Centro de Informatización Universitaria

Facultad 1

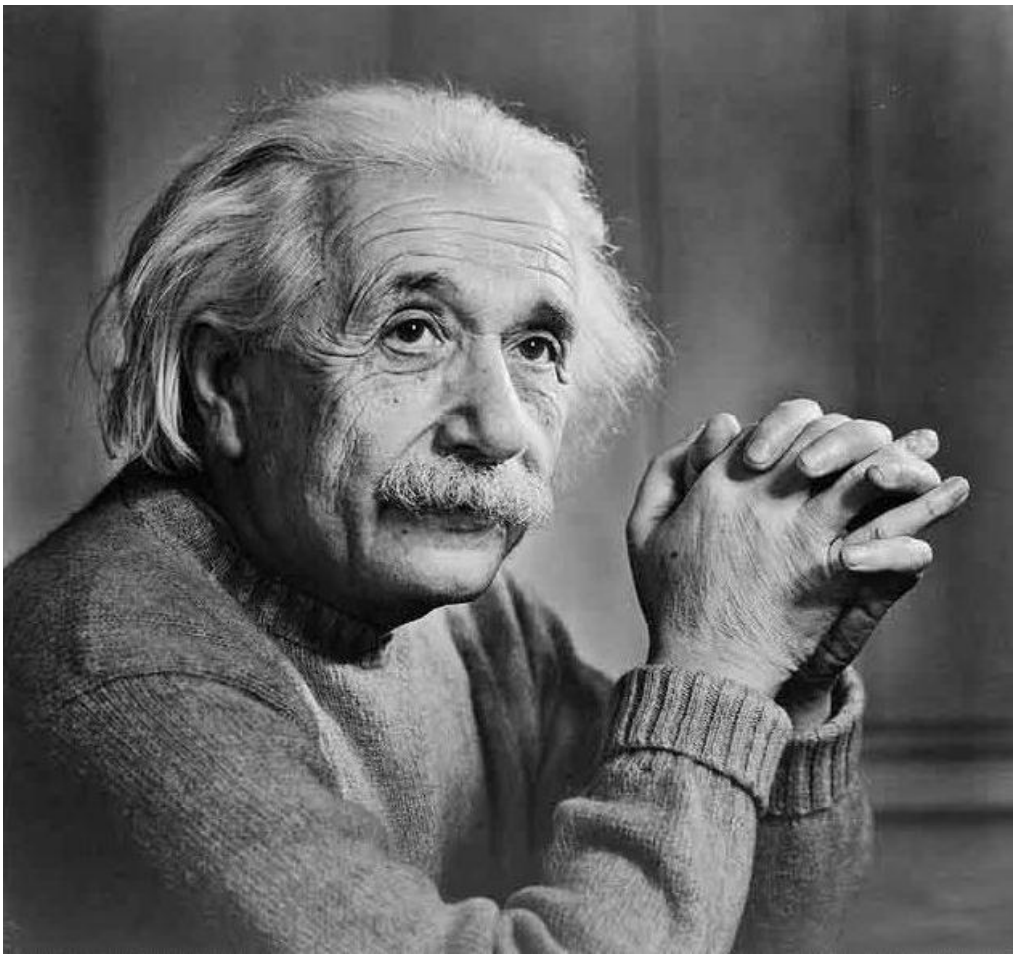
Administración de usuarios desde el cliente web del Gestor de Documentos Administrativos eXcriba

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

Autor(a): Marisol Busutil Expósito

Tutor(es): Ing. Dayelis Blanco Hernández
Ing. Lizandra Candelario Rodríguez
Ing. Reinier Elejalde Chacon

La Habana, Cuba, 2013



"Si quieres vivir una vida feliz, átala a una meta, no a una persona o a un objeto."
Albert Einstein

Declaración de autoría

Declaro que soy el único autor de este trabajo y autorizo al Centro de Informatización Universitaria de la Universidad de las Ciencias Informáticas, para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los ___ días del mes de junio del año 2013.

Firma del autor

Marisol Busutil Expósito

Firma del tutor

Dayelis Blanco Hernández

Firma del tutor

Lizandra Candelario Rodríguez

Firma del tutor

Reinier Elejalde Chacon

Agradecimientos

Agradezco a mis padres, por ser un ejemplo para mí, demostrándome día a día que para alcanzar cualquier sueño hay que perseverar y sacrificarse, les agradezco por estar siempre dispuestos a escucharme, por su apoyo incondicional, les agradezco que se aventuraran conmigo en esta travesía y me siento muy feliz que hoy podamos sentirnos orgullosos de haberlo logrado.

Agradezco a mi hermano por su cariño y comprensión, por sentirse orgulloso de mí y por su apoyo incondicional en todo momento.

A mi prima Ailene y mi tia Marisol por la ayuda y apoyo que me brindaron.

Agradezco a mis tutoras Lisandra y Dayelis por su ayuda y buenos consejos. Agradezco especialmente a mi tutor Reinier por brindarme todo el tiempo su apoyo incondicional, por no dejarme nunca para después, por ser además de mi tutor, mi amigo, con el que he podido compartir todo tipo de experiencias, todo tipo de comentario, gracias por ser tan buena persona y por ser mi cómplice en este sueño.

Agradezco a mi gran amigo Hector al cual conocí desde el primer día que entre en esta universidad y del cual no me he despegado nunca más. Le agradezco mucho que me ayudara incondicionalmente desde ese día hasta hoy.

Agradezco a Dany por sus incontables consejos, por sus críticas constructivas para que todo saliera bien, por apoyarme siempre que lo he necesitado, le agradezco además por demostrarme que nada es imposible, por siempre pretender hacer de mí una mejor persona y permitirme ver la vida de otra forma, por hacerme reír, por su compañía y por su cariño incondicional.

Agradezco a mis amistades Yaima y Yinet por la amistad que me han brindado, a ale por ser mi compañero de proyectos, por las palabras de aliento en contra de los nervios antes de cada exposición, por los dibujos y los origamis que me regalaba para ponerme feliz. A Marbe y Yanet, por apoyarme en este tiempo. A todos los han aportado su granito de arena en mi formación les doy las gracias.

Marisol Busutil Expósito

Dedicatoria

Dedico esta tesis a las personas más importantes en mi vida, mis padres y mi hermano, por impulsarme a seguir adelante, por su amor incondicional y su entrega, por los incontables consejos y por todo su amor.

A mi abuela Edelia por cuidarme tan bien en mis primeros años de vida y a papá Reinaldo, por las enseñanzas que me dio durante todo ese tiempo.

A mi tía Edilia, en donde quiera que esté, por todo el tiempo que estuvo conmigo, por su dedicación y entrega.

Marisol Busutil Expósito

Resumen

El Gestor de Documentos Administrativos (GDA) eXcriba es un sistema de gestión documental cuyo objetivo fundamental es gestionar los documentos de una empresa de manera eficaz. Dicho sistema hace uso del repositorio de contenidos y servicios que provee el Gestor de Contenido Empresarial Alfresco. La administración de usuarios y grupos en este sistema se realiza a través de Alfresco Explorer, proyecto que se distribuye con Alfresco. Sin embargo, en diferentes despliegues del sistema eXcriba se han identificado un conjunto de deficiencias que inciden en la calidad del producto y por consiguiente en la aceptación de los clientes y usuarios.

A raíz de esta problemática, se identificó la necesidad de independizar del proyecto Alfresco Explorer la administración de los usuarios y grupos de usuarios del GDA eXcriba. En aras de solucionar esta situación se decidió implementar un módulo que permitiera administrar los usuarios y grupos de usuarios desde la propia interfaz *web* del Gestor de Documentos Administrativos eXcriba. Para el desarrollo de este módulo se seleccionaron un conjunto de herramientas, tecnologías y lenguajes de programación que facilitaran el diseño y construcción del mismo. Además, se desarrollaron y consumieron un conjunto de servicios *web* para la comunicación con el repositorio de contenidos desde el cliente *web* de eXcriba. A lo largo del presente documento se irán presentando los principales artefactos y resultados que se fueron obteniendo durante el proceso de desarrollo de la solución.

Palabras clave: gestión de usuarios y grupos, eXcriba, usuarios.

