tanto el servidor escogido será el de DSpace.

2.4. Diagrama de paquetes

El objetivo de estos diagramas es obtener una visión más clara del sistema de información orientado a objetos, organizándolo en subsistemas, agrupando los elementos del análisis, diseño o construcción y detallando las relaciones de dependencia entre ellos. El mecanismo de agrupación se denomina *Paquete*. (Cillero, Manuel Cillero, 2019).

Estos diagramas contienen dos tipos de elementos:

- Paquetes: Un paquete es una agrupación de elementos, bien sea casos de uso, clases o componentes. Los paquetes pueden contener a su vez otros paquetes anidados que en última instancia contendrán alguno de los elementos anteriores.
- Dependencias entre paquetes: Existe una dependencia cuando un elemento de un paquete requiere de otro que pertenece a un paquete distinto. Es importante resaltar que las dependencias no son transitivas.

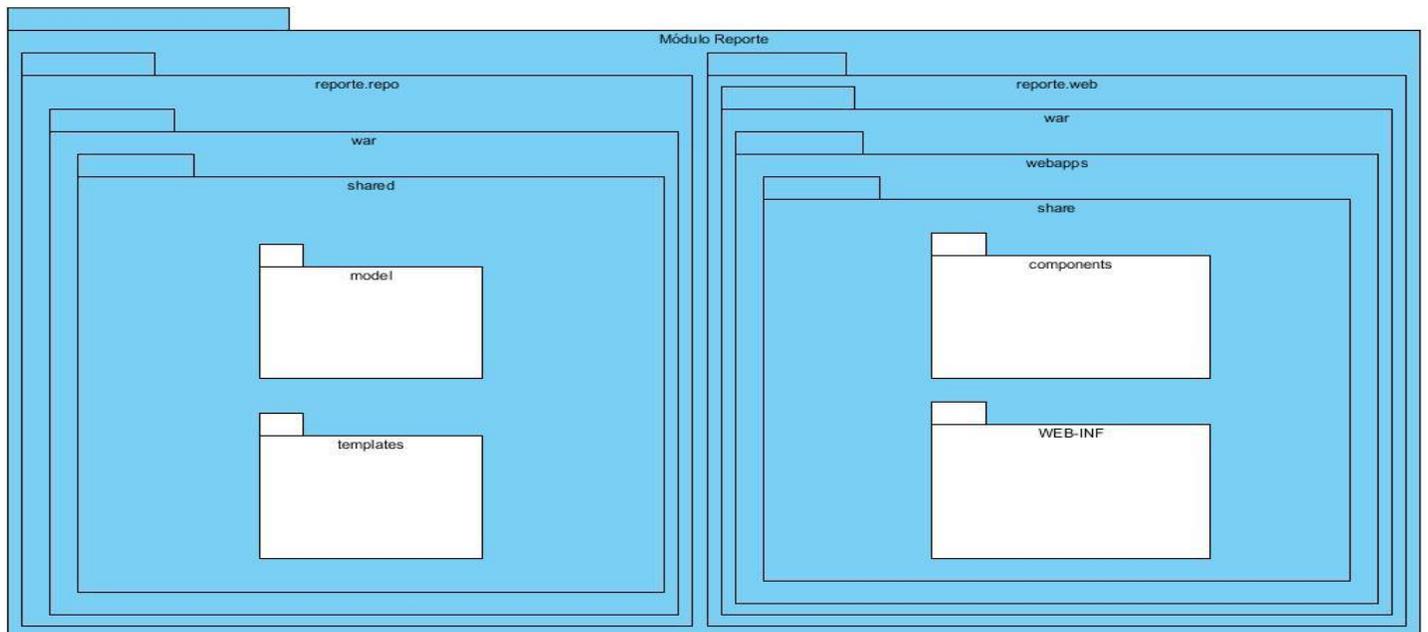


Figura 3 Diagrama de paquetes

2.4.1. Descripción de los paquetes

En este diagrama se puede apreciar un paquete que contiene todos los demás paquetes el cual es el módulo de reportes. Este paquete se encuentra dividido en dos principales los cuales contienen varios paquetes pero el principal para la solución del problema son los paquetes model (contiene el modelo de contenido), templates (contiene los servicios de datos), components (vistas y clases jsp, paquetes jspui y clases de java) y el paquete WEB-INF se encuentra las librerías necesarias para el correcto funcionamiento del sistema.

2.5. Especificación de requisitos de software

La ingeniería de requisitos ayuda a los ingenieros de software a entender el problema en cuya solución trabajarán. Incluyen el conjunto de tareas que conducen a comprender cuál será el impacto del software sobre el negocio, que es lo que el cliente quiere y como interactuarán los usuarios finales con él software.

Los requisitos se clasifican en dos tipos; los requisitos funcionales y los requisitos no funcionales. Los requisitos funcionales describen lo que el sistema debe hacer. Mientras que los requisitos no funcionales

son aquellos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (PRESSMAN R. S., 2005).

2.5.1. Especificación de los requisitos funcionales

Los requerimientos funcionales (RF) son capacidades o condiciones que el sistema debe cumplir, define qué es lo que el sistema debe hacer, así como las funciones que será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas, para producir salidas (PRESSMAN R. S., 2005).

Gestión de servidores:

RF1: Agregar servidor solr.

RF2: Listar servidores solr.

RF3: Buscar servidor solr.

RF4: Filtrar búsqueda por nombre de servidor solr.

RF5: Filtrar búsqueda por dirección de servidor solr.

RF6: Editar servidor solr.

RF7: Eliminar servidor solr.

Gestión de consultas:

RF1: Crear consulta solr.

RF2: Listar consultas solr.

RF3: Buscar consulta solr.

RF4: Filtrar búsqueda por título de consulta solr.

RF5: Filtrar búsqueda por parámetros de consulta solr.

RF6: Filtrar búsqueda por nombre de parámetros de consulta solr.

RF7: Filtrar búsqueda por servidor de consulta solr.

RF8: Filtrar búsqueda por fl de consulta solr.

RF9: Filtrar búsqueda por nombre fl de consulta solr.

RF10: Mostrar consulta solr.

RF11: Editar consulta solr.

RF12: Eliminar consulta solr.

Gestión de servidores

RF1: Crear reporte.

RF2: Listar reporte.

RF3: Buscar reporte.

RF4: Filtrar búsqueda por título de reporte.

RF5: Filtrar búsqueda por autor de reporte.

RF6: Mostrar reporte.

RF7: Exportar reporte a formato .pdf.

RF8: Editar reporte.

RF9: Eliminar reporte.

En la tabla 1 se muestran los requisitos funcionales del sistema.

Tabla 1 Requisitos Funcionales del módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.

No.	Nombre	Descripción	Nivel de acceso
RF1	Agregar servidor solr	Permite adicionar un nuevo servidor solr	Administrativo
RF2	Buscar servidor solr	Permite buscar un servidor solr	Administrativo
RF3	Listar servidores solr	Lista todos los servidores agregados	Administrativo
RF4	Filtrar búsqueda por nombre de servidor solr	Permite filtrar la búsqueda por el nombre del servidor	Administrativo

RF5	Filtrar búsqueda por dirección de servidor solr	Permite filtrar la búsqueda por la dirección del servidor	Administrativo
RF6	Editar servidor solr	Permite editar los datos del servidor seleccionado	Administrativo
RF7	Eliminar servidor solr	Permite eliminar el servidor seleccionado	Administrativo
RF8	Crear consulta solr	Permite crear una consulta solr	Administrativo
RF9	Listar consultas solr	Lista todas las consultas creadas solr	Administrativo
RF10	Buscar consulta solr	Permite buscar un servidor solr	Administrativo
RF11	Filtrar búsqueda por título de consulta solr	Permite filtrar la búsqueda por el título de la consulta	Administrativo
RF12	Filtrar búsqueda por parámetros de consulta solr	Permite filtrar la búsqueda por los parámetros de la consulta	Administrativo
RF13	Filtrar búsqueda por nombre de parámetros de consulta solr	Permite filtrar la búsqueda por el nombre de parámetros de la consulta	Administrativo
RF14	Filtrar búsqueda por servidor de consulta solr	Permite filtrar la búsqueda por el servidor asociado a la consulta	Administrativo
RF15	Filtrar búsqueda por fl de consulta solr	Permite filtrar la búsqueda por el fl de la consulta	Administrativo
RF16	Filtrar búsqueda por nombre fl de consulta solr	Permite filtrar la búsqueda por el nombre fl de la consulta	Administrativo
RF17	Mostrar consulta solr	Permite visualizar en una ventana emergente los datos de la consulta seleccionada	Administrativo
RF18	Editar consulta solr	Permite editar los datos de la consulta seleccionado	Administrativo

RF19	Eliminar consulta solr	Permite eliminar la consulta seleccionada	Administrativo
RF20	Crear reporte	Permite crear un reporte	Público
RF21	Listar reporte	Lista todos los reportes creados	Público
RF22	Buscar reporte	Permite buscar un reporte creado	Público
RF23	Filtrar búsqueda por título de reporte	Permite filtrar la búsqueda por el título del reporte	Público
RF24	Filtrar búsqueda por autor de reporte	Permite filtrar la búsqueda por el autor del reporte	Público
RF25	Mostrar reporte	Permite visualizar en una ventana emergente los datos del reporte seleccionado	Público
RF26	Exportar reporte a formato .pdf	Permite exportar a formato .pdf el reporte seleccionado	Público
RF27	Editar reporte	Permite editar los datos de un reporte seleccionado	Público
RF28	Eliminar reporte	Permite eliminar el reporte seleccionado	Público

2.5.2. Especificación de los requisitos no funcionales

Los requerimientos no funcionales (RNF) son propiedades o cualidades que el producto debe tener, debe pensarse en propiedades que hacen al producto atractivo, usable, rápido o confiable. Son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino a las propiedades emergentes de este, como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento (PRESSMAN R. S., 2005).

Seguridad

El sistema debe garantizar la protección de los datos almacenados. Para ello se establecerán dos roles (administrador y usuario) garantizando que cada persona tenga acceso solamente a las funcionalidades según los permisos de su rol. Se aplicará en la autenticación de usuario la encriptación de la contraseña.

Interfaces de usuario

Según el uso del framework de diseño Bootstrap versión 2.3.2, para acceder a Módulo de Reportes, debe usarse una versión del navegador Mozilla/Firefox igual o superior a la 27.0 o Internet Explorer igual o superior a la 9.0. Específicamente a partir de la versión 8.0 de Internet Explorer no soporta distintas funcionalidades CSS 3. No se garantiza su correcto funcionamiento en otros navegadores.

Interfaces Software

El sistema debe integrarse con los siguientes productos de software en estas versiones:

PostgreSQL 9.4.1: es un sistema de administración de base de datos relacional. Almacena de forma persistente la información necesaria para el funcionamiento del sistema.

Apache Tomcat 7.0: es un servidor HTTP en el cual estará publicado el sistema.

2.6. Definiciones de actores

Actores del sistema

Un actor es una agrupación uniforme de personas, sistemas o máquinas externas al módulo que se está desarrollando, se relaciona con éste ya que solicita sus funcionalidades. En el Módulo de Reportes interactúan dos actores, los cuales se definen a continuación:

- Administrador: Establece todos los permisos sobre el sistema.
- Usuario: Establece los criterios de obtención de los Reportes.

2.7. Diagramas de casos de uso del sistema

El diagrama de casos de uso documenta el comportamiento de un sistema desde el punto de vista del usuario. Por lo tanto, los casos de uso determinan los requisitos funcionales del sistema, es decir,

representan las funciones que un sistema puede ejecutar. Su ventaja principal es la facilidad para interpretarlos, lo que hace que sean especialmente útiles en la comunicación con el cliente. Observar en la figura 4 el diagrama de casos de uso del sistema.

Definición de los casos de uso del sistema

- Gestionar Servidor Solr
- Buscar Servidor Solr
- Gestionar Consulta Solr
- Buscar Consulta Solr
- Gestionar Reportes
- Buscar Reporte
- Exportar a Formato PDF

Diagrama de casos de Uso

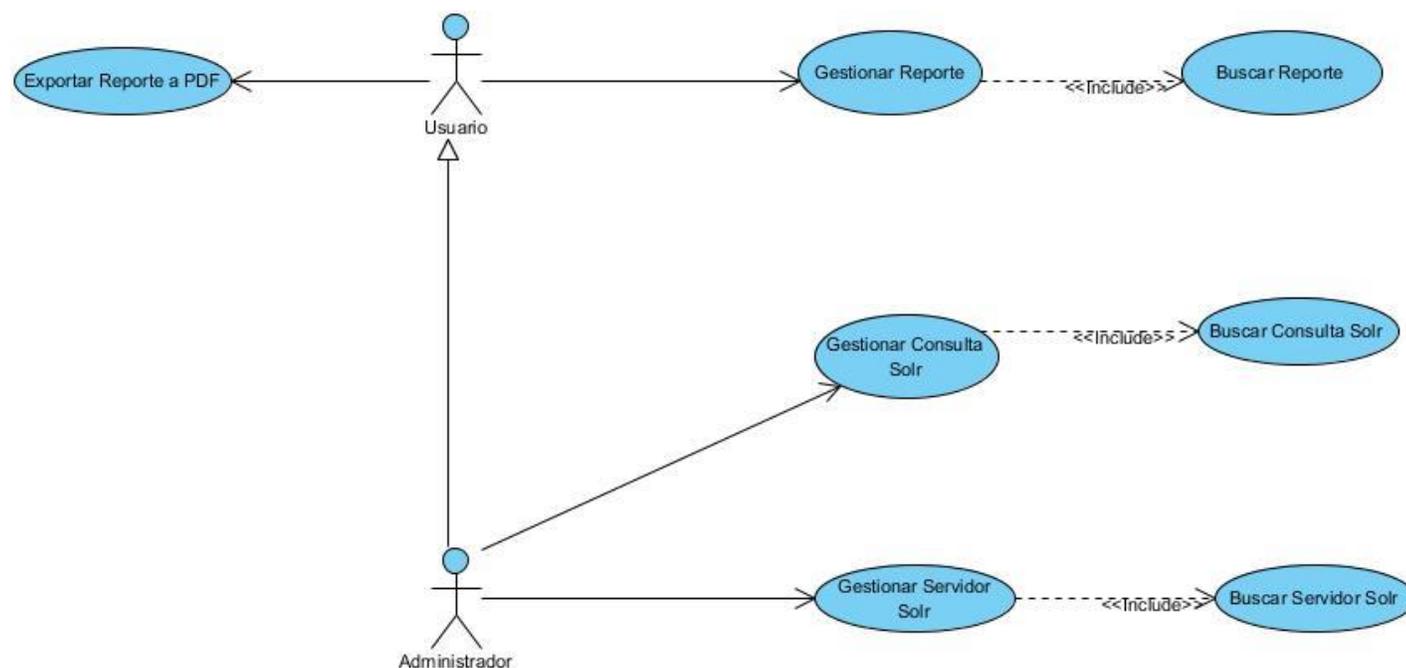


Figura 4 Diagrama de Caso de Usos del Sistema.

2.7.1. Descripción de los casos de usos del sistema

A continuación, se muestran el resumen de las descripciones correspondiente a los casos de uso identificados anteriormente:

Tabla 2 Diagrama de CU Gestionar Consulta Solr

Objetivo	Registrar en el sistema la consulta SOLR, pudiendo en cualquier momento mostrar dicha información, modificarla e incluso eliminarla.	
Actores	Administrador: (Inicia) Crea, modifica y elimina la consulta SOLR.	
Resumen	El administrador crea la consulta SOLR, puede en este caso modificarla, eliminarla y mostrarla.	
Complejidad	Alta	
Prioridad	Media	
Precondiciones		
Postcondiciones	Es creado, modificado y eliminado la consulta SOLR.	
Flujo de eventos		
Flujo básico Gestionar consulta SOLR		
	Actor	Sistema
1.	Selecciona la opción Consulta SOLR.	
2.		Muestra un menú con las siguientes opciones: <ul style="list-style-type: none"> • Crear Consulta SOLR • Modificar Consulta SOLR • Eliminar Consulta SOLR.
3.	Selecciona una de las opciones: crear Consulta SOLR, modificar Consulta SOLR o eliminar Consulta SOLR.	
4.		Se ejecuta algunas de las siguientes acciones: <ol style="list-style-type: none"> 1 Si decide crear un Consulta SOLR, ir a la sección 1“Crear Consulta SOLR”. 2 Si decide Modificar los datos de un Consulta SOLR, ir a la sección 2 “Modificar Consulta SOLR”. 3 Si decide eliminar un Consulta SOLR, ir a la sección 3“Eliminar Consulta SOLR”.
5.		Termina caso de uso
Sección 1: “Crear Consulta SOLR”		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción crear Consulta SOLR.	
2.		Muestra una plantilla que permite introducir los siguientes datos: <ul style="list-style-type: none"> • Título • Parámetros

		<ul style="list-style-type: none"> • Nombre de parámetros • FI • Nombre fl Permite seleccionar: <ul style="list-style-type: none"> • Servidor Además brinda la posibilidad de: <ul style="list-style-type: none"> • Aceptar • Cancelar
3.	Introduce/ selecciona los datos y presiona el botón Aceptar.	
4.		Verifica los datos.
5.		Almacena los datos y muestra la lista de Consulta Solr.
6.		Termina caso de uso.
Flujos alternos		
N.º 3a Selecciona la opción cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 4a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje de información: "El campo está vacío".
2.		Termina caso de uso.
Sección 2: "Modificar Consulta SOLR"		
Flujo básico Modificar Consulta SOLR		
	Actor	Sistema
1.	Selecciona la opción Modificar.	
2.		Permite realizar una búsqueda de Consulta SOLR, para seleccionar la consulta SOLR a modificar. (Ver caso de uso Buscar Consulta SOLR).
3.	Selecciona el botón modificar.	
4.		Muestra plantilla con los parámetros de la consulta SOLR: <ul style="list-style-type: none"> • Título • Parámetros • Nombre de parámetros • FI • Nombre fl Permite seleccionar: <ul style="list-style-type: none"> • Servidor Permite: <ul style="list-style-type: none"> • Aceptar • Cancelar

5.	Introduce todos los datos y presiona el botón Aceptar.	
6.		Verifica los datos.
7.		Almacena los datos.
8.		Termina caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 6a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje “El campo está vacío”.
2.		Termina caso de uso.
Sección 3: “Eliminar Consulta SOLR”		
Flujo básico Eliminar Consulta SOLR		
	Actor	Sistema
1.	Selecciona la opción Eliminar Consulta SOLR.	
2.		Permite realizar una búsqueda de Consulta SOLR, para seleccionar la consulta Solr a eliminar. (Ver caso de uso Buscar Consulta SOLR).
3.	Selecciona el botón Eliminar.	
4.		Muestra un mensaje “Está seguro que desea eliminar la consulta SOLR”.
5.	Selecciona el botón Aceptar.	
6.		Elimina la consulta SOLR seleccionada y muestra la lista de Consulta SOLR.
7.		Termina el caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Relaciones	CU incluidos	Eliminar Consulta SOLR. Ver CU Buscar Consulta SOLR . Modificar Consulta SOLR. Ver CU Buscar Consulta SOLR .
	CU extendidos	NA
Requisitos funcionales	no	NA

Asuntos pendientes	NA
---------------------------	----

Tabla 3 Diagrama de CU Buscar Consulta Solr

Objetivo	Buscar una Consulta SOLR en la base de datos.	
Actores	Administrador.	
Resumen	El caso de uso se inicia cuando el administrador decide buscar una Consulta SOLR dado su nombre para modificarla o eliminarla.	
Complejidad	Baja	
Prioridad	Media	
Precondicione s	La consulta SOLR debe estar registrada.	
Postcondicion es	Se muestran todos los datos de la consulta SOLR.	
Flujo de eventos		
Flujo básico Buscar Consulta SOLR		
	Actor	Sistema
1	Selecciona la opción Buscar Consulta SOLR.	
2		Muestra una ventana con el campo de búsqueda: <ul style="list-style-type: none"> • Nombre de Consulta SOLR. Permite: <ul style="list-style-type: none"> • Buscar • Cancelar
3	Introduce los datos y presiona el botón Buscar.	
4		Verifica que los datos introducidos estén correctos.
5		Muestra una ventana con la consulta SOLR que coincide con los parámetros de búsqueda.
6		Termina el caso de uso.
Flujos alternos		
N.º 4a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje “El campo está vacío”.
2.		Termina caso de uso.
Flujos alternos		
N.º 4a Búsqueda Vacía.		
	Actor	Sistema
1		Muestra un mensaje de confirmación: “Búsqueda Vacía”.

Relaciones	CU incluidos	NA
	CU extendidos	NA
Requisitos no funcionales	NA	
Asuntos pendientes	NA	

Tabla 4 Diagrama de CU Gestionar Reporte

Objetivo	Gestionar los datos de los reportes, pudiendo en cualquier momento mostrar dichos datos, modificarlos o eliminarlos.	
Actores	Usuario (Inicia)	
Resumen	EL caso de uso inicia cuando el usuario crea o modifica un reporte.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe estar registrado al menos una Consulta SOLR.	
Postcondiciones	Reporte creado en la base de datos.	
Flujo de eventos		
Flujo básico Gestionar reporte		
	Actor	Sistema
1.	Selecciona de la página inicial la opción reportes.	
2.		Muestra un menú con las siguientes opciones: <ul style="list-style-type: none"> • Crear reportes. • Modificar reportes.
3.	Selecciona una de las opciones: crear o modificar reportes.	
4.		Se ejecuta algunas de las siguientes acciones: <ol style="list-style-type: none"> Si decide crear un reporte, ir a la sección 1“Crear reporte” Si decide modificar los datos de un reporte, ir a la sección 2 “Modificar reporte”.
5.		Termina caso de uso.
Sección 1: “Crear Reporte”		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción Crear reporte.	
2.		Permite realizar una búsqueda de Consulta SOLR, para seleccionar la consulta SOLR por el cual creará el reporte. (Ver caso de uso Buscar Consulta SOLR).

3.	Selecciona el botón Crear.	
4.		<p>Muestra plantilla con los parámetros para entrar los datos:</p> <ul style="list-style-type: none"> • Título • Autor <p>Permite seleccionar:</p> <ul style="list-style-type: none"> • Consulta solr <p>Además permite:</p> <ul style="list-style-type: none"> -Crear -Cancelar
5.	Introduce los datos y presiona el botón Crear.	
6.		Verifica los datos.
7.		Almacena los datos.
8.		Termina caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 6a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje "El campo está vacío".
2.		Termina caso de uso.
Sección 2: "Modificar Reporte"		
Flujo básico Modificar Reporte		
	Actor	Sistema
1.	Selecciona la opción Modificar.	
2.		Permite realizar una búsqueda de reporte, para seleccionar el reporte que modificará creando un nuevo reporte. (Ver caso de uso Buscar Reporte).
3.	Selecciona el botón Modificar.	
4.		<p>Muestra plantilla del Reporte especificado con los parámetros para entrar los nuevos datos:</p> <ul style="list-style-type: none"> • Título. • Los otros datos son en dependencia de los parámetros del antiguo reporte. <p>Permite:</p> <ul style="list-style-type: none"> • Modificar • Cancelar

5.	Introduce el título y los datos y presiona el botón Modificar.	
6.		Verifica los datos.
7.		Almacena los datos.
8.		Termina caso de uso.
Flujos alternos		
N.º Evento 5a Opción cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 5a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje "El campo está vacío".
2.		Termina caso de uso.
Sección 3: "Eliminar Reporte"		
Flujo básico Eliminar Consulta Reporte		
	Actor	Sistema
1.	Selecciona la opción Eliminar.	
2.		Permite realizar una búsqueda de Reportes, para seleccionar el Reporte a eliminar. (Ver caso de uso Buscar Reporte).
3.	Selecciona el botón Eliminar.	
4.		Muestra un mensaje "Está seguro que desea eliminar el reporte". Permite: <ul style="list-style-type: none"> • Aceptar • Cancelar
5.	Selecciona el botón Aceptar	
6.		Elimina el reporte seleccionado y re direcciona la página hacia la lista de Reportes.
7.		Termina caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.