Tabla de contenido

Introducción 1
Capítulo 1.Conceptos fundamentales sobre la administración de usuarios 6
1.1 Preliminares 6
1.2 Administración de identidades 6
1.2.1 Componentes de la administración de identidades 7
1.2.2 Ventajas de la administración de identidades 8
1.2.3 Desventajas de la administración de identidades 8
1.3 Administración de cuentas de usuario 9
1.3.1 El nombre de usuario 9
1.3.2 Tipos de cuentas de usuario 9
1.3.3 Ciclo de vida de las cuentas de usuario 10
1.4 Administración de usuarios en los sistemas de gestión documental 11
1.4.1 Nuxeo DM 11
1.4.2 KnowledgeTree 12
1.4.3 OpenKM 14
1.4.4 Alfresco Explorer 15
1.4.5 AlfrescoUserGroupManager 15
Resultados del estudio 17
1.5 El proceso de desarrollo de *software* 17
1.5.1 Metodología de desarrollo 17
1.5.2 Lenguaje de modelado 18
1.5.3 Herramienta para el modelado 18
1.6 Lenguajes de programación, herramientas y tecnologías 19
1.6.1 Tecnologías para el desarrollo de las interfaces de usuario 19
1.6.2 Tecnologías para el desarrollo de la lógica de negocio 21
1.6.3 Tecnologías para la comunicación con el repositorio de contenidos 21
1.6.4 Tecnologías para el desarrollo de los servicios *web* 22
Capítulo 2. Propuesta de solución 24
2.1 Análisis crítico de la situación problemática 24
2.2 Modelo de Dominio 27
2.3 Especificación de los requisitos del software 28
2.3.1 Técnicas para la captura de los requisitos 28
2.3.2 Requerimientos funcionales 30

2.3.3 Requerimientos no funcionales 32
2.3.4 Técnicas de Validación de Requisitos 32
2.4 Captura de requisitos como casos de uso 33
2.4.1 Definición de los actores 33
2.4.2 Diagrama de caso de uso 34
2.4.2 Validación de requerimientos: Matriz de trazabilidad 34
2.4.3 Descripción de los casos de uso 36
Capítulo 3. Diseño de la solución 40
3.1 Arquitectura del sistema 40
3.1.1 Análisis general de la Arquitectura del GDA eXcriba 41
3.1.2. Diseño de la arquitectura del módulo 43
3.1.3 Relación entre los componentes del módulo 44
3.2 Patrones de diseño 46
3.2.1 Experto 46
3.2.2 Controlador 47
3.3 Diseño del caso de uso gestionar cuentas de usuario 47
3.3.1 Diagrama de clases del diseño 48
3.3.2 Descripción de las clases del diseño 48
3.3.3 Diagramas de colaboración del diseño 53
Capítulo 4. Implementación y Prueba de la Solución 55
4.1 Implementación del módulo 55
4.1.1 Diagrama de componentes 55
4.1.2 Diagrama de despliegue 56
4.2 Prueba 58
4.2.1 Resultados de las pruebas realizadas 62
Conclusiones 63
Recomendaciones 64
Referencias Bibliográficas 65
Bibliografía 70
Glosario de Términos 75
Anexos 76

Índice de tablas

Tabla 1 Definición de los actores del sistema 33
Tabla 2 Matriz de trazabilidad 35

Tabla 3 Descripción del caso de uso Gestionar cuentas de usuarios 39
Tabla 4 Descripción de la clase UI_Gestionar_Usuarios 49
Tabla 5 Descripción de la clase Crear_Cuenta..... 49
Tabla 6 Descripción de la clase Eliminar_Cuenta..... 49
Tabla 7 Descripción de la clase Modificar_Cuenta 50
Tabla 8 Descripción de la clase Gestionar_Usuarios..... 50
Tabla 9 Descripción de la clase Wscript_Authorities 51
Tabla 10 Descripción del servicio “Crear cuenta de usuario” 52
Tabla 11 Descripción del servicio “Eliminar cuenta de usuario” 52
Tabla 12 Descripción del servicio “Obtener cuenta de usuario” 52
Tabla 13 Descripción del servicio “Modificar cuenta de usuario..... 53
Tabla 14 CP1_CU Gestionar cuentas de usuario. Crear cuentas de usuario..... 61
Tabla 15 CP1_CU Gestionar cuentas de usuario. Modificar cuentas de usuario 61
Tabla 16 CP1_CU Gestionar cuentas de usuario. Eliminar cuentas de usuario..... 62
Tabla 17 Resultado de las pruebas 62
Tabla 18 Especificación del caso de uso gestionar grupo de usuarios. 78
Tabla 19 Descripción de la clase UI_Gestionar_Usuarios 79
Tabla 20 Descripción de la clase Crear_Grupo 80
Tabla 21 Descripción de la clase Eliminar_Grupo 80
Tabla 22Descripción de la clase Gestionar_Usuarios..... 81
Tabla 23 Descripción de la clase Wscript_Authorities 81
Tabla 24 Descripción del servicio “Crear grupo de usuario” 82
Tabla 25 Descripción del servicio “Eliminar grupo de usuario” 82

Índice de figuras

Figura 1 Descripción del ciclo de vida de las cuentas de usuario 10
Figura 2 Diagrama de Modelo del dominio..... 27
Figura 3 Diagrama de Casos de Uso 34
Figura 4 Vista de la arquitectura de eXcriba..... 41
Figura 5 Vista en subsistemas de la arquitectura del módulo Administración de usuarios y grupos ... 45
Figura 6 Ejemplo del patrón Experto 47
Figura 7 Ejemplo del patrón Controlador..... 47
Figura 8 Diagrama de clases del caso de uso Gestionar cuentas de usuario..... 48

Figura 9 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Crear cuentas de usuario” 53
Figura 10 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Eliminar cuentas de usuario” 54
Figura 11 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Modificar cuentas de usuario” 54
Figura 12 Diagrama de componentes correspondiente al caso de uso Gestionar cuentas de usuario 55
Figura 13 Diagrama de despliegue 57
Figura 14 Diagrama de clases del caso de uso Gestionar grupos de usuario 79
Figura 15 Diagrama de colaboración del caso de uso Gestionar Grupo de usuarios, sección Crear grupo 82
Figura 16 Diagrama de colaboración del caso de uso Gestionar Grupo de usuarios, sección Eliminar grupo 83
Figura 17 Diagrama de componentes del caso de uso Gestionar grupos de usuario 84

Introducción

El hombre siempre ha intentado crear artefactos que le faciliten el trabajo. Desde tiempos remotos ha ideado maquinarias que le ayuden en muchos ámbitos de su vida: máquinas para transportar, para cortar o realizar cualquier tarea que se le presente en su quehacer diario. Sin embargo, las máquinas de calcular no evidenciaron su evolución notoriamente hasta el nacimiento del primer ordenador. Posteriormente llegan los sistemas operativos y con ellos las aplicaciones para computadoras, las cuales a través de procesadores de texto, hojas de cálculo, editores de presentaciones, entre otras, permitían incrementar la productividad y efectividad del trabajo.

A pesar de los beneficios que ofrecían dichas aplicaciones, aún no eran capaces de satisfacer todas las necesidades de los usuarios, principalmente en las empresas. Se hacían necesarios sistemas que agilizaran los disímiles problemas que se presentaban en la gestión de negocios y control interno en las empresas. A raíz de esta situación surgen los sistemas de gestión empresarial, los cuales permitían realizar todos y cada uno de los pasos que exige la administración de una empresa, desde la producción, pasando por la logística, hasta la entrega del producto en el punto de venta (1). Estos sistemas se basan en la premisa que puntualiza que cada eslabón de la cadena de producción puede ser llevado a cabo de la manera más eficaz con un sistema empresarial que integre a los trabajadores con las máquinas.

Por otro lado la credibilidad de los datos constituye un factor de notable importancia en este sector y por ende es fundamental que la información sea estrictamente fiable, por lo que resulta indispensable establecer un control sobre los usuarios que acceden a la misma. Para resolver este tipo de situación la administración de usuarios y grupos es un punto de partida para controlar adecuadamente los accesos de usuarios a la información.

La mayor parte de las aplicaciones informáticas y redes actuales mantienen de uno u otro modo una relación de identidades personales (usuarios) asociadas normalmente con un perfil de seguridad, roles y permisos. De la misma forma funciona el Gestor de Documentos Administrativos (GDA) eXcriba, sistema que se está desarrollando en uno de los proyectos productivos correspondientes al Centro de Informatización Universitaria

(CENIA) en la Universidad de las Ciencias Informáticas. La meta fundamental del GDA eXcriba es gestionar de manera eficaz los documentos de una empresa, desde su creación o captura hasta su eliminación, disposición o conservación permanente en un archivo, para ello, dicho sistema hace uso del repositorio de contenidos y servicios que provee el ECM (Gestor de Contenido Empresarial) Alfresco en su versión Comunity 3.0, el cual constituye el núcleo del sistema.

Como cualquier otro sistema de gestión documental, eXcriba cuenta con un subsistema de seguridad y control de acceso cuyo eje central es el usuario; en función de estos queda definido todo un modelo y políticas de seguridad que garantiza que los usuarios solo puedan ejecutar un conjunto bien determinado de operaciones sobre cada contenido. La cuenta de usuario es el mecanismo que usa Alfresco para mantener de una u otra forma una relación de la identidad personal con el perfil de seguridad, roles y permisos de una persona, de modo que se pueda restringir y controlar qué puede hacer determinada persona sobre determinado recurso en el sistema.

Hasta la versión 2.0 de eXcriba toda la administración se lleva a cabo desde el cliente *web* del proyecto Alfresco Explorer[1], que se provee con la distribución de Alfresco. Esto significa que desde este cliente *web* se gestionan los usuarios y grupos de usuarios, las reglas de negocio y flujos de trabajo, los servicios de continuidad y mantenimiento del sistema, entre otros componentes de administración. En lo que a administración de usuarios respecta, a través de Alfresco Explorer se lleva este proceso se lleva a cabo desde la creación de una cuenta de usuario hasta la eliminación de la misma, la inserción de usuarios en grupos de usuarios y la configuración de perfiles de usuario.

A pesar de las funcionalidades y beneficios que provee Alfresco Explorer para la administración de cuentas de usuarios, en los diferentes despliegues del sistema se han identificado un conjunto de deficiencias que deben ser corregidas en pos de incrementar el nivel de satisfacción de los futuros usuarios. Por ejemplo, desde Alfresco Explorer resulta muy engorroso exportar e importar los usuarios del sistema, no es posible registrar manualmente los usuarios de un directorio activo, no se permite visualizar los

[1] Es el Explorador del cliente *web* Alfresco, el cual proporciona una completa interfaz *web* de usuario y capacidades de gestión de documentos, colaboración y administración.

grupos de manera jerárquica, ni es permisible obtener los grupos a los que pertenece un usuario determinado, entre otras dificultades.

Además de las deficiencias mencionadas anteriormente sobre la administración de cuentas y grupos de usuarios, existen otras que se ven más relacionadas con usuarios que no poseen cuenta administrativa en el sistema. Dichas deficiencias se refieren en este caso a cuestiones como el cambio de contraseña o clave personal, la modificación de los datos personales o asociados a la cuenta personal, entre otras configuraciones que pudieran influir en la comodidad del usuario al trabajar con el sistema. Todas estas acciones, hasta la versión actual del sistema, quedan bajo responsabilidad de los administradores; teniendo en cuenta que estas son tareas comunes y repetitivas, el tiempo y recursos que consumen estos procesos hacen que la productividad de los usuarios disminuya al mismo tiempo que se incrementa la labor de los administrativos.

Por las razones antes planteadas se plantea el siguiente **problema de investigación**: ¿cómo independizar del cliente web del proyecto Alfresco Explorer los procesos de administración de cuentas de usuario del Gestor de Documentos Administrativos eXcriba?

Para dar solución a la problemática planteada la presente investigación centra su **objeto de estudio** en la administración de identidades, delimitándose como **campo de acción** la administración de usuarios.

Se define como **objetivo general**: desarrollar un módulo para el Gestor de Documentos Administrativos eXcriba, que permita administrar las cuentas de usuario mediante el desarrollo y consumo de un conjunto de servicios del repositorio Alfresco.

Con el propósito de dar cumplimiento a lo previamente planteado se definen las siguientes **tareas de investigación**:

- sistematización de la información relacionada con los elementos teóricos que caracterizan la gestión de usuarios
- análisis de diferentes soluciones informáticas que incorporan entre sus servicios la administración de usuarios y grupos de usuarios
- selección de la metodología, las herramientas y tecnologías más adecuadas para el desarrollo de la solución

- identificación de los requerimientos del módulo a desarrollar
- diseño del módulo para la administración de usuarios y grupos de usuarios en el GDA eXcriba
- implementación del módulo para la administración de usuarios y grupos de usuarios para el GDA eXcriba
- selección de las técnicas de validación para las pruebas de caja negra
- validación del correcto funcionamiento de módulo mediante la utilización de pruebas de caja negra

Para el desarrollo de la presente investigación se hace uso de los siguientes **métodos científicos**:

Analítico - Sintético: para la elaboración del presente trabajo de diploma se dividirá el objeto de estudio en conceptos que serán examinados por separado, estudiándose rigurosamente cada uno de ellos de manera independiente y confrontando el criterio de disímiles autores y las correspondencias entre ellos. Además, se realizará un análisis de diferentes herramientas, metodología y técnicas necesarias para el desarrollo de la solución.

Histórico-lógico: con la utilización de este método se podrá realizar el estudio del estado del arte e investigar acerca de aplicaciones o soluciones similares, en este caso se contará como fuente principal de estudio el Gestor de contenido empresarial (ECM) Alfresco específicamente el proyecto Alfresco Explorer.

Justificación de la investigación

La presente investigación se realiza atendiendo a la necesidad que presentan los administradores y usuarios del Gestor de Documentos Administrativos eXcriba en la actualidad, de contar con un módulo que posibilite la administración de usuarios y grupos de usuarios desde la propia interfaz *web* de este sistema. Además se hace necesario que dicho módulo ofrezca funcionalidades que permitan disminuir las tareas repetitivas a las que se enfrentan los administradores del sistema. El desarrollo de esta investigación ha creado bases sobre el conocimiento necesario para lograr el desarrollo de una solución que facilite la administración de usuarios en la interfaz *web* de eXcriba, sin necesidad de hacer dicho proceso desde la interfaz *web* del proyecto Alfresco Explorer como se ha hecho hasta el momento.

La información obtenida en la investigación ha generado conocimientos nuevos, que se reflejan en la aplicación de estos en la solución que se propone, la cual contribuirá notablemente con el GDA eXcriba haciéndolo un sistema más completo.

El presente documento está desglosado en los siguientes capítulos:

Capítulo I: Fundamentos teóricos sobre la administración de usuarios: En ese capítulo se presentan un conjunto de conceptos y elementos teóricos necesarios para comprender mejor la administración de identidades, específicamente la administración de usuarios. Se plasma el resultado del estudio realizado a diferentes soluciones de Gestión Documental para analizar de ellas la forma en que gestionan los usuarios y grupos. Se describe además la metodología seleccionada para guiar el proceso de desarrollo de *software*, los lenguajes de programación, tecnologías y herramientas que se seleccionaron para el diseño e implementación de la solución.

Capítulo II: Propuesta de solución: en este capítulo se hace un análisis del estado actual de la administración de usuarios y grupos de usuarios del GDA eXcriba, en el presente capítulo, se expone a partir de un conjunto de requerimientos funcionales y no funcionales cómo se le dará solución a cada una de las problemáticas identificadas. Finalmente, se definen los casos de uso del sistema y se describen los mismos.

Capítulo III: Diseño del módulo: En este capítulo serán desarrollados los aspectos relacionados con el diseño de la solución, se representa y describe la arquitectura del sistema eXcriba y sus módulos, así como la descripción de los componentes que la conforman y los patrones de diseño utilizados. Se refleja además los diagramas de clases del diseño utilizando estereotipos *web* y los diagramas de colaboración correspondientes al caso de uso gestionar cuentas de usuario.

Capítulo IV: Implementación y prueba de la solución: Este capítulo describe cómo los elementos del modelo de diseño son implementados en términos de componentes y la manera en que se organizan de acuerdo con los nodos referidos en el modelo de despliegue. Se expone el nivel, tipo de prueba y técnica a los que se someterá el módulo.

Capítulo 1.Conceptos fundamentales sobre la administración de usuarios

En ese capítulo se presentan un conjunto de conceptos y elementos teóricos necesarios para comprender mejor la administración de identidades, específicamente la administración de usuarios. Se plasma el resultado del estudio realizado a diferentes soluciones de gestión documental para analizar de ellas la forma en que gestionan los usuarios y grupos. Se describe además la metodología seleccionada para guiar el proceso de desarrollo de *software*, los lenguajes de programación, tecnologías y herramientas que se seleccionaron para el diseño e implementación de la solución.

1.1 Preliminares

A continuación se dan a conocer un conjunto de definiciones imprescindibles para la comprensión de los temas que se abarcaron durante la investigación y que se presentan como referentes en el marco de este documento.

Usuario (en informática): usuario, es un individuo o entidad que utiliza una computadora, sistema operativo, servicio u otro sistema informático, además se utiliza para clasificar diferentes privilegios o permisos a los que tiene acceso un usuario o grupo de usuario, para interactuar con el ordenador o con los programas instalados en este (2).

Contraseña: se denomina contraseña al método de seguridad que se utiliza para identificar a un usuario frente a determinado sistema u aplicación. Además este término es usado fundamentalmente para permitir el acceso en un sistema donde le han sido otorgados permisos a un usuario o entidad, para realizar acciones específicas sobre determinados recursos del sistema (3).