2.		Termina el caso de uso.
Relaciones	CU incluidos	Eliminar Reporte. Ver CU Buscar Reporte . Modificar Reporte. Ver CU Buscar Reporte . Crear Reporte. Ver CU Buscar Consulta SOLR .
	CU extendidos	NA
Requisitos funcionales	noNA	
Asuntos pendientes	NA	

Tabla 5 Diagrama de CU Buscar Consulta Solr

Objetivo	Buscar un Reporte en la base de datos.	
Actores	Usuario.	
Resumen	El caso de uso se inicia cuando el usuario decide buscar un Reporte dado su nombre para modificarlo, eliminarlo o exportarlo a pdf.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El Reporte debe estar registrada.	
Postcondiciones	Se muestran todos los datos del Reporte.	
Flujo de eventos		
Flujo básico Reportes		
	Actor	Sistema
1	Selecciona la opción Buscar Reporte.	
2		Se introduce en el campo de búsqueda: <ul style="list-style-type: none"> Nombre de Reporte. Permite: <ul style="list-style-type: none"> Buscar Cancelar
3	Introduce los datos y presiona el botón Buscar.	
4		Verifica que los datos introducidos estén correctos.
5		Muestra el resultado del reporte buscado
6		Termina el caso de uso.
Flujos alternos		
N.º 4a Campos Vacíos.		
	Actor	Sistema

1.		No se realiza búsqueda.
2.		Termina caso de uso.
Flujos alternos		
N.º 4a Búsqueda Vacía.		
	Actor	Sistema
1		Muestra un mensaje de: "Búsqueda Vacía".
Relaciones	CU incluidos	NA
	CU extendidos	NA
Requisitos no funcionales	NA	
Asuntos pendientes	NA	

Tabla 6 Diagrama de CU Buscar Reporte

Objetivo	Buscar un reporte en la base de datos.	
Actores	Usuario.	
Resumen	El caso de uso inicia cuando el usuario decide buscar un reporte dado su nombre para verlo o modificarlo.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	El reporte debe estar creado.	
Postcondiciones	Se muestran todos los datos del reporte.	
Flujo de eventos		
Flujo básico Buscar Reporte		
	Actor	Sistema
1.	Selecciona la opción Buscar reporte.	
2.		Se introduce el criterio de búsqueda <ul style="list-style-type: none"> • Título del reporte. • Nombre de reporte.
3.	Introduce los datos y presiona el botón Buscar.	
4.		Verifica que los datos introducidos estén correctos.
5.		Muestra una ventana con el reporte que coincide con los parámetros de búsqueda.
6.		Termina el caso de uso.

Flujos alternos		
N.º 4a Campos Vacíos.		
	Actor	Sistema
1		Muestra un mensaje de información: “El campo está vacío”.
2		Termina caso de uso.
Flujos alternos		
N.º 4a Búsqueda Vacía.		
	Actor	Sistema
1		Muestra un mensaje de información: “Búsqueda Vacía”.
	Relaciones	CU incluidos NA
		CU extendidos NA
	Requisitos funcionales no	NA
	Asuntos pendientes	NA

Tabla 7 Diagrama de CU Gestionar Servidor Solr

Objetivo	Gestionar los Servidores a los cuales se va a conectar el sistema, pudiendo en cualquier momento registrar dichos servidores, modificarlos y eliminarlos.	
Actores	Administrador (Inicia)	
Resumen	El caso de uso inicia cuando el administrador crea un servidor para conectar el sistema a este. Puede además modificarlos y eliminarlos.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Tener registrados los servidores reales a los cuales puede conectarse el sistema.	
Postcondiciones	Servidor creado en la base de datos.	
Flujo de eventos		
Flujo básico Gestionar Servidor SOLR		
	Actor	Sistema
1.	Selecciona de la página inicial Gestionar servidor.	
2.		Muestra un menú con las siguientes opciones: <ul style="list-style-type: none"> • Agregar Servidor. • Modificar Servidor. • Eliminar Servidor.

3.	Selecciona una de las opciones: agregar, modificar o eliminar servidor.	
4.		Se ejecuta algunas de las siguientes acciones: a) Si decide registrar un servidor, ir a la sección 1 "Agregar Servidor". b) Si decide modificar los datos de un Servidor, ir a la sección 2 "Modificar Servidor". c) Si decide eliminar un servidor, ir a la sección 3 "Eliminar Servidor".
5.		Termina caso de uso.
Sección 1: "Agregar Servidor"		
Flujo básico Agregar Servidor		
	Actor	Sistema
1.	Selecciona la opción registrar servidor.	
2.		Muestra plantilla con los parámetros del servidor: <ul style="list-style-type: none"> • Nombre • Dirección Permite: <ul style="list-style-type: none"> • Agregar • Cancelar
3.	Introduce los datos y presiona el botón Crear.	
4.		Verifica todos los datos.
5.		Termina caso de uso.
Flujos alternos		
Nº 3a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Nº 4a Campos vacíos		
1.		Muestra un mensaje: "Campos vacíos".
2.		Termina el caso de uso.
Sección 2: "Modificar Servidor"		
Flujo básico Modificar servidor.		
	Actor	Sistema
1.	Selecciona la opción Modificar.	
2.		Muestra un menú con una lista de servidores para seleccionar uno.

3.	Selecciona el servidor y presiona el botón Modificar.	
4.		<p>Muestra plantilla del servidor especificado con los parámetros para entrar los nuevos datos:</p> <ul style="list-style-type: none"> • Nombre • Dirección <p>Permite:</p> <ul style="list-style-type: none"> • Aceptar • Cancelar
5.	Introduce los datos y presiona el botón Aceptar.	
6.		Verifica los datos de los campos.
7.		Almacena los datos.
8.		Termina caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema
1.		Re direcciona hacia la página principal.
2.		Termina el caso de uso.
Flujos alternos		
N.º 6a Campos Vacíos.		
	Actor	Sistema
1.		Muestra un mensaje: "Campos vacíos".
2.		Termina el caso de uso.
Sección 3: "Eliminar servidor "		
Flujo básico Eliminar servidor.		
	Actor	Sistema
1.	Selecciona la opción Eliminar.	
2.		Muestra un menú con una lista de servidores para seleccionar uno.
3.	Selecciona un servidor y presiona el botón Eliminar.	
4.		Muestra un mensaje para confirmar la eliminación: "Está seguro que desea eliminar".
5.	Selecciona el botón Aceptar.	
6.		Elimina el servidor seleccionado y re direcciona la página hacia la lista de servidores con el correspondiente eliminado.
7.		Termina caso de uso.
Flujos alternos		
N.º 5a Selecciona la opción Cancelar.		
	Actor	Sistema

1.		Re direcciona hacia la página principal.
2.		Termina caso de uso.
Relaciones	CU incluidos	NA
	CU extendidos	NA
Requisitos no funcionales	NA	
Asuntos pendientes	NA	

Tabla 8 Diagrama de CU Exportar a formato PDF

Objetivo	Exporta a formato a PDF los reportes que desee el usuario.	
Actores	Usuario (Inicia)	
Resumen	El usuario después de obtener su reporte selecciona la opción exportar a formato PDF.	
Complejidad	Baja	
Prioridad	Media	
Precondiciones	Debe haber un Reporte creado y mostrado.	
Postcondiciones	El reporte es mostrado en formato PDF.	
Flujo de eventos		
Flujo básico		
	Actor	Sistema
1.	Selecciona la opción exportar a PDF.	
2.		Muestra el Reporte con sus datos e información actualizados en formato físico PDF.
3.		Termina el caso de uso.
Relaciones	CU incluidos	NA
	CU extendidos	NA
Requisitos no funcionales	NA	
Asuntos pendientes	NA	

2.8. Interfaces del sistema

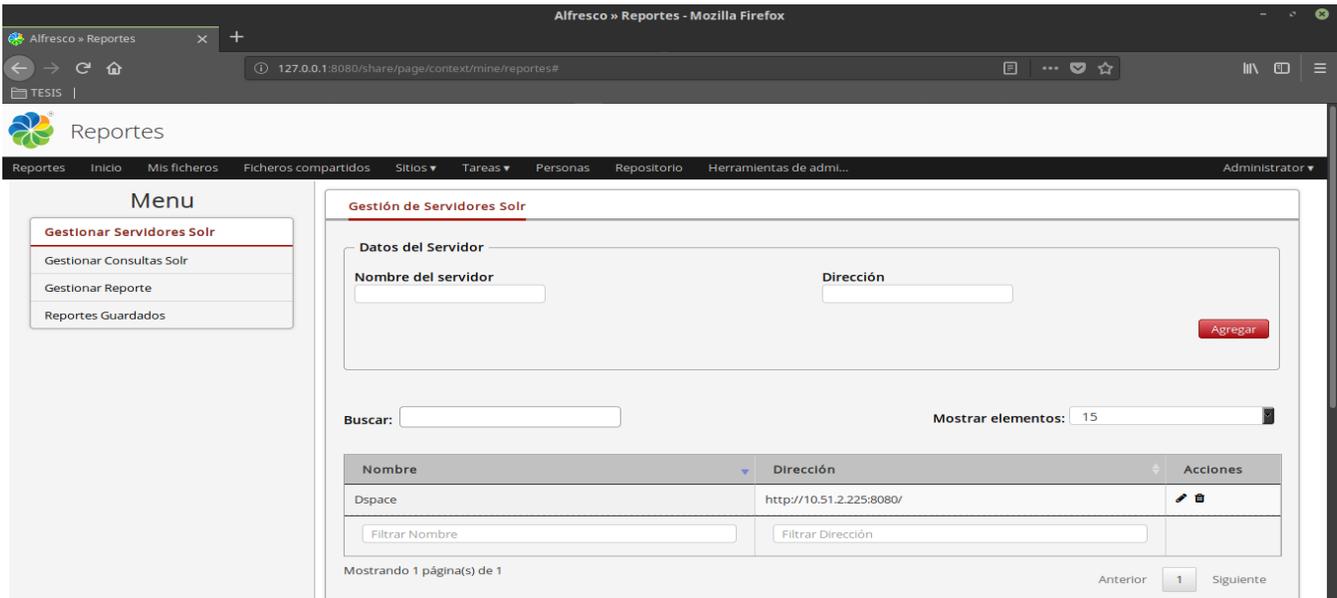


Figura 5 Interfaces del sistema de los casos de uso "Gestionar Servidor Solr " y "Buscar Servidor Solr"