Grupos de usuario: es la unión de dos usuarios o más, agrupados con un objetivo específico (4).

En términos de seguridad muchos conceptos se relacionan llegando a formar otros, tal es el caso de las cuentas de usuario que traen implícitamente los conceptos antes referenciados, según el Diccionario de Informática e Internet de Microsoft las **Cuentas de usuario:** consisten en información acerca del usuario, como la contraseña, permisos, datos personales por citar algunos. En un sistema de computadora seguro o multiusuario, la cuenta de usuario es un medio establecido para que un individuo pueda acceder al sistema y a sus recursos (5).

1.2 Administración de identidades

La administración de identidades es un conjunto de procesos, herramientas y estándares utilizados para la creación, mantenimiento y utilización de identidades digitales por parte de personas, sistemas y servicios con el fin de lograr el cumplimiento de objetivos tales como (6):

- los usuarios puedan acceder a los sistemas y funcionalidades requeridas para el desarrollo de sus funciones
- sea posible controlar adecuadamente los accesos a información sensible y los conflictos de segregación de funciones
- se posibilite verificar que los usuarios correspondan a la persona a la que están asociados
- se tenga un entendimiento de los procesos para agregar, cambiar y eliminar accesos
- exista disponibilidad permanente de la información de los accesos utilizados

1.2.1 Componentes de la administración de identidades

Los servicios de administración de identidades proveen una forma para gestionar eficazmente la identidad en la organización y se agrupan en cinco componentes principales (6):

Control de acceso

- políticas y estándares
- autenticación y autorización
- definición de roles empresariales y de aplicación (control de acceso basado en roles)
- acceso a información sensible / segregación de funciones

Administración de usuarios

- aprovisionamiento de usuarios
- análisis y aprobación de acceso
- administración delegada
- autoservicio
- administración de contraseñas
- administración de activos físicos

Servicios de información de identidades

- definición y normalización de directorios y repositorios
- sincronización de datos

Auditoría y cumplimiento

- revisión de accesos y eventos
- certificaciones

Servicios federados

- autorizaciones a socios y proveedores
- administración de usuarios

1.2.2 Ventajas de la administración de identidades

Los principales beneficios que se pueden lograr por medio de la implementación de una solución de gestión de identidades son (7):

- permite la asignación de roles y permisos condicionados por las responsabilidades o cargo que ocupe el usuario
- posibilidad de habilitar o deshabilitar cuentas de usuario, lo que implica que en caso que un trabajador o usuario le correspondan vacaciones, licencias o presente alguna incapacidad, la cuenta se desactivaría de manera momentánea hasta que el mismo se incorpore, en aras de evitar accesos no deseados a la aplicación o sistema
- integración de servicios y repositorios de datos de usuarios bajo una misma arquitectura, lo que facilita la administración de las cuentas de usuario y posibilita que un usuario solo necesite manejar una cuenta para acceder a diferentes aplicaciones y servicios empresariales
- consistencia en los datos relacionados con la identidad de los usuarios, lo que garantiza que si un dato es modificado en el repositorio central, dicho cambio se va a ver reflejado en las aplicaciones integradas en una solución de gestión de identidades. Por ejemplo, si a un empleado le cambian el contrato y el nuevo cargo que ocupará será económico en el departamento de industria, dicha información debe ser reflejada en la información de dicho sujeto en el directorio de empleados y en el acceso de este a las aplicaciones de la institución condicionado por el nuevo cargo que ocupa

1.2.3 Desventajas de la administración de identidades

Las principales desventajas que se pueden observar en la implementación de una solución de gestión de identidades (7):

- una solución de gestión de identidades permite que un usuario pueda acceder a diferentes aplicaciones usando la misma contraseña, lo que por un lado resulta cómodo para el usuario pudiese ser perjudicial en caso que no se utilice un esquema de autenticación robusto teniendo en cuenta: no utilizar contraseñas que contengan elementos personales (nombre, apellidos, fecha de nacimiento o el nombre de algún familiar entre otros), usar una combinación de letras, números y caracteres especiales, haciéndola compleja de descifrar por usuarios ajenos, entre otros aspectos importantes. Sin embargo a pesar de estas precauciones dichas claves pueden ser vulneradas fácilmente a través de un *keylogger* o *mouselogger* permitiendo eventos de suplantación de identidad, robo de información, entre otros
- en una solución de control de acceso y gestión de identidades, el acceso a las aplicaciones del negocio se realiza por medio de la autenticación de los usuarios contra el repositorio unificado de

identidades. Si se presentan fallos en los procesos de autenticación y autorización, esto afectaría todas las aplicaciones integradas bajo este esquema, mientras que en uno tradicional las fallas afectan solo a las aplicaciones puntuales

1.3 Administración de cuentas de usuario

La administración de cuentas de usuario forma una parte esencial de la administración de sistemas informáticos dentro de una organización. Sin embargo, para hacer este proceso de manera efectiva, un buen administrador de sistemas primero debe entender, qué son las cuentas de usuario y cómo funcionan. Las cuentas de usuario son el mecanismo mediante el cual el usuario se identifica con el sistema. Las mismas recogen un cúmulo de datos del usuario al ser creadas, los datos pueden ser personales, profesionales o combinaciones de ellos. Dentro de los datos personales generalmente se recoge el nombre, apellidos, fecha de nacimiento, por citar algunos, mientras que los profesionales suelen ser cargo laboral, institución en la que trabaja, ubicación dentro de la misma, entre otros, aunque siempre no se piden los mismos datos ya que estos pueden variar por interés de la organización o sistema en la que sean creadas. La actividad fundamental que realizan las cuentas de usuario es permitir el acceso de un usuario a determinado sistema a través de un nombre de usuario y contraseña.

Por otra parte es común encontrar las cuentas de usuario agrupadas en grupos de usuario con un fin específico, ya sea para garantizar organización (ejemplo: agrupar todos los usuarios de un departamento) o porque sea propicio para la asignación de permisos en un sistema informático. El uso de grupos es una manera inteligente de asignar privilegios puesto que dicha asignación se realiza de manera rápida y eficaz, sin necesidad de emplear tiempo y esfuerzo asignando permisos usuario por usuario cuando en ocasiones le son asignados los mismos privilegios a varios de estos.

1.3.1 El nombre de usuario

Desde el punto de vista del sistema, el nombre de usuario es la respuesta a la pregunta ¿Quién es usted?, los nombres de usuario tienen un requerimiento principal: deben ser únicos. Debido a este requerimiento, es de vital importancia establecer una convención de nombres[2] para lograrlo (8).

1.3.2 Tipos de cuentas de usuario

Las cuentas de usuario pueden ser clasificadas en dos tipos (9):

[2] Entiéndase por convención de nombres como una técnica o estrategia para establecer normas a la hora de definir los nombres de los usuarios

- cuenta de usuario: todas aquellas cuentas que sean utilizadas por los usuarios para acceder a los diferentes sistemas de información. Estas cuentas permiten el acceso para consulta, modificación, actualización o eliminación de información
- cuenta de administración: corresponde a la cuenta de usuario que permite al administrador realizar tareas específicas de usuario a nivel directivo, como por ejemplo: agregar/modificar/eliminar cuentas de usuario del sistema

1.3.3 Ciclo de vida de las cuentas de usuario

El ciclo de vida de las cuentas de usuario es un término usado para describir el flujo del proceso de gestión en un sistema basado en determinados eventos o factores de tiempo. Las etapas de dicho ciclo son: Inexistentes, Habilitadas, Deshabilitadas y Eliminadas (10).

La siguiente figura ilustra las diferentes etapas del ciclo de vida de las cuentas, todas las transiciones posibles, y las operaciones que establecen las transiciones.

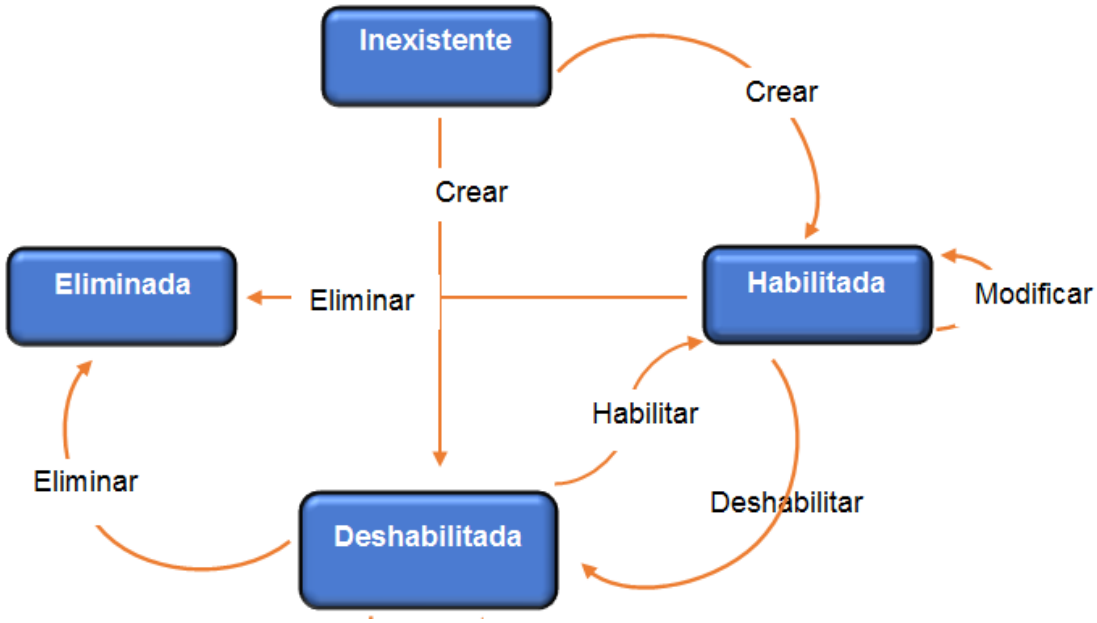


Figura 1 Descripción del ciclo de vida de las cuentas de usuario (10)

Inexistente

En esta etapa es donde las cuentas son creadas y se abren paso a una etapa posterior. Es aquí donde serán habilitadas o no, y se inicia el ciclo de vida de dicha cuenta.

Habilitada

Después de creada una cuenta con los datos pertinentes ya sea nombre de usuario y contraseña u otros datos, la misma puede ser habilitada gozando el usuario de todos sus privilegios que le correspondan o puede modificarse haciendo cambios en los datos de la cuenta. En este estado las cuentas pueden pasar al estado de deshabilitadas o simplemente ser borradas.

Deshabilitada

Al estar una cuenta deshabilitada el usuario no podrá acceder al sistema y por tanto tampoco a los privilegios que debe tener asignados. Además, en esta etapa dicha cuenta puede ser modificada lo que implica cambios en los datos de la misma ya sea el cambio de clave o el correo electrónico por citar algunos. La cuenta podrá entonces ser habilitada nuevamente o pasar a ser borrada.

Eliminada

Cuando una cuenta es eliminada, deja de existir de manera automática, no se encuentra en las bases de datos del sistema ningún dato referente a la misma por tanto el usuario al cual correspondía no tendrá acceso al sistema.

1.4 Administración de usuarios en los sistemas de gestión documental

En este epígrafe se plasman las características fundamentales de diferentes soluciones de gestión documental desarrolladas tanto a nivel nacional como internacional. De cada una de ellas se reflejan los principales aspectos relacionados con la administración de usuarios.

1.4.1 Nuxeo DM

Nuxeo es una empresa de gestión de contenidos que ofrece una plataforma (“la plataforma Nuxeo”) para construir aplicaciones ECM de código abierto (11). De las soluciones que ofrece Nuxeo como plataforma resulta conveniente conocer más a fondo acerca de una aplicación en especial, Nuxeo *Document Management* (DM) por ser un gestor documental de escala empresarial.

En cuanto a la administración de usuarios y grupos en Nuxeo DM (versión 5.5), cada usuario accede al sistema a través de su cuenta de usuario. Una vez autenticados, los usuarios solo podrán acceder a los recursos y ejecutar las acciones que le sean permitidas según los permisos asignados por el administrador. Este es el único encargado de la creación de nuevos usuarios y grupos, así como modificar los datos de las cuentas o borrarlas.

Principales funcionalidades

- crear usuario
- modificar usuario
- eliminar usuario
- crear grupos de usuario
- modificar grupos de usuario
- eliminar grupos de usuario

Aspectos positivos

Nuxeo DM presenta algunos aspectos positivos en cuanto a la administración de usuarios y grupos que son válidos destacar:

- el sistema permite la conexión a un directorio externo (ejemplo LDAP)
- en este sistema los usuarios se gestionan directamente desde la interfaz de Nuxeo DM

Aspectos negativos

De manera negativa Nuxeo DM presenta las siguientes deficiencias:

- la búsqueda de los grupos o subgrupos para proceder a editarlos solo es posible introduciendo como parámetro de búsqueda el nombre de usuario, cuando sería más factible realizar la búsqueda por nombre o apellidos adicionalmente al nombre de usuario
- la interfaz que permite editar los grupos de usuario es poco convencional, pues no permite realizar esta acción como normalmente lo hacen la mayoría de los sistemas que incorporan la administración de usuarios
- no es posible aplicar una misma operación sobre varios objetos, por ejemplo para modificar un grupo o subgrupo solo se puede adicionar o eliminar los usuarios de uno en uno, siendo la solución más factible poder eliminar o adicionar varios usuarios al mismo tiempo
- el sistema no ofrece la posibilidad de habilitar o no las cuentas de usuario

1.4.2 KnowledgeTree

KnowledgeTree es una solución de Gestión Electrónica de Documentos (GED) desarrollada por la sociedad sudafricana JamWarehouse. Dicha solución cuenta con un completo conjunto de funcionalidades y varios módulos que permiten la correcta integración en un entorno ofimático (12).

En este sistema la administración de usuarios y grupos se realiza de la siguiente forma: los usuarios pueden ser adicionados al sistema de manera manual o por una vía de autenticación externa por ejemplo usando LDAP. Una vez en el sistema dichos usuarios deben ser asignados a un grupo o rol específico

para que de esta forma le sean asignados los permisos que le corresponden, pues de manera individual no le serán asignados. Los grupos o roles son creados únicamente por el administrador y a partir de la existencia de un grupo se le puede añadir uno o varios subgrupos o en caso necesario también es posible eliminar estos últimos, además se permite visualizar todos los grupos existentes en el sistema editarlos o eliminarlos.

El uso de grupos y roles es una manera inteligente de llevar a cabo la asignación pues así se podrán agrupar usuarios con los mismos derechos en el sistema sin tener que ir asignando permisos a varios de ellos que pueden realizar las mismas acciones. Dichas acciones serán las que darán acceso o no a los usuarios en este sistema ya sea a algún documento o carpeta. Por otra parte el sistema permite habilitar o no las cuentas de los usuarios, por defecto cuando se crea una cuenta está habilitada, solo es deshabilitada para evitar accesos no deseados al sistema.

Principales funcionalidades

- crear cuenta de usuario
- habilitar o deshabilitar usuario
- modificar cuenta de usuario
- eliminar cuenta de usuario
- habilitar o deshabilitar cuentas de usuario
- crear grupos de usuario
- modificar grupos de usuario
- eliminar grupo de usuario
- adicionar usuarios a un grupo
- borrar usuario de un grupo
- adicionar o eliminar sub-grupo en un grupo
- asignar privilegios a todos los miembros de un grupo
- crear roles
- editar roles
- eliminar roles

Aspectos positivos

- brinda la posibilidad de visualizar todos los grupos existentes en el sistema
- utiliza una infraestructura de autenticación con un directorio activo o LDAP
- en este sistema los usuarios se gestionan directamente desde la interfaz de KnowledgeTree

Aspectos negativos

- la versión de código abierto con la que cuenta Knowledge Tree presenta módulos que requieren el pago de licencia
- una vez en el sistema los usuarios deben ser asignados a un grupo o rol específico para que de esta forma le sean asignados los permisos en el sistema, pues de manera individual no le serán asignados.

1.4.3 OpenKM

OpenKM es un gestor documental que permite el almacenamiento de documentos. Este sistema tiene dos versiones, una versión libre y una con soporte de pago (13).

Dichos usuarios pueden modificar algunos datos de su cuenta como la contraseña, recomendándose que sea modificada inmediatamente su contraseña por una más compleja que la que le fue asignada en la creación inicial de la cuenta. El establecimiento de permisos se realiza a través de roles, este sistema cuenta con dos roles, rol de administrador (*AdminRole*) y rol de usuario (*UserRole*) cada uno de ellos con permisos diferentes, el administrador tiene todos los permisos y puede realizar cualquier acción en el sistema, mientras que para el rol de usuario los permisos son restringidos y controlados por el administrador.