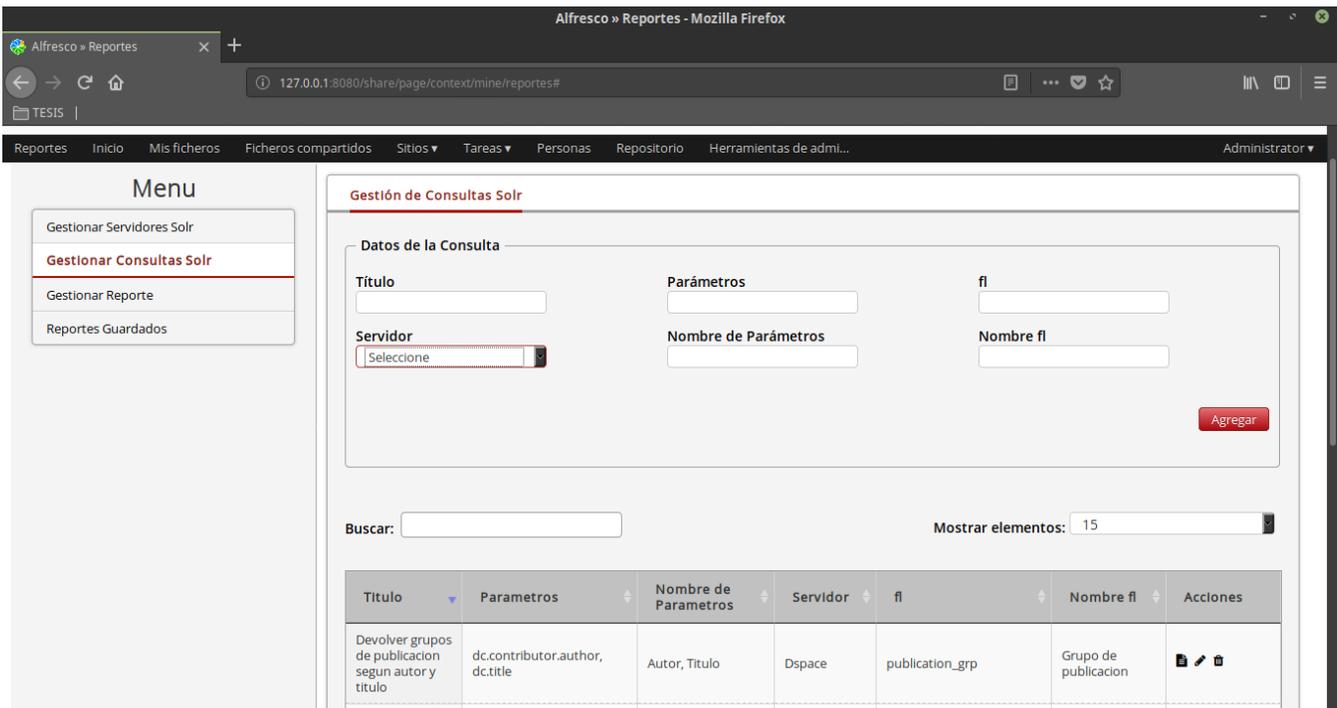


Figura 6 Interfaces del sistema de los casos de uso "Gestionar Consulta Solr " y "Buscar Consulta Solr"

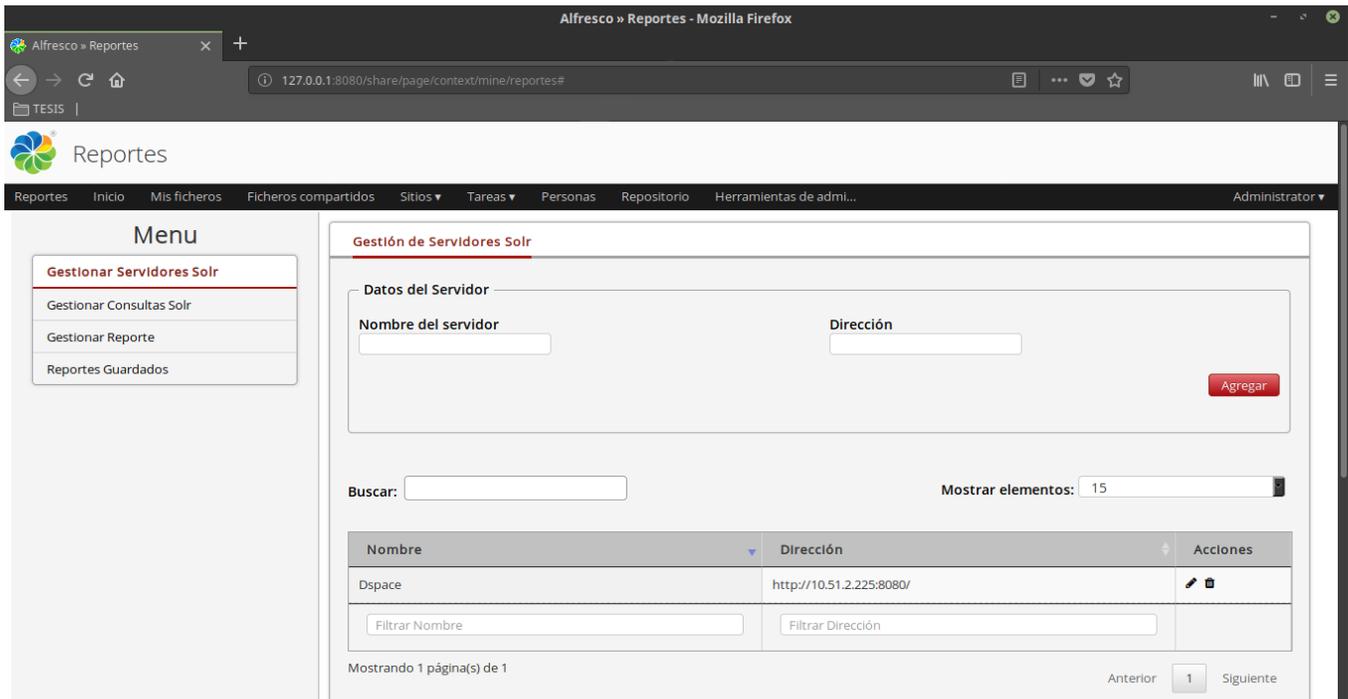


Figura 7 Interfaces del sistema de los casos de uso "Gestionar Reporte"

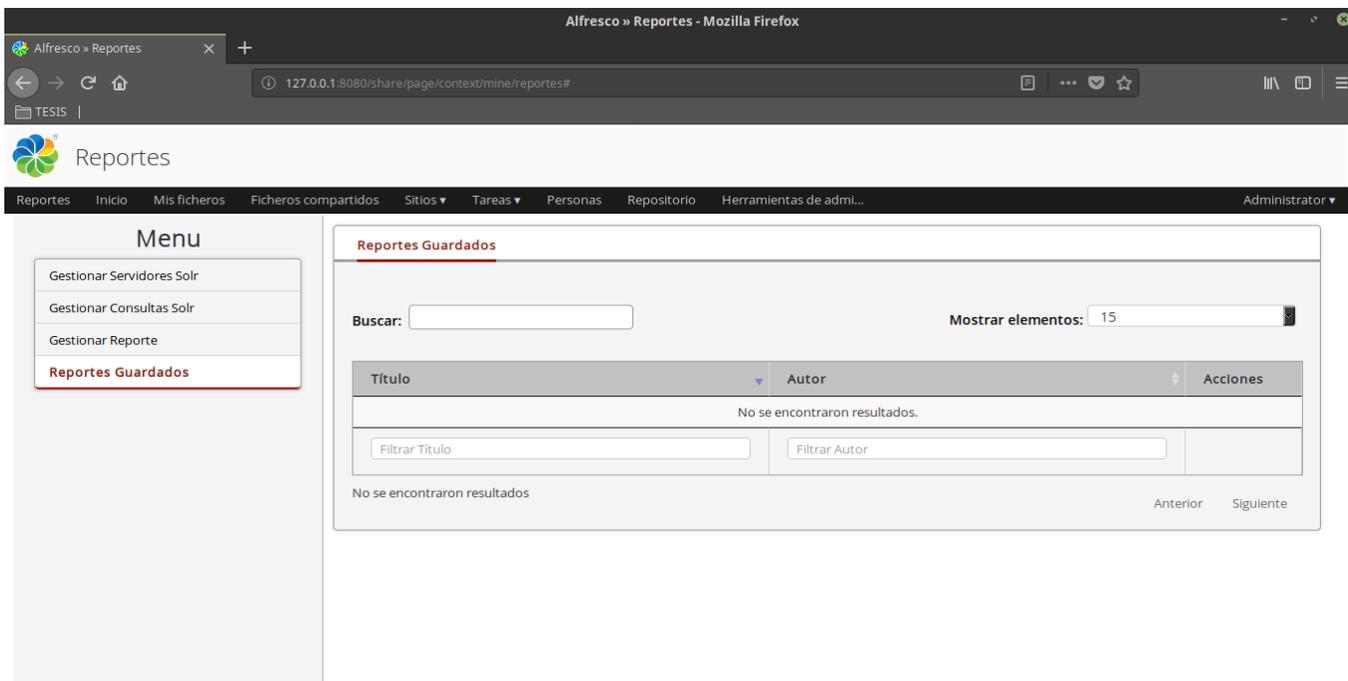


Figura 8 Interfaces del sistema de los casos de uso "Buscar Reporte"

2.9. Matrices de trazabilidad

La matriz de trazabilidad relaciona dos elementos esenciales para la buena ejecución de las labores de un proyecto: los requisitos establecidos para dicha ejecución y el valor que cada uno de ellos agrega al conjunto del proceso. Es una herramienta clave para la ingeniería de los proyectos, así como para el seguimiento de los diversos elementos que los componen (OBS, 2018).

(28) Requirement	Agregar servidor solr	Buscar consulta solr	Buscar reporte	Buscar servidor solr	Crear consulta solr	Crear reporte	Editar consulta solr	Editar reporte	Editar servidor solr	Eliminar consulta solr	Eliminar reporte	Eliminar servidor solr	Exportar reporte a formato ...	Filtrar búsqueda por autor ...	Filtrar búsqueda por direcc...	Filtrar búsqueda por fl de c...	Filtrar búsqueda por nombr...	Filtrar búsqueda por nombr...	Filtrar búsqueda por nombr...	Filtrar búsqueda por parám...	Filtrar búsqueda por servid...	Filtrar búsqueda por título d...	Filtrar búsqueda por título d...	Listar consultas solr	Listar reporte	Listar servidores solr	Mostrar consulta solr	Mostrar reporte
(28) Requirement																												
Agregar servidor solr																												
Buscar consulta solr																	✓	✓			✓	✓	✓	✓		✓		
Buscar reporte																									✓			
Buscar servidor solr															✓											✓		
Crear consulta solr																												
Crear reporte																												
Editar consulta solr																												
Editar reporte																												
Editar servidor solr																												
Eliminar consulta solr																												
Eliminar reporte																												
Eliminar servidor solr																												
Exportar reporte a formato ...																												✓
Filtrar búsqueda por autor ...																												
Filtrar búsqueda por direcc...				✓																								
Filtrar búsqueda por fl de c...	✓																											
Filtrar búsqueda por nombr...	✓																											
Filtrar búsqueda por nombr...				✓																								
Filtrar búsqueda por nombr...	✓																											
Filtrar búsqueda por parám...	✓																											
Filtrar búsqueda por servid...	✓																											
Filtrar búsqueda por título d...	✓																											
Filtrar búsqueda por título d...																												
Listar consultas solr		✓			✓		✓			✓						✓	✓			✓	✓	✓	✓				✓	
Listar reporte			✓			✓	✓		✓					✓														✓
Listar servidores solr	✓			✓					✓					✓														
Mostrar consulta solr																												
Mostrar reporte													✓												✓			

Figura 9 Matriz de trazabilidad Requisito-Requisito

(28) Requirement																														
By: <input type="text" value="Transitor"/>		Agregar servidor solr	Buscar consulta solr	Buscar reporte	Buscar servidor solr	Crear consulta solr	Crear reporte	Editar consulta solr	Editar reporte	Editar servidor solr	Eliminar consulta solr	Eliminar reporte	Eliminar servidor solr	Exportar reporte a formato ...	Filtrar búsqueda por autor ...	Filtrar búsqueda por direcc...	Filtrar búsqueda por fi de c...	Filtrar búsqueda por nombr ...	Filtrar búsqueda por nombr ...	Filtrar búsqueda por nombr ...	Filtrar búsqueda por parám...	Filtrar búsqueda por servid...	Filtrar búsqueda por título d...	Filtrar búsqueda por título d...	Listar consultas solr	Listar reporte	Listar servidores solr	Mostrar consulta solr	Mostrar reporte	
(7) Use Case																														
<input checked="" type="radio"/> Buscar Consulta Solr																														
<input checked="" type="radio"/> Buscar Reporte																														
<input checked="" type="radio"/> Buscar Servidor Solr																														
<input checked="" type="radio"/> Exportar PDF																														
<input checked="" type="radio"/> Gestionar Consulta Solr			✓			✓		✓			✓						✓	✓			✓	✓	✓	✓						
<input checked="" type="radio"/> Gestionar Reporte				✓			✓		✓			✓			✓	✓											✓			
<input checked="" type="radio"/> Gestionar Servidor Solr		✓			✓					✓			✓															✓		

Figura 10 Matriz de trazabilidad Caso de Uso-Requisito

(7) Use Case								
By: <input type="text" value="Transitor"/>		Buscar Consulta Solr	Buscar Reporte	Buscar Servidor Solr	Exportar PDF	Gestionar Consulta Solr	Gestionar Reporte	Gestionar Servidor Solr
(7) Test Case								
<input checked="" type="radio"/> Buscar Consulta Solr		✓						
<input checked="" type="radio"/> Buscar Reporte			✓					
<input checked="" type="radio"/> Buscar Servidor Solr				✓				
<input checked="" type="radio"/> Exportar a pdf					✓			
<input checked="" type="radio"/> Gestionar Consulta Solr						✓		
<input checked="" type="radio"/> Gestionar Reporte							✓	
<input checked="" type="radio"/> Gestionar Servidor Solr								✓

Figura 11 Matriz de trazabilidad Caso de Uso- Caso de Prueba

Se muestran 3 matrices de trazabilidad, la primera es requisito contra requisitos donde se evidencia la relación entre todos los requisitos; o sea la dependencia entre ellos. La segunda matriz es caso de uso

contra requisitos, donde se pone de manifiesto que todos los requisitos coinciden con los casos de uso. La última matriz diseñada es caso de uso contra caso de prueba, y como la anterior coincide que todos los casos de uso están relacionados con todos los casos de prueba.