Principales funcionalidades

- crear usuario
- editar usuario
- eliminar usuario
- habilitar o deshabilitar cuentas de usuario
- crear rol
- editar rol
- eliminar rol

Aspectos positivos

- brinda la posibilidad de habilitar o deshabilitar cuentas de usuario, lo que permite evitar accesos no deseados a la aplicación o sistema
- en este sistema los usuarios se gestionan directamente desde la interfaz de OpenKM

Aspectos negativos

- algunas funcionalidades pueden generar errores en circunstancias específicas como el cambio de contraseña y la creación de un usuario

1.4.4 Alfresco Explorer

El explorador del cliente *web* de Alfresco ofrece una completa interfaz *web* de usuario, proporcionando capacidades de gestión de documentos, colaboración y administración. Anteriormente conocido como el cliente *web* de Alfresco, fue renombrado explorador después del lanzamiento de Alfresco Share en Alfresco 3.0 (14). En este sistema, solo el administrador puede crear usuarios así como modificar sus datos o eliminarlos. Los usuarios pueden pertenecer a más de un grupo o subgrupo (15).

Principales funcionalidades

- crear usuarios
- eliminar usuarios
- actualizar información de un usuario
- permite la creación de espacios para los usuarios
- crear grupos de usuarios
- eliminar grupos de usuarios
- modificar grupos de usuarios
- crear grupos y subgrupos
- añadir usuarios a grupos y subgrupos
- asignación de roles a usuarios

Aspectos positivos

A continuación se presenta algunas ventajas detectadas en el sistema Alfresco:

- permite visualizar todos los usuarios que han sido creados
- en este sistema los usuarios se gestionan directamente desde la interfaz de Alfresco Explorer

Aspectos negativos

Alfresco Explorer presenta algunas deficiencias tales como:

- desde Alfresco Explorer resulta engorroso exportar e importar los usuarios del sistema
- no permite registrar manualmente los usuarios de un directorio activo
- no visualiza los grupos de manera jerárquica
- no es posible obtener los grupos a los que pertenece un usuario determinado

1.4.5 AlfrescoUserGroupManager

El módulo AlfrescoUserGroupManager fue desarrollado a mediados de 2010. La idea de llevar a cabo esta implementación surgió a partir de la necesidad identificada durante la concepción del proyecto

"Informatización de las OACE[3]", de contar con un centro de control o panel de herramientas de administración que como su nombre lo indica se conformaría de un grupo de componentes cada uno de los cuales permitiría automatizar algún área de gestión o administración del sistema eXcriba.

En aquel entonces, a partir de las características del proyecto y como consecuencia de las deficiencias e inconformidades encontradas a los componentes "Administración de usuarios" y "Administración de grupos de usuarios" que forman parte de la consola de administración de Alfresco Explorer, se decidió que la primera herramienta que formaría parte del centro de control de eXcriba sería la de administración de cuentas y grupos de usuarios. A los efectos visuales, la herramienta funcionaba de la misma manera que un instalador de una aplicación de Windows; la idea era aprovechar la experiencia de los usuarios al tiempo que se mejoraban y proveían nuevas opciones de administración de usuarios.

Principales funcionalidades

- crear cuenta de usuario local
- importar usuario de LDAP
- importar usuarios desde un fichero LDIF [4]
- eliminar usuario
- crear grupo
- modificar grupo
- eliminar grupo

Aspectos positivos

Entre los aspectos positivos que caracterizan la herramienta en cuestión se puede mencionar:

- es posible crear nuevas cuentas de usuarios asociadas a usuarios registrados en un servidor LDAP
- es posible importar los usuarios registrados en un servidor LDAP desde un fichero LDIF
- es posible incorporar varios usuarios a un grupo sin necesidad de hacerlo uno a uno
- es posible visualizar los grupos de forma jerárquica, en forma de árbol
- en este sistema los usuarios gestionan directamente desde la interfaz de AlfrescoUserGroupManager

Aspectos negativos

Es posible mencionar también algunos elementos que caracterizan negativamente la herramienta:

[3] Oficina de administración del consejo de estado.

[4] El *LDAP Data Interchange Format* (LDIF) es un formato que se utiliza para la importación y exportación de datos independientemente del servidor LDAP que se esté utilizando.

- la aplicación solo es capaz de comunicarse con el repositorio de contenidos cuando ambos, el ordenador cliente desde el cual se ejecuta y el servidor en el cual se encuentra desplegado el repositorio, están sincronizados con una diferencia de 2 minutos
- la aplicación no permite saber dado un usuario los grupos a los que pertenece y de ahí modificar este estado
- la aplicación no permite realizar salvas de las cuentas de usuario ni de los grupos de usuarios previamente creados

Resultados del estudio

Después de realizar el análisis de algunos sistemas de gestión documental fue posible apreciar que ninguno de ellos permite solucionar las dificultades asociadas a la administración de usuarios y grupos que presenta el GDA eXcriba, por lo que se decide implementar un módulo que permita gestionar los usuarios y grupos del GDA eXcriba desde su propia interfaz *web*. A raíz del estudio realizado a los sistemas tratados anteriormente ha sido posible analizar los aspectos positivos y negativos que estos presentan en cuanto a la administración de usuarios y grupos y tenerlos en cuenta a la hora de desarrollar el módulo. Este análisis ha permitido además detectar funcionalidades que en caso de adicionarlas a la solución facilitarían el trabajo e interacción con el sistema, tanto a los administradores como a los usuarios que no poseen cuenta administrativa.

1.5 El proceso de desarrollo de *software*

1.5.1 Metodología de desarrollo

Las metodologías de desarrollo de *software* ofrecen una guía para encaminar de manera correcta el trabajo y así lograr un producto de calidad. Estas van orientando cómo separar el proyecto por etapas, las tareas que se irán haciendo en dichas etapas y las herramientas que se podrán emplear.

Para guiar el proceso de desarrollo del módulo Administración de usuarios y grupos se escogió la metodología RUP (Proceso Unificado Rational) por ser la utilizada para generar los artefactos que se fueron generando durante el desarrollo del GDA eXcriba y todos sus módulos, adicionalmente el equipo de proyecto tiene mayor experiencia en el trabajo con RUP, lo que ha permitido apreciar cuán satisfactoria es su aplicación en la gestión de proyectos.

Proceso Unificado Rational (RUP)

RUP es más que un simple conjunto de actividades necesarias para transformar los requerimientos del usuario en el sistema de *software*, es un marco de trabajo genérico que puede especializarse para una

gran variedad de sistemas de *software*. Este define claramente quién, cómo, cuándo y qué debe hacerse en un proyecto (16).

Esta metodología divide el proceso de desarrollo del *software* en cuatro fases: inicio, elaboración, construcción y transición, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto (16).

El proceso de *software* propuesto por RUP tiene tres características fundamentales: dirigido por casos de uso, centrado en la arquitectura y que es iterativo e incremental. RUP divide el proceso de software en 4 fases: inicio, elaboración, construcción y transición. Además son definidos flujos de trabajo que serán desarrollados a lo largo de todo el proceso de desarrollo del software, aunque el esfuerzo dedicado a cada flujo de trabajo varía dependiendo de la fase en la que se encuentre el proyecto, dichos flujos son: entorno, gestión de proyecto, gestión de configuración y cambio, despliegue, prueba, implementación, análisis y diseño, requisitos, y por último modelado de negocio (16).

Conociendo las particularidades de la solución a diseñar y las capacidades de cada una de las metodologías de desarrollo que existen, se decide usar RUP con Nivel 2 de CMMI característica, que hace de RUP una buena elección. Además, la selección de esta metodología ha estado condicionada por restricciones de diseño establecidas para el proyecto eXcriba y sus módulos.

1.5.2 Lenguaje de modelado

UML (*Unified Modeling Lenguage*) es un lenguaje gráfico que permite visualizar, construir y documentar los elementos que forman un sistema de *software* orientado a objetos, de ahí que se pueda realizar un análisis en UML indistintamente del lenguaje en el que finalmente sea implementado el sistema. UML entrega una forma de modelar elementos conceptuales como: procesos de negocio y funciones de sistema, además de actividades concretas como: escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de *software* reusables (17).

Para modelar los artefactos generados durante un proceso de desarrollo guiado por la metodología RUP se utilizará UML en su versión 2.0 pues es el lenguaje creado y propuesto por Jacobson, Booch y Rumbaugh para modelar los artefactos en la metodología RUP.

1.5.3 Herramienta para el modelado

Visual Paradigm para UML, es una herramienta CASE (*Computer Aided Software Engineering*) que sustenta el ciclo de vida completo del desarrollo de *software:* análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este está diseñado para una amplia gama de usuarios, incluidos los ingenieros de *software*, analistas de sistemas y analista de negocios (18).

Para el modelado de los artefactos generados durante el desarrollo del módulo se utilizará Visual Paradigm debido a que es la herramienta en la cual el equipo de trabajo tiene mayor experiencia, además es la herramienta que ha sido empleada para generar todos los artefactos durante el proceso de desarrollo de eXcriba desde su concepción.