2.10. Descripción de la arquitectura

La arquitectura de software de un sistema de programa o computación es la estructura del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos (PRESSMAN R. S., 2005).

2.10.1. Arquitectura n capas

Es un estilo de programación, basado en la separación de la capa presentación, la capa web y la capa de acceso a repositorios, con el objetivo primordial de simplificar la comprensión y organización del desarrollo del sistema. Este patrón disminuye las dependencias, ya que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. La arquitectura propuesta añade una gran flexibilidad al diseño de la aplicación, así como una interoperabilidad en entornos distribuidos con un nivel de abstracción superior. Las tres capas que se definieron para el desarrollo del módulo son: Presentación, Web y Acceso a Repositorio.

Capa de Presentación

En esta capa se encuentra el conjunto de interfaces de usuario, que les hace posible al cliente y la aplicación establecer la comunicación, manipular los datos, así como representar en términos de componentes visuales, toda la información necesaria, consultada y/o generada por el par aplicación-usuario. Algunos de los componentes son jQuery y AjaxRequest. El framework que se usa es Bootstrap en su versión 2.3 y cuenta con una capa de componentes desarrollados por Alfresco Inc. que hace función tanto de utilitarios como de integración con algunas restricciones del diseño como la internacionalización y la comunicación con los servicios Rest de la capa web. Durante el desarrollo del módulo esta capa se pone de manifiesto en los formularios, tablas y páginas web (Suarez, 2011).

Capa Web

En esta capa se ejecutan todos los procesos de negocio que han sido previamente implementados, se preparan a su vez las transformaciones de datos, sirviendo como un mediador entre las demandas del

cliente y las respuestas de los datos. Controla y dirige el flujo de la aplicación en sentido general. Esta capa no maneja una lógica de negocio fuerte, pero sí tiene implementado servicios para la gestión de la autenticación, la configuración de la caché que será enviada a la presentación, también tiene la posibilidad de comunicarse con más de un repositorio, así como con servicios externos, ella se evidencia en los servicios de presentación. Su función principal es la de adecuar los datos transformando la información del repositorio en estructuras web entendibles por un navegador de archivos (html, js, css, imágenes).

Capa de Acceso a Repositorio

Esta capa es la más compleja y extensa, en ella se implementan un conjunto de servicios que permiten manipular de forma distribuida y a través de la red la información no estructurada y sus metadatos. Esta capa se pone de manifiesto en los modelos de contenido y servicios de datos y hace posible hacer búsquedas sobre los contenidos exportando todos los resultados a través de Rest.

2.10.2. Patrón arquitectónico: Modelo-Vista-Controlador

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

- El **Modelo** que contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia (modelo de contenido de Alfresco Community).
- La **Vista**, o interfaz de usuario, que compone la información que se envía al cliente y los mecanismos interacción con éste (web-inf, vistas para share).
- El **Controlador**, que actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno (archivos .js controladores. Ejem: reportes.js).

El **Modelo** es responsable de:

- Acceder a la capa de almacenamiento de datos.
- Define las reglas de negocio (la funcionalidad del sistema).
- Lleva un registro de las vistas y controladores del sistema.

El **Controlador** es responsable de:

- Recibe los eventos de entrada (un clic, un cambio en un campo de texto, etc.).
- Contiene reglas de gestión de eventos, del tipo "Si Evento Z, entonces Acción W". Estas acciones pueden suponer peticiones al modelo o a las vistas. Una de estas peticiones a las vistas puede ser una llamada al método "validarFormReportes()". Una petición al modelo puede ser "eliminarFila()".

Las **Vistas** son responsables de:

- Recibir datos del modelo y los muestra al usuario.
- Tienen un registro de su controlador asociado.

2.10.3. Patrones de diseño

Un patrón de diseño se define como una solución a un problema que se usa repetidamente en contextos similares, se encarga de identificar, abstraer y nombrar los aspectos elementales de una estructura de diseño, donde los componentes, son las clases y objetos, y sus mecanismos de interacción son mensajes. Ayudan a elegir diseños alternativos que hacen un sistema reutilizable y evitan alternativas que comprometan la reutilización. (LARMAN, 2003).

Para el desarrollo del trabajo se utilizaron los siguientes patrones de diseño:

- Experto

Es el principio básico de asignación de responsabilidades. Pertenece a los patrones generales de software para asignación de responsabilidades (GRASP, por sus siglas en inglés). Indica, que la responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. De este modo se obtiene un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento) (LARMAN, 2003).

En el sistema este patrón se pone de manifiesto en el controlador Reporte.js, este es el encargado de la creación de la consulta y la realización de la petición al servidor Solr para la obtención de cada reporte. De igual forma en el sistema cada controlador tiene asignada la responsabilidad y conoce toda la información necesaria para la gestión de cada objeto o implementación de cada método correspondiente a la entidad a la que pertenece.

- Creador

El patrón creador identifica quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases. Sugiere encontrar clases de objetos que estén vinculadas para hacerlas responsables de la creación de los objetos, manteniendo así un bajo acoplamiento. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, o que agregue o contenga instancias del objeto. En el sistema este patrón se pone de manifiesto en los controladores, pues cada uno de ellos está encargado de la creación de nuevas instancias de las clases entidad a la que están asociados.

- Controlador

Un controlador es un objeto responsable del manejo de los eventos del sistema, que no pertenece a la interfaz del usuario. El controlador recibe la solicitud del servicio desde la capa de presentación y coordina su realización delegando a otros objetos. El uso de este patrón en el sistema está evidenciado en los controladores existentes que son los encargados de manejar los eventos. En el proyecto se pone de manifiesto con los servicios de datos pertenecientes al repositorio.

Conclusiones del capítulo

Se realizó la descripción de la herramienta propuesta, aportando una posible solución a la problemática planteada. Se especificó el modelo de dominio que permitió definir los conceptos fundamentales del negocio. De esta forma se obtuvo una visión más clara y óptima del entorno sobre el cual se desarrolla el problema a resolver. Al identificar las características con las que debe contar la propuesta final de solución se obtuvieron los requisitos funcionales y los requisitos no funcionales. Se modeló el diagrama de casos de uso del sistema posibilitando un mejor entendimiento del proceso. Con las especificaciones de los casos de uso del sistema se establecen los flujos básicos de estos y se logra una descripción detallada para una mejor comprensión del módulo que se va a implementar.

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBAS

3.1. Introducción

En el presente capítulo se describe cómo se ha desarrollado la etapa de implementación y prueba de la propuesta de solución. Para ello se realiza el modelo de implementación del sistema apoyándose en el diagrama de despliegue y además se determinan los tipos de prueba que se realizan y las técnicas utilizadas, así como los casos de pruebas que se le aplican al sistema con el objetivo de comprobar los errores que pueda tener y comprobar que la propuesta de solución cumple con todas sus especificaciones.

Implementación

La disciplina “Implementación” que propone la metodología AUP-UCI, muestra cómo, a partir de los resultados del Análisis y Diseño se construye el sistema (UCI, Metodología de desarrollo para la Actividad productiva de la UCI, 2016).

A partir de los resultados de diseño obtenidos fue realizada la implementación del módulo a través de ficheros de código fuente, scripts, freemarker, entre otras. Esta disciplina explica cómo desarrollar, organizar e integrar el módulo implementado basándose en las especificaciones del diseño. La finalidad del módulo es:

- Agregar servidores a los cuales se les realizará consultas de datos.
- Crear consultas y que estas queden guardadas para la posterior obtención de reportes.
- Mostrar, archivar y/o exportar reportes obtenidos a través de consultas solr.

3.1.1. Diagrama de Despliegue

A continuación en la siguiente figura se muestra el diagrama de despliegue correspondiente al Módulo Reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2, en el mismo se muestran los componentes necesarios para realizar el despliegue y las carpetas donde estos componentes deben ser ubicados:

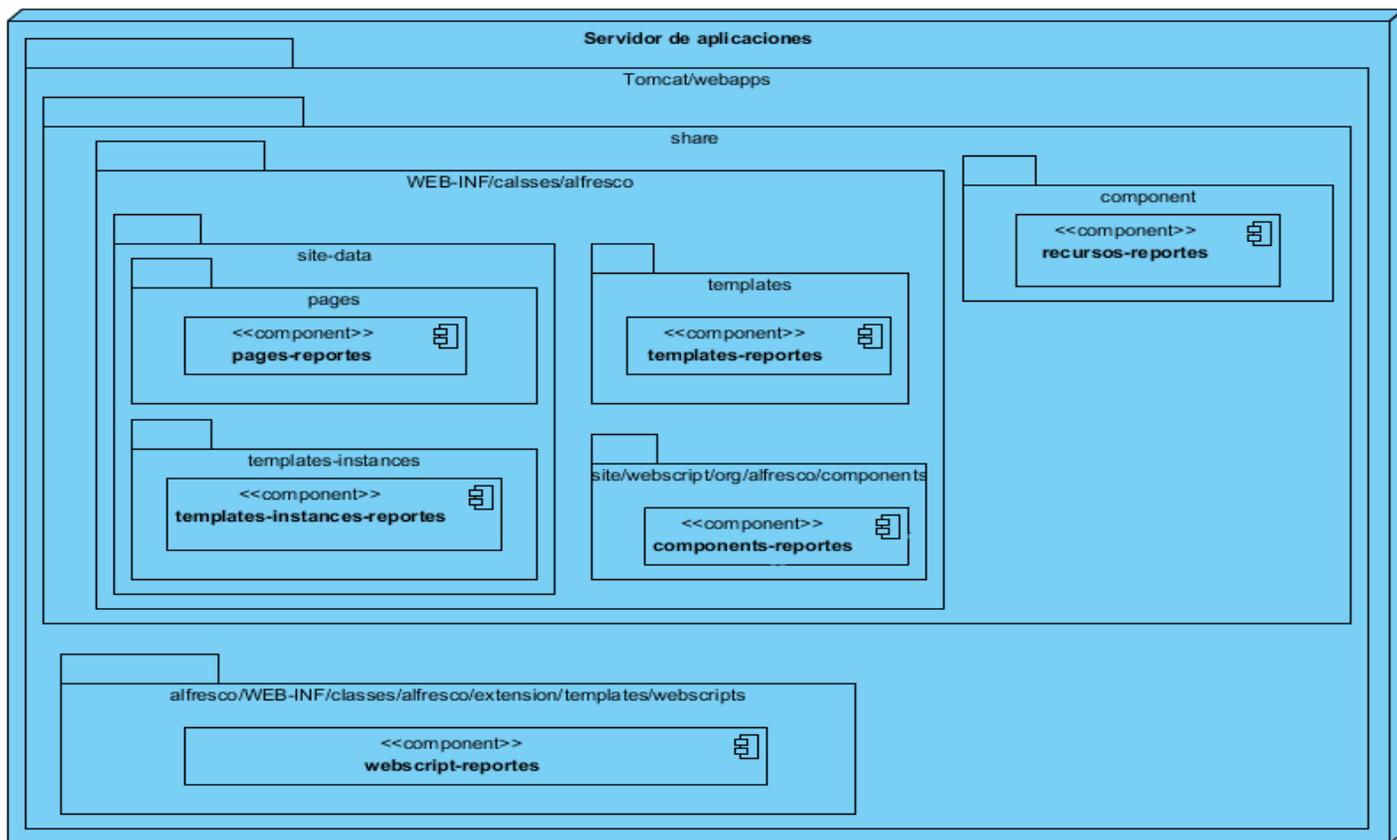


Figura 12 Diagrama de despliegue

Descripción de los Protocolos de Comunicación

Se utiliza la familia de protocolos de comunicación TCP/IP y HTTP debido a que el Módulo Reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2 se desarrolla sobre tecnología Web siguiendo los estándares para la misma.