1.6 Lenguajes de programación, herramientas y tecnologías

En la conformación de una aplicación informática intervienen factores decisivos como es el caso de la elección de lenguajes de programación, herramientas y tecnologías apropiadas. Esta selección se condiciona en gran medida por los requerimientos que debe tener la aplicación a desarrollar. En el presente epígrafe se expone la fundamentación de todos los elementos seleccionados para el proceso de construcción del módulo Administración de usuarios y grupos.

1.6.1 Tecnologías para el desarrollo de las interfaces de usuario

1.6.1.1 HTML

Es el Lenguaje de Marcado de Hipertexto (*Hyper-Text Markup Language*, HTML por sus siglas en inglés) con el cual se construyen las páginas *web* que interpretan los navegadores *web*, las cuales no son más que documentos de hipertexto. De modo básico, constituye un conjunto de etiquetas de marcado que sirven para definir el texto y otros elementos que compondrán una página. Una combinación de estas, agrupadas y anidadas, constituye la estructura del documento. Dichas etiquetas le dicen al navegador *web* como mostrar todos los contenidos de la página y describe la estructura y la información en forma de texto, además de que complementa esta información con otros elementos como son las imágenes.
Este fue el lenguaje utilizado para la construcción de las páginas *web* que formarían posteriormente las interfaces de usuario (26).

1.6.1.2 CSS

Las hojas de estilo en cascada se crearon con el mero objetivo de controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS, en el contexto de la *web* es la mejor forma de separar los contenidos de su presentación y es imprescindible para la creación de páginas *web* complejas. Esta separación contribuye a la creación de documentos HTML bien definidos y con significado completo. Además, mejora la accesibilidad, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes (27).
Tal como se hizo para el desarrollo del GDA eXcriba, incluyendo el caso de particular la solución propuesta como parte de esta investigación, mientras HTML se empleó para marcar los contenidos de las páginas, es decir, para designar lo que es un párrafo o lo que es una lista de elementos, CSS se utilizó para definir

el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos, la separación entre estos, la tabulación con la que se muestran los elementos de una lista y así con el resto de los elementos de las páginas *web*.

1.6.1.3 JavaScript (Client-Side)

JavaScript es un lenguaje de programación que se utiliza principalmente para incorporar a las páginas *web* efectos, animaciones y acciones sin tener que acceder a un servidor *web*. Técnicamente JavaScript es un lenguaje interpretado, o sea, no es necesario compilar los programas escritos en este lenguaje para poder ejecutarlos. En otras palabras, estos programas se pueden ejecutar directamente en cualquier navegador sin necesidad de procesos de compilación intermedios (24).

Por las características antes mencionadas, básicamente, durante el desarrollo de la solución se hizo uso de JavaScript para brindar dinamismo a las diferentes interfaces de usuario. También se emplearon diferentes funciones de dicho lenguaje para validar los formularios antes de enviar los datos al servidor y contribuir así a la calidad de los datos a almacenar en el repositorio de contenidos de Alfresco, en este caso, los datos referentes a las cuentas de usuario.

1.6.1.4 JQuery

jQuery es un *framework* JavaScript de código abierto que sirve como base para la programación de aplicaciones para la *web*. Brinda un conjunto de funciones y objetos que permiten abstraerse de uno de los principales problemas a los que se enfrenta el desarrollo de una aplicación *web*: la incompatibilidad entre navegadores. De manera general, también soluciona o da soporte a problemas frecuentes en el desarrollo de aplicaciones *web* actuales, entre ellos (25):

- acceso a una parte específica de una página
- modificación de la apariencia de una página *web*, su contenido o parte de este
- respuesta a la interacción de los usuarios con las páginas
- animaciones
- obtención y envío de información al servidor sin refrescar o recargar la página

Teniendo en cuenta que todos los componentes (visuales, de interacción y de comunicación con el servidor) de la capa de presentación del GDA eXcriba están escritos o hacen uso de este *framework*, y por los aspectos anteriormente citados se decidió emplear JQuery en conjunto con diferentes funciones nativas de JavaScript para el desarrollo de la capa de presentación del módulo.

1.6.2 Tecnologías para el desarrollo de la lógica de negocio

1.6.2.1 PHP

PHP (Hypertext Preprocessor), Es un lenguaje de código abierto, interpretado y de alto nivel, embebido en páginas HTML y ejecutado en el servidor. La mayoría de su sintaxis es similar a C, Java y Perl. PHP es un lenguaje de programación simple y eficiente, hecho pensando en la *web* (28).

Se decide utilizar PHP en su versión 5.3 ya que es el lenguaje propuesto por la arquitectura del sistema sobre el cual se ha desarrollado el cliente *web* de eXcriba, por lo que al desarrollarse las diferentes funcionalidades que le darán cumplimiento a la presente investigación se debe seguir esta restricción de diseño, para que la integración de estas funcionalidades al sistema se realice de manera satisfactoria y el resultado obtenido sea el esperado.

1.6.2.2 CodeIgniter

CodeIgniter, originalmente desarrollado por Rick Ellis, es un *framework* para desarrollar aplicaciones o sitios *web* usando PHP. Su objetivo es facilitar el desarrollo de proyectos mucho más rápido que lo que se podría hacer comenzando desde cero, proveyendo un conjunto de bibliotecas para tareas comunes y una estructura lógica para acceder a esas bibliotecas (29). Se decide usar este marco de trabajo ya que es una restricción de diseño heredada, es decir, fue propuesto en la arquitectura del sistema por estar bien equipado para el desarrollo de aplicaciones escritas en el lenguaje PHP.

Se decide utilizar CodeIgniter en su versión 1.7.2 del lado del servidor como marco de trabajo para implementar la lógica de negocio de la solución a desarrollar, ya que esta es una restricción de la arquitectura como el marco de trabajo para la capa de aplicación.

1.6.3 Tecnologías para la comunicación con el repositorio de contenidos

1.6.3.1 REST (Representational State Transfer)

REST se define como una técnica o estilo arquitectónico de *software* orientado a sistemas de hipermedia distribuido, como la *World Wide Web* (www) y a su vez es una abstracción de los elementos arquitectónicos dentro de tales sistemas (19).

La implementación de servicios siguiendo los principios de REST utilizando HTTP, posibilita que al realizar una solicitud al servicio este no retorne la base de datos completa, sino un tipo de datos interpretable por el cliente (JSON[5] , XML, HTML). Los sistemas basados en los principios de REST se conocen como

[5]JSON: acrónimo de JavaScript *Object Notation* (en español Notación de Objetos de JavaScript), es un formato de texto para el intercambio de datos

RESTful y a los servicios que se implementan según lo anteriormente expuesto, son conocidos como servicio *web* RESTful (20).

Dado que el Gestor de Documentos Administrativos eXcriba tiene todos sus servicios implementados en el estilo arquitectónico REST, se hace necesario desarrollar los servicios correspondientes al módulo Administración de usuarios y grupos siguiendo este mismo estilo arquitectónico. El aprovechamiento y uso de la API RESTful que brinda el ECM Alfresco para el cumplimiento del objetivo fundamental del presente trabajo de diploma, se fundamenta en la facilidad de uso de dicha API, la amplia documentación referente al tema, así como una vasta comunidad para el intercambio y la colaboración.

1.6.4 Tecnologías para el desarrollo de los servicios *web*

1.6.4.1 WebScript

Webscript[6] es el *framework* que provee Alfresco para el desarrollo de servicios *web* basados en el estilo arquitectónico REST. Proporciona una forma de interactuar con el repositorio de contenido e integrar Alfresco con otros sistemas haciendo uso del lenguaje JavaScript (21). Este *framework* ha sido utilizado para el desarrollo de aplicaciones que utilizan Alfresco como repositorio, entre ellas Alfresco Share, *Web* Studio incluso el propio GDA eXcriba. Se decidió utilizar este *framework* porque ha sido utilizado para desarrollar todos los servicios que forman parte de la versión actual del GDA eXcriba, además se tuvo en cuenta la facilidad de implementación de los servicios haciendo uso del lenguaje JavaScript.

1.6.4.2 FreeMarker

FreeMarker es un motor de plantillas[7] así como una herramienta genérica para generar la salida de texto basado en plantillas. No es una aplicación para los usuarios finales en sí mismo, sino un paquete de Java que los programadores pueden utilizar en sus productos (22). Este motor es usado por el *framework* WebScript de Alfresco para proveer un mecanismo de generación de respuestas preformateadas, una vez finalizada la ejecución de un *web script*[8], o sea, una vez finalizada la ejecución de la lógica de negocio se emite una respuesta cuya salida queda determinada por la estructura de la plantilla de presentación correspondiente al servicio en ejecución.