<<HTTP>>: El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW). HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (cliente, servidor, proxy) para comunicarse.

<<TCP/IP>>: La familia de protocolos de Internet es un conjunto de protocolos de red en la que se basa Internet y que permite la transmisión de datos entre redes de computadoras. Denominada en muchas ocasiones conjunto de protocolos TCP/IP en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos

primeros en definirse, y que son los más utilizados de la familia. El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos sobre redes de área local (LAN) y área extensa (WAN).

3.2. Estándares de codificación

Al programar en un lenguaje en específico se deben seguir reglas. Estas permitirán que cualquier persona que se desempeñe como codificador de dicho lenguaje, pueda interpretar de manera eficiente la escritura del código. Con el objetivo de mejorar la legibilidad del código del Sistema de Reportes, facilitar el mantenimiento y permitir su comprensión rápida por parte de cualquier desarrollador, se siguieron los siguientes estándares de codificación.

Los estándares utilizados son:

- Utilizar una nomenclatura en castellano para el nombre de clases y métodos, sean lo más descriptivo posible. Ejemplo: Listar Reporte, Listar Servidores.
- Los nombres de los métodos, variables y atributos se escribirán en minúscula. Ejemplo: título, insert.
- Los nombres de las clases comenzaran con letra inicial mayúscula. Ejemplo: Reporte.
- Comentar los métodos que puedan resultar de difícil comprensión.

3.3. Pruebas de software

Las pruebas de los software son elementos críticos para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (PRESSMAN R. , 2010). Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema. Básicamente es una fase en el desarrollo de software que consiste en probar la aplicación construida. Se integran dentro de las diferentes fases del ciclo de la Ingeniería de Software ejecutando un programa que mediante técnicas experimentales se trata de descubrir los errores que tiene el sistema. Para determinar el nivel de calidad se deben efectuar las medidas o pruebas que permiten comprobar el grado de cumplimiento respecto a las especificaciones iniciales del producto, siendo el resultado observado y registrado.

Elementos a tener en cuenta para que una prueba tenga éxito (LARA, 2013):

- Estrategia de prueba.
- Niveles de Prueba.

- Tipo de prueba.
- Método de prueba.
- Caso de prueba.

3.3.1. Estrategia de prueba

Las estrategias de pruebas de software proporcionan una guía que describe los pasos que deben realizarse como parte de la prueba, cuándo se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requerirán. Por tanto, cualquier estrategia de prueba debe incorporar la planificación, diseño, ejecución, recolección y evaluación de los casos de prueba (PRESSMAN R. , 2010).

Para realizar una correcta estrategia de pruebas se deben tener en cuenta algunos criterios de importancia (RUIZ, 2015.).

- Describe el enfoque y los objetivos generales de las actividades de prueba.
- Incluye los niveles de prueba a ser diseccionados, el tipo de prueba a ser ejecutada y los casos de prueba diseñados para lograr los objetivos.
- Define:
 - ✓ Técnicas de pruebas (manual o automática) y herramientas a ser usadas.
 - ✓ Criterios de éxitos y culminación de las pruebas.
 - ✓ Consideraciones especiales relacionadas con los recursos necesarios para realizar las pruebas.

3.3.2. Niveles de prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Se distinguen los siguientes niveles de pruebas (RUIZ, 2015.):

- Prueba de desarrollador.
- Prueba independiente.
- Prueba de unidad.
- Prueba de integración.
- Prueba de regresión.
- Prueba de sistema.
- Prueba de aceptación.

Se trabajará en los siguientes niveles:

- En el nivel de desarrollador haciendo uso del método de caja blanca para realizar pruebas del código implementado haciendo uso de la técnica camino básico.
- En el nivel de sistema con el objetivo de probar las funcionalidades del módulo, esto se hará mediante el método de caja negra empleando la técnica de partición de equivalencia.
- Aplicando las pruebas de regresión para verificar que una vez que termine una iteración y se corrijan no conformidades los nuevos cambios no afecten la calidad de lo que se había hecho antes previniendo que se cometan nuevos errores.
- En el nivel de aceptación para ver si el desarrollo del módulo contribuyó o no a la comunicación y colaboración en el Share de Alfresco Community.

Es válido aclarar que la metodología AUP-UCI propone 7 disciplinas (4 ingenieriles y 3 de gestión de proyectos), dentro de las cuales se encuentra la disciplina de prueba que consiste en realizar una evaluación objetiva para garantizar la calidad. Esto incluye la búsqueda de defectos, validar que el sistema funciona tal como está establecido, y verificando que se cumplan los requisitos.

1- Pruebas internas: Se realizan en el centro y se verifica el resultado de la implementación probando cada construcción. Se desarrollan los casos de pruebas. Se realizaron pruebas de caja negra que permiten obtener conjuntos de condiciones de entrada que ejerciten todos los requisitos funcionales para encontrar varios tipos de errores (interfaz, rendimiento, estructura de datos). Se utilizó la técnica de la partición de equivalencia permite examinar los valores válidos y no válidos de las entradas existentes en el software.

2- Pruebas de liberación: Pruebas diseñadas y ejecutadas por una entidad certificadora externa antes de ser entregados al cliente para su aceptación. Se realizaron pruebas de caja negra que verifican que el sistema funcione apropiadamente y sin errores.

3- Pruebas de aceptación: Se verifica que el software está listo y que puede ser usado por usuarios finales. Se hizo entrega del módulo al cliente, y este realizó un acta de aceptación planteando haberse cumplido el objetivo del mismo.

3.3.3. Tipos de prueba

Un grupo de actividades de pruebas pueden tener por objetivo verificar el sistema de software (o parte el sistema) en base a un motivo u objetivo específico para probar. Un tipo de prueba se centra en un objetivo de prueba en particular, que puede ser cualquiera de los siguientes (ISTQB, 2011).

- Una función a realizar por el software.
- Una característica de calidad no funcional, tales como la fiabilidad o la usabilidad.
- La estructura o arquitectura del software o sistema.
- Cambios asociados, es decir, confirmar que se han solucionado los defectos (pruebas de confirmación) y localizar cambios no intencionados (pruebas de regresión).

Puede desarrollarse y/o utilizarse un modelo del software en las pruebas estructurales (por ejemplo, un modelo de flujo de control o un modelo de estructura de menús), en las pruebas no funcionales (por ejemplo, un modelo de rendimiento, un modelo de amenaza de seguridad y un modelo de usabilidad), y en las pruebas funcionales (por ejemplo, un modelo de flujo de procesos, un modelo de transición de estados o una mera especificación de lenguaje).

✓ **Pruebas funcionales**

Las funciones que un sistema, subsistema o componente deben llevar a cabo pueden describirse en productos de trabajo tales como una especificación de requisitos, casos de uso o una especificación funcional, o incluso pueden no estar documentadas. Las funciones son “lo que” el sistema hace.

Las pruebas funcionales se basan en funciones y prestaciones (descritas en documentos o entendidas por los probadores) y en su interoperabilidad con sistemas específicos, y pueden llevarse a cabo en todos los niveles de pruebas.

✓ **Pruebas no funcionales**

Las pruebas no funcionales incluyen, pero sin limitarse a ello, pruebas de rendimiento, pruebas de carga, pruebas de estrés, pruebas de usabilidad, pruebas de mantenibilidad, pruebas de fiabilidad y pruebas de portabilidad. Estas pruebas se refieren a “cómo” funciona el sistema. Las pruebas no funcionales pueden ejecutarse a todos los niveles de prueba. El término pruebas no funcionales hace referencia a las pruebas necesarias para medir las características de los sistemas y software que pueden cuantificarse según una escala variable, tales como los tiempos de respuesta en el caso de las pruebas de rendimiento.

✓ Pruebas estructurales

Las pruebas estructurales (caja blanca) pueden realizarse en todos los niveles de prueba. Las técnicas estructurales son más idóneas de utilizar después de las técnicas basadas en la especificación, para ayudar a medir la exhaustividad de las pruebas mediante una evaluación de la cobertura de un tipo de estructura. La cobertura es la medida en que un juego de pruebas ha probado una estructura, expresada como porcentaje de los elementos cubiertos. Si la cobertura no es del 100%, entonces podrán diseñarse más pruebas para probar los elementos que faltan para aumentar la cobertura.

✓ Pruebas de regresión

Una vez detectado y corregido un defecto, el software debe volver a probarse para confirmar que el defecto original ha sido corregido con éxito. A esto se le denomina confirmación. La depuración (localizar y corregir defectos) es una actividad de desarrollo, no una actividad de pruebas. Las pruebas de regresión son pruebas reiteradas de un programa ya probado, después de haber sido modificado, con vistas a localizar defectos surgidos o no descubiertos como resultado del cambio o de los cambios. Estos defectos pueden estar en el software objeto de las pruebas, o en cualquier otro componente de software asociado o no asociado. Se realizan cuando el software, o su entorno, sufren modificaciones. El alcance de las pruebas de regresión depende del riesgo de no encontrar defectos en el software que antes funcionaba.

El tipo de prueba a aplicar es funcional. Estas pruebas fijan su atención en la validación de las funciones, métodos, servicios y casos de uso. Permiten comprobar el correcto funcionamiento de los requisitos funcionales del módulo.

3.3.4. Métodos de prueba

Los métodos de prueba definen la estrategia a seguir en función de la verificación y validación del sistema diseñado para descubrir fallos. Los métodos estudiados fueron las pruebas de caja blanca y las pruebas de caja negra.

Pruebas de caja negra

Las pruebas de caja negra también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, la prueba de caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (PRESSMAN R. , 2010). Esta prueba intenta encontrar errores de las siguientes categorías: funciones incorrectas o ausentes,

errores de interfaz, errores en estructura de datos o en accesos a bases de datos externas, errores de rendimiento y errores de inicialización y terminación.

Pruebas de caja blanca

Las pruebas de caja blanca, denominadas a veces pruebas de caja de cristal es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (PRESSMAN R. , 2010). Con la aplicación de este método se garantiza que se verifique por lo menos una vez todos los caminos independientes de cada módulo, que se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas, que se ejecuten todos los bucles en sus límites y con sus límites operacionales y se entrenen las estructuras internas de datos para asegurar su validez.

Se hace uso de la técnica del camino básico que permite derivar casos de prueba a partir de un conjunto de caminos independientes por los cuales puede circular el flujo de control.

Pasos para el desarrollo de la técnica del Camino básico:

1. Se escoge el diseño o el código de la aplicación como base, y se dibuja el correspondiente grafo de flujo. Se obtiene la complejidad ciclomática del grafo de flujo.
2. Se determina la complejidad ciclomática del grafo resultante, posteriormente se determinan los casos de pruebas que permitan la ejecución de los caminos anteriores.
3. Se establece un conjunto básico de caminos linealmente independientes.
4. Se elaboran los casos de pruebas que permitirán la ejecución de cada camino del conjunto básico.

Notación del grafo de flujo:

- **Nodo del grafo de flujo(N):** Es un círculo que representa las sentencias procedimentales.
- **Aristas(A):** Son las flechas o enlaces que representan el flujo de control del grafo.
- **Regiones:** Son las áreas delimitadas por las aristas y nodos. Al realizar el conteo de las regiones también se incluye el área exterior del grafo.
- **Complejidad ciclomática:** Se calcula a partir del grafo de control del software bajo análisis y mide, a grandes rasgos, el número de caminos independientes que pueden ocurrir en la ejecución, entre el inicio y el fin del software.

Se define como: $V(G) = A - N + 2$.

A continuación, se muestra un fragmento de código perteneciente al algoritmo desarrollado:

```

9   try {
10      var uuid = json.get('uuid');
11      var nodo = getNodo(uuid);
12      var tituloConsulta = nodo.properties['rep:tituloConsulta'];
13      var parametrosConsulta = nodo.properties['rep:parametrosConsulta'];
14      var nombreparametrosConsulta = nodo.properties['rep:nombreparametrosConsulta'];
15      var servidorConsulta = "";
16      var nombreServidor = "";
17      var dirServidorConsulta = "";
18      if (nodo.assocs['rep:assocConsulta-Servidor'] != null) {
19         servidorConsulta = nodo.assocs['rep:assocConsulta-Servidor'][0].id;
20         nombreServidor = nodo.assocs['rep:assocConsulta-Servidor'][0].properties["rep:nombreServidor"];
21         dirServidorConsulta = nodo.assocs['rep:assocConsulta-Servidor'][0].properties["rep:direccionServidor"];
22      }
23
24      var flConsulta = nodo.properties['rep:flConsulta'];
25      var nombreflConsulta = nodo.properties['rep:nombreflConsulta'];
26
27      model.tituloConsulta = tituloConsulta;
28      model.parametrosConsulta = parametrosConsulta;
29      model.nombreparametrosConsulta = nombreparametrosConsulta;
30      model.servidorConsulta = servidorConsulta;
31      model.dirServidorConsulta = dirServidorConsulta;
32      model.flConsulta = flConsulta;
33      model.nombreflConsulta = nombreflConsulta;
34      model.nombreServidor = nombreServidor;
35
36   } catch (e) {
37      model.error = e.message;
38   }

```

Figura 13 Fragmento de código desarrollado

Posteriormente se procede a la elaboración del grafo de flujo teniendo en cuenta dicha enumeración.

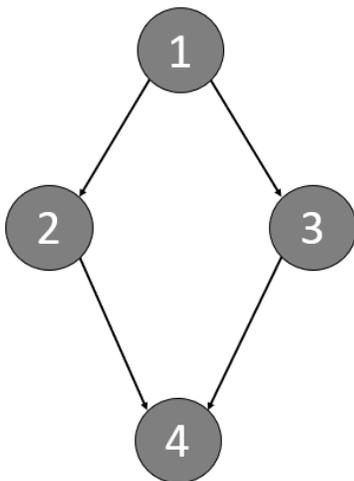


Figura 14 Representación del grafo de flujo de camino básico de aplicar matriz de concurrencia.

Complejidad ciclomática

$$V(G) = A - N + 2$$

$$V(G) = 4 - 4 + 2 = 2$$

Caminos independientes

1, 2, 4

1, 3, 4

Cada camino independiente es un caso de prueba a realizar, de forma que se garantiza que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. En el caso anterior se calcularon 2 caminos básicos, por tanto surge la necesidad de hacer igual número de casos de prueba. A continuación se muestra uno de los casos de pruebas realizados.

Tabla 9 Casos de prueba de caja blanca para los caminos básicos

Caso de prueba 1 para el camino básico	
Descripción Mostrar un error si no se puede agregar un nuevo servidor.	
Condición de ejecución No se agrega el nuevo servidor.	
Datos de entrada	
Tipo de datos esperado	
Evaluación del caso de Prueba: Satisfactoria	
Caso de prueba 2 para el camino básico	
Descripción Se logra agregar un nuevo servidor.	
Condición de ejecución Se agrega un nuevo servidor.	
Datos de entrada	Nombre, dirección.
Tipo de datos esperado	Nodo servidor
Evaluación del caso de Prueba: Satisfactoria	

3.3.5. Casos de prueba

Un caso de prueba es una especificación de un caso para probar el sistema, incluyendo qué probar, con qué entradas y resultados y bajo qué condiciones. Su principal objetivo es obtener un conjunto de pruebas que tengan una mayor probabilidad de descubrir los defectos del software.

Se utiliza el método de caja negra aplicando la técnica partición de equivalente, la que permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase

de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico (JACOBSON, 2004).

Se describieron los casos de pruebas para cada caso de uso, especificando el escenario, la descripción, las variables, la respuesta del sistema y el flujo central, así como los resultados obtenidos una vez ejecutado el caso de prueba y las condiciones que deben cumplirse para que este se ejecute.

Tabla 10 Diseño de caso de prueba del caso de uso "Gestionar Servidor"

Escenario	Descripción	Nombre	Dirección	Respuesta del sistema	Flujo Central
EC1.1 Agregar Servidor	El escenario de prueba permite agregar un nuevo servidor solr.	V	V	El sistema agrega un nuevo servidor. Se muestra el mensaje: "El servidor ha sido creado satisfactoriamente."	1. Se selecciona la opción Gestionar Servidor. 2. Inserta los datos. 3. Selecciona el botón Agregar en la parte inferior del formulario.
		V	V		
EC1.2 Campos obligatorios vacíos.	El escenario de prueba permite al usuario validar los campos vacíos.	I	I	El sistema señala los campos obligatorios vacíos y muestra el mensaje: "El valor no puede estar vacío."	1. Se selecciona la opción Gestionar Servidor. 2. Selecciona el botón

					Aceptar que aparece en la parte inferior del formulario.
EC1.3 Validar campos incorrectos.	El escenario de prueba permite al usuario validar los campos del formulario.	I	I	El sistema muestra el mensaje: "Solo acepta los valores a-z, A-Z, 0-9, /.".	<p>1. Se selecciona la opción Gestionar Servidor.</p> <p>2. Selecciona el botón Aceptar que aparece en la parte inferior del formulario.</p>

Tabla 11 Diseño de caso de prueba del caso de uso "Gestionar Consulta"

Escenario	Descripción	Título	Servidor	Parámetros	Nombre de parámetros	FL	Nombre de FL	Respuesta del sistema	Flujo Central
EC1.1	El escenario	V	V	V	V	V	V	El sistema agrega una	1. Se selecciona

Agregar Consulta	o de prueba permite agregar una nueva consulta solr.	V	V	V	V	V	V	nueva consulta. Se muestra el mensaje: "La consulta ha sido creada satisfactoriamente."	la opción Gestionar Consulta. 2. Inserta los datos. 3. Selecciona el botón Agregar en la parte inferior del formulario.
EC1.2 Campos obligatorios vacíos.	El escenario de prueba permite al usuario validar los campos vacíos.	I	I	I	I	I	I	El sistema señala los campos obligatorios vacíos y muestra el mensaje: "El valor no puede estar vacío."	1. Se selecciona la opción Gestionar Consulta. 2. Selecciona el botón Aceptar que aparece en la parte inferior del formulario.
EC1.3 Validar	El escenario	I	I	I	I	I	I		

campos incorrectos.	o de prueba permite al usuario validar los campos del formulario.							mensaje: "Solo acepta los valores a-z, A-Z, 0-9, _".	la opción Gestionar Consulta. 2. Selecciona el botón Aceptar que aparece en la parte inferior del formulario.
---------------------	---	--	--	--	--	--	--	--	--

Tabla 12 Diseño de caso de prueba del caso de uso "Buscar Consulta"

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC1.1 Búsqueda válida	El escenario permite buscar una consulta.	El sistema muestra los resultados de la búsqueda de la consulta.	1. Se selecciona la opción Gestionar Consulta. 2. Se introduce el criterio de búsqueda.
EC1.2 Búsqueda inválida	El escenario permite buscar una consulta.	El sistema no muestra los resultados de la búsqueda.	1. Se selecciona la opción Gestionar Consulta. 2. Se introduce el criterio de búsqueda.

Tabla 13 Diseño de caso de prueba del caso de uso "Buscar Servidor"

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC1.1 Búsqueda válida	El escenario permite buscar un servidor.	El sistema muestra los resultados de la búsqueda del servidor.	1. Se selecciona la opción Gestionar Servidor. 2. Se introduce el criterio de búsqueda.
EC1.2 Búsqueda inválida	El escenario permite buscar un servidor.	El sistema no muestra los resultados de la búsqueda.	1. Se selecciona la opción Gestionar Servidor. 2. Se introduce el criterio de búsqueda.

Tabla 14 Diseño de caso de prueba del caso de uso "Buscar Reporte"

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC1.1 Búsqueda válida	El escenario permite buscar un reporte.	El sistema muestra los resultados de la búsqueda del reporte.	1. Se selecciona la opción Reportes Guardados. 2. Se introduce el criterio de búsqueda.
EC1.2 Búsqueda inválida	El escenario permite buscar un reporte.	El sistema no muestra los resultados de la búsqueda.	1. Se selecciona la opción Reportes Guardados. 2. Se introduce el criterio de búsqueda.

Tabla 15 Diseño de caso de prueba del caso de uso "Exportar Reporte a .pdf"

Escenario	Descripción	Respuesta del sistema	Flujo Central
EC1.1 Exportar reporte a formato .pdf	El escenario permite exportar un reporte a formato .pdf.	El sistema exporta el reporte deseado a pdf.	<ol style="list-style-type: none"> 1. Se selecciona la opción Reportes Guardados. 2. Se selecciona la opción ver detalles del servidor deseado. 3. Se selecciona la opción exportar a pdf.

Tabla 16 Diseño de caso de prueba del caso de uso "Gestionar Reporte"

Escenario	Descripción	Título Reporte	Nombre Reporte	Respuesta del sistema	Flujo Central
EC1.1 Agregar Reporte	El escenario de prueba permite agregar un nuevo reporte.	V	V	El sistema agrega un nuevo reporte. Se muestra el mensaje: "El reporte ha sido creado satisfactoriamente."	<ol style="list-style-type: none"> 1. Se selecciona la opción Gestionar Reporte. 2. Inserta los datos.

		V	V		3. Selecciona el botón Agregar en la parte inferior del formulario.
EC1.2 Campos obligatorios vacíos.	El escenario de prueba permite al usuario validar los campos vacíos.	I	I	El sistema señala los campos obligatorios vacíos y muestra el mensaje: "El valor no puede estar vacío."	1. Se selecciona la opción Gestionar Reporte. 2. Selecciona el botón Aceptar que aparece en la parte inferior del formulario.
EC1.3 Validar campos	El escenario de prueba	I	I	El sistema muestra el mensaje: "Solo	1. Se selecciona la

incorrectos.	permite al usuario validar los campos del formulario.			acepta los valores a-z, A-Z, 0-9, _,".	opción Gestionar Reporte. 2. Selecciona el botón Aceptar que aparece en la parte inferior del formulario.
--------------	---	--	--	--	--

Conclusiones del capítulo

A partir de los resultados obtenidos en este capítulo se logró implementar el módulo de reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2. El diagrama de despliegue permitió analizar la forma en que se comunicarán los componentes del módulo. La utilización de un estándar de codificación permitió homogenizar el código generado y posibilitó que la herramienta sea entendible por otros programadores. Las pruebas de sistema realizadas permitieron la verificación de funcionalidades y la validación de parte de los requisitos no funcionales del sistema.

CONCLUSIONES GENERALES

Para dar cumplimiento al objetivo general de este trabajo y a la problemática planteada, se realizaron con éxito las tareas trazadas en el comienzo del mismo, llegando a las siguientes conclusiones:

1. Se elaboró el marco teórico conceptual en lo referente al desarrollo del módulo reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.
2. Se analizaron sistemas homólogos para conocer aspectos regulares en el diseño del módulo reportes de XABAL Repxos en la tecnología del Share de Alfresco Community 5.2.
3. Se analizaron las herramientas informáticas y metodologías de desarrollo de software asumidas por el proyecto para realizar la implementación del módulo reportes sobre la tecnología del Share de Alfresco Community 5.2.
4. El uso de la metodología RUP permitió documentar todo el ciclo de desarrollo del módulo reportes sobre la tecnología del Share de Alfresco Community 5.2.
5. Se identificaron las principales funcionalidades del sistema para la posterior implementación del mismo, teniendo en cuenta los requisitos definidos anteriormente.
6. Se realizaron los diferentes tipos de pruebas de software comprobando así el correcto desarrollo del mismo y que este cumplía con los requisitos definidos.

RECOMENDACIONES

Teniendo en cuenta el estudio realizado durante todo el proceso de desarrollo de la herramienta y en aras de enriquecer la solución, el autor recomienda:

- 1- Incluir más metadatos (dubling core u otros) en el componente de agregación de parámetros y fl perteneciente a la creación de consultas, con el fin de aumentar las posibilidades de generar mayor cantidad de consultas para la obtención de reportes.

REFERENCIAS BIBLIOGRÁFICAS

1. Alegsa, Leandro. Definición de JavaScript. [online]. 2016. Definición de JavaScript. [En línea] 2016. <http://www.alegsa.com.ar/Dic/javascript.php>.
2. Alvarez, M. A. (2010). "Manual de jQuery." Recuperado el 17.
3. Barrueco, J. M. and C. García Testal (2009). "Repositorios institucionales universitarios: evolución y perspectivas." Zaragoza: 99-107.
4. Bermúdez, J. and D. Fardy (2014). "Implementación de un sistema de gestión documental utilizando la norma ISO 15489 y plataforma Alfresco Community."
5. Cillero, Manuel. 2019. [En línea] 2019. <https://manuel.cillero.es/doc/metrica-3/tecnicas/diagrama-de-paquetes/>
6. Duperet Cabrera, E., et al. (2015). "Importancia de los repositorios para preservar y recuperar la información." MEDISAN 19(10): 3080-3087.
7. Hernández Pérez, A. and M. A. García Moreno (2013). "Datos abiertos y repositorios de datos: nuevo reto para los bibliotecarios."
8. Rodríguez Cruz, Y., et al. (2016). "Gestión documental, de información, del conocimiento e inteligencia organizacional: particularidades y convergencia para la toma de decisiones estratégicas." Revista Cubana de Información en Ciencias de la Salud 27(2): 206-224.
9. Soares Guimarães, M. a. C., et al. (2012). "Los repositorios temáticos en la estrategia de la iniciativa Open Access." Nutrición Hospitalaria 27(2): 34-40.
10. Solis, J. (2014). "¿ Qué es Bootstrap y cómo funciona en el diseño Web." Obtenido de: [http://www.arweb.com/chucherias/editorial/% C2% BFque-es-bootstrap-y-comofunciona-en-el-diseno-web. htm](http://www.arweb.com/chucherias/editorial/%C2%BFque-es-bootstrap-y-comofunciona-en-el-diseno-web.htm).
11. Suarez, Michel David. 2011. repositorio institucional uci. Propuesta arquitectónica para el desarrollo del Gestor de Documentos Administrativos eXcriba 2.0 [En línea] 2011. http://repositorio_institucional.uci.cu/jspui/bitst.
12. Definicion.de. 2014. Definicion de. [En línea] 2014. <http://definicion.de/reporte>.

13. Felipe. 2015. Real Academia Española. Diccionario de la lengua española. [En línea] 5 de Junio de 2015. <http://lema.rae.es/drae/?val=repositorio>.
14. GeoDir 2017. [En línea] 2019. <https://geodir.wordpress.com/2017/03/06/definicion-y-utilizacion-del-html5-css3/>
15. ISTQB, colectivo de autores. 2011. International Software Testing Qualification Boards Probador Certificado Programa de estudio de nivel básico. 2011.
16. IV, Felipe. Real Academia Española. Diccionario de la lengua española. [En línea] [Citado el: 5 de Junio de 2015.] <http://lema.rae.es/drae/?val=repositorio>.
17. JACOBSON, Ivar, BOOCH, Grady and RUMBAUGH, James. 2004. El proceso unificado de desarrollo de software. s.l. : Félix Varela, 2004.
18. Lamarca, María Jesús. 2013. Hipertexto. [En línea] 2013. <http://www.hipertexto.info/documentos/uml.htm>.
19. LARA, M. 2013. Pruebas de software. 2013. págs. 1-34.
20. LARMAN, C. 2003. Modelo de Casos de Uso: Escritura de Requisitos en Contexto. 2003.
21. Lornel Rivas, Maria Perez, Luis Eduardo Mendoza Morales, Anna Grimán. 2014. Herramientas de Desarrollo de Software: Hacia la. Caracas - Venezuela. : Laboratorio de Investigación en Sistemas de Información (LISI), Departamento de Procesos y Sistemas, Universidad Simón Bolívar, 2014.
22. O'Reilly Media. 2016. Inc. XML.com. [En línea] 2016. <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>.
23. Paradigm, Visual. 2013. Visual paradigm for uml. Visual Paradigm for UML-UML tool for software application development. 2013.
24. PRESSMAN, R.S. 2010. Ingeniería del software. Un enfoque práctico. 2010.
25. PRÓMETA, INFANTE. 2009. Sistema de Gestión de Reportes Dinámicos. 2009.
26. RUIZ, B y VAILLANT, C. 2015.. Módulo Recomendaciones del sistema para repositorios digitales REPXOS 3.0. Universidad de las Ciencias Informáticas. 2015.

27. Scientec. (2016, octubre 17). Scientec. Retrieved from <https://www.scientec.com.mx/axure-rp/>
28. UCI. 2016. 2016.

BIBLIOGRAFÍA

1. About Dspace. DSPACE. [En línea] 2015. [Citado el: 20 de febrero de 2015.] <http://www.dspace.org/introducing>.
2. Alegsa, Leandro. Definición de JavaScript. [online]. 2016. Definición de JavaScript. [En línea] 2016. <http://www.alegsa.com.ar/Dic/javascript.php>.
3. Alfresco. 2008. Alfresco. [En línea] 2008. <https://docs.alfresco.com/4.1/>
4. Alfresco, un gestor documental de código abierto [En línea] <https://www.ticportal.es/temas/sistema-gestion-documental/programas-gestion-documental/alfresco>
5. Alfresco Share. 2015. Alfresco Share. [En línea] 2015. <https://community.alfresco.com/docs/DOC-5672-alfresco-share>
6. ALFRESCO SOFTWARE, I. 2015. Alfresco Community Edition. 2015.
7. Amerikati. Generador de reportes. [En línea] 2012. http://www.amerikati.com/?page_id=364.
8. Ant, Apache. 2016. Apache Ant. [En línea] 2016. <http://ant.apache.org/>
9. BARATZ, C. 2015. Los 10 beneficios de la gestión documental en las organizaciones. Gestión documental en las organizaciones. 2015.
10. CAMPILLO TORRES, Irima. 2010. Sistema de Gestión Integral de Documentos de archivo para empresas de la construcción del territorio de Camagüey. Granada: Universidad de Granada, 2010
11. David Suarez, Michel. 2011. repositorio institucional uci. Propuesta arquitectónica para el desarrollo del Gestor de Documentos Administrativos eXcriba 2.0 [En línea] 2011. http://repositorio_institucional.uci.cu/jspui/bitst.
12. DSpace Release 3.0 Notes, creado por Tim Donohue, modificado por última vez en mayo 08, 2015 [En línea] <https://wiki.duraspace.org/display/DSPACE/DSpace+Release+3.0+Notes>
13. DSpace Release 6.0 Status, creado por Hardy Pottinger, modificado por última vez por Terry Brady en noviembre 09, DSpace 2016 [En línea] <https://wiki.duraspace.org/display/DSPACE/DSpace+Release+6.0+Status>

14. DSpace Community, 2019 [En línea] <https://duraspace.org/dspace/>
15. Flanagan, D. JavaScript. La Guía Definitiva. Madrid : s.n., 2007. ISBN: 978-84-415-2202-2 84-415-2202-2.
16. Gestión documental, de información, del conocimiento e inteligencia organizacional: particularidades y convergencia para la toma de decisiones estratégicas. Yunier Rodríguez Cruz, Antonieta Castellanos Crespo, Zaylí Ramírez Peña. 2016. La Habana : Revista Cubana de Información en Ciencias de la Salud, 2016.
17. Hablando de DSpace, version 6.0 [En línea] <http://www.arvo.es/dspace/tag/version-6>
18. Javier Eguiluz. Introducción a CSS. Uniwebsidad [En línea]. <https://uniwebsidad.com/libros/css>
19. Juan Peláez. Arquitectura basada en capas [En línea] <https://geeks.ms/jkpelaez/2009/05/30/arquitectura-basada-en-capas/>
20. Maida, Esteban Gabriel y Pacienza, Julián. 2015. Metodologías de desarrollo de software. Argentina : Biblioteca digital de la universidad católica de Argentina, 2015.
21. Mtorres. Tecnicas de Prueba.
22. Manual de Bootstrap 2015 [En línea] <https://getbootstrap.com/2.3.2/#transitions>
23. Modelo vista controlador (MVC). Servicio de Informática ASP.NET MVC 3 Framework [En línea] <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
24. LARA, M. 2013. Pruebas de software. 2013. págs. 1-34.
25. LARMAN, C. 2003. Modelo de Casos de Uso: Escritura de Requisitos en Contexto. 2003.
26. Lornel Rivas, Maria Perez, Luis Eduardo Mendoza Morales, Anna Grimán. 2014. Herramientas de Desarrollo de Software: Hacia la. Caracas - Venezuela. : Laboratorio de Investigación en Sistemas de Información (LISI), Departamento de Procesos y Sistemas, Universidad Simón Bolívar, 2014.
27. OBS, B.S. 2018. *¿Cuál es la utilidad de la matriz de trazabilidad?* | OBS Business School. 2018.

28. O'Reilly Media. 2016. Inc. XML.com. [En línea] 2016. <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58>.
29. Paradigm, Visual. 2013. Visual paradigm for uml. Visual Paradigm for UML-UML tool for software application development. 2013.
30. PRESSMAN, Roger S. 2005. Ingeniería de Software. Un enfoque práctico. sexta. Mc Graw Hill,. México : s.n., 2005.
31. Sanchez, T.R. 2015. Metodología de desarrollo para la Actividad productiva de la UCI. 2015.
32. SISTEMA DE GESTIÓN INTEGRAL DE DOCUMENTOS, 2015. Sistema de gestión integral de documentos. Almería. Sánchez, T.R. 2015. *Metodología de desarrollo para la Actividad productiva de la UCI*. 2015.
33. Sommerville, Ian. Software Engineering. 2006. ISBN 84 7829 074 5.
34. POTTS, J., 2008. Alfresco developer guide. S.l.: Packt Publishing Ltd. ISBN 1-84719-312-9.
35. REPORTES DINÁMICOS GESTIÓN 22. REPORTES DINÁMICOS. [En línea] 2011. <http://gestion22.cl/reportes-dinamicos-2/>.
36. Repositorio Digital EPN. Repositorio Digital EPN. [En línea] 2010. [Citado el: 14 de diciembre de 2014.] <http://bibdigital.epn.edu.ec/handle/15000/2615>.
37. Repositorio Digital Universidad Técnica del Norte. [En línea] 2010. <http://repositorio.utn.edu.ec>.
38. Scientec. (2016, octubre 17). Scientec. Retrieved from <https://www.scientec.com.mx/axure-rp/>
39. Serie Científica de la Universidad de Ciencias Informáticas [En línea] <https://publicaciones.uci.cu/?journal=serie>
40. Sommerville, Ian. Software Engineering. 2006. ISBN 84 7829 074 5.
41. UNAD. 2018. *Diagrama de Clases de Diseño | LENGUAJE DE MODELADO UNIFICADO - UML*. 2018.

ANEXOS

Entrevista

La entrevista se realizó de manera informal. Las preguntas fluyeron en dependencia del conocimiento del cliente sobre el módulo implementado.

1. ¿Qué requisitos debe tener el módulo reportes de XABAL Repxos en la tecnología del Share de Alfresco 5.2?
2. ¿Cuál es el objetivo de implementar este módulo?
3. ¿Qué datos se manejan en el módulo?
4. ¿Quién define los usuarios que deben acceder al área?
5. ¿Qué tipo de reportes se deben generar?
6. ¿Cómo garantizar la seguridad en el módulo?

Entrevista realizada a:

Director del centro CIGED: Aurelio Antelo Collado.