Una de las principales ventajas que usa el *framework* WebScript de este motor es que permite generar diferentes vistas y representaciones como resultado de la ejecución de un mismo servicio. De esta manera

[6] Nombre del framework webscript que provee Alfresco.

[7] Una plantilla de presentación es un documento de texto que, aplicado a un modelo de datos produce una salida (un nuevo documento) que tiene la estructura definida en la plantilla.

[8] Es una URI unido a un servicio utilizando los métodos estándar de HTTP, como GET, POST, PUT O DELETE.

se logra la separación definitiva entre la lógica de negocio que se ejecuta durante una petición y el formato de salida de la respuesta generada.

1.6.4.3 JSON

JSON (Java Script Object Notation – Notación de objetos de JavaScript) es un formato de texto utilizado para el intercambio de datos. Está basado en un subconjunto del lenguaje de programación Java Script. JSON es completamente independiente del lenguaje que se emplee pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (23). Este es el formato utilizado para el intercambio de datos entre cliente y servidor, dicho intercambio puede ser entre el navegador *web* y el Apache cuando el cliente envía una petición o entre el cliente *web* de eXcriba y Alfresco en el momento que se hacen las peticiones a los servicios.

1.6.4.4 JavaScript (Server-Side)

JavaScript del lado del servidor (SSJS, por sus siglas en inglés) se refiere a JavaScript que se ejecuta en el lado del servidor y por tanto, no es descargado al cliente, generalmente al navegador. El término se utiliza para diferenciarlo del JavaScript regular, o sea, aquel que se ejecuta del lado del cliente y que es descargado por tanto al navegador (30).

Además de aliviar la complejidad del desarrollo del lado del servidor JavaScript ofrece algunas prestaciones que para algunos pueden no ser consideradas como tal. Entre estas se puede citar:

- con el mismo código se puede validar los datos en el cliente y en el servidor (por seguridad), por lo que las validaciones no perderán la sincronización
- el mismo código puede preparar (tanto del lado del servidor como del cliente) el modelo de objetos de la página (HTML DOM) y modificarlo en el cliente, cuando el usuario cambia los datos o los mismos se actualizan desde el propio servidor
- este lenguaje en el servidor y el cliente aumenta la cantidad de código que puede ser compartido o reutilizado y disminuye el costo y esfuerzo necesario para contratar a los desarrolladores ya que no es necesario contratar a personas diferentes para escribir código en diferentes lugares

Este lenguaje fue utilizado en el módulo para el desarrollo de la lógica de negocio de los servicios *web* haciendo uso de la API de JavaScript que provee Alfresco.

Capítulo 2. Propuesta de solución

Partiendo de un análisis del estado actual de la administración de usuarios y grupos de usuarios del GDA eXcriba, en el presente capítulo, se expone a partir de un conjunto de requerimientos funcionales y no funcionales cómo se le dará solución a cada una de las problemáticas identificadas. Finalmente, se definen los casos de uso del sistema y se describen los mismos.

2.1 Análisis crítico de la situación problemática

Actualmente la administración de usuarios y grupos de usuarios del GDA eXcriba se lleva a cabo desde el cliente *web* del proyecto Alfresco Explorer o desde la herramienta AlfrescoUserGroupManager; sin embargo, por los aspectos negativos de la misma, antes citados (véase Epígrafe 1.4.5) la herramienta más usada es Alfresco Explorer. No obstante, aún existen un conjunto de inconvenientes e inconformidades bien identificados y definidos respecto a esta última.

La principal problemática que afecta al proyecto eXcriba es la dependencia hacia el proyecto Alfresco Explorer en cuanto a la administración de usuarios y grupos. Desde la versión 3.0 de Alfresco el proyecto Alfresco Explorer ha venido presentando las mismas funcionalidades y deficiencias por lo que la administración de usuarios en eXcriba acarrea los mismos problemas desde entonces sin la posibilidad de una mejora en este aspecto ni siquiera migrando a una versión superior de Alfresco.

Modificación del estado de una cuenta de usuario

Tal como se había mencionado en el (epígrafe 1.3.3), existen varios estados por los que puede transitar un usuario en su ciclo de vida: inexistente, habilitada, deshabilitada, eliminada. En este caso un usuario pudiera pasar del estado habilitada a deshabilitada por un sinfín de razones, por ejemplo, si el usuario va a estar fuera de la organización por un año quizás sería razonable desactivar el acceso al sistema a través de sus credenciales, evitando así que otros puedan apropiarse de los privilegios que le fueron conferidos al otro usuario. Actualmente no es posible realizar este tipo de configuración en el sistema.

Cambio de contraseña

El cambio de contraseñas es una de las tareas que con más frecuencia se ejecuta en el contexto de la administración de identidades. Si bien los usuarios pierden u olvidan sus claves con periodicidad es común que se les exija a los mismos que cambien dicha clave por cuestiones de seguridad, previniendo robos y uso indebido de contraseñas. Actualmente el cambio de contraseñas de los usuarios del GDA eXcriba solo se puede llevar a cabo desde Alfresco Explorer, al cual solamente debe tener acceso los administradores, de esta manera toda la labor caería en manos de estos.

Operaciones de salva y mantenimiento del sistema

En todo sistema informático que manipula datos importantes, es necesario contar con alguna herramienta que garantice la salva de tal información sobre todo para casos de contingencia. En el caso de sistemas empresariales de gran escala es necesario tener en cuenta que estos datos a salvar no solo se refieren a la información en forma de documentos sino también otros como por ejemplo las cuentas y datos de los usuarios que interactúan con el sistema. Lo realmente conveniente es que el sistema pudiera ser restaurado en las mismas condiciones en que se encontraba justo antes de colapsar.

En Alfresco Explorer es posible actualmente salvar los datos de espacios específicos, sin embargo, no es posible hacer lo mismo con los datos asociados a las cuentas de usuario. Esto trae como consecuencia que cuando el sistema deja de funcionar por determinada razón, a la hora de realizar el proceso de restauración, se puede reconstruir el sistema de ficheros configurado hasta el punto de ruptura pero los administradores del sistema tendrían que asumir la responsabilidad de volver crear las cuentas de usuarios, lo que no garantiza que los datos sean los mismos, sobre todo las contraseñas y datos personales. De la misma manera habría que conformar nuevamente todos los grupos de usuarios.

Por otra parte, existe un mecanismo de importación de usuarios desde un fichero en formato XML con una estructura de datos bien definida. Esta herramienta que se provee con el propio núcleo de Alfresco no es muy utilizada por tres razones fundamentales, en primer lugar, no existe una herramienta para exportar las cuentas de usuarios a la estructura de datos correcta, en segundo lugar y como consecuencia de lo anterior los

administradores tendrían que escribir el fichero con la estructura de datos en el formato XML manualmente, y en tercer lugar, la operación se lleva a cabo desde la línea de comandos que no constituye una tarea sencilla para cualquier persona.

Otras consideraciones importantes

Un elemento fundamental, identificado además como deficiencia en la administración de grupos de usuarios desde Alfresco Explorer, es la forma en que se visualizan los grupos. Actualmente la estructura definida por los grupos de usuarios es navegable en forma de carpetas. Esta forma de visualización es bastante incómoda, sobre todo porque la mayoría de las personas están acostumbradas a interpretar y moverse por una estructura jerárquica en forma de árbol. Se ha comprobado que la navegación de estructuras jerárquicas con muchos niveles en forma de carpeta disminuye la rapidez y efectividad a la hora de buscar lo que se desea.

Un elemento fundamental que además resuelve alguno de los problemas anteriormente planteados es contar con alguna funcionalidad que permita obtener los grupos a los que pertenece un usuario. Los administradores actualmente no cuentan con una funcionalidad que le permita obtener esta información, por tanto, cuando se desea por ejemplo, eliminar un usuario de todos o de un conjunto de los grupos a los que pertenece entonces el administrador debe navegar por toda la estructura jerárquica que como se había mencionado no es la más adecuada. Por esta razón se considera que esta es una de las principales deficiencias encontradas por los administradores que hasta el momento son los que han asumido esta tarea.

También existen un conjunto de funcionalidades que pueden ser de gran ayuda y que actualmente no existe una vía razonablemente efectiva, al menos desde Alfresco Explorer, que permita llevarlas a cabo. No es posible realizar filtros por tipos de cuentas de usuarios, o sea, obtener un listado de los usuarios que son administradores del sistema, o los que no lo son, entre otras.

De manera general, se han expresado en este mismo epígrafe un conjunto de elementos y consideraciones que demuestran los inconvenientes que trae consigo realizar la administración de usuarios y grupos de eXcriba dependiendo de Alfresco

Explorer. La solución para esta problemática debe ser incorporar un módulo que desde el propio cliente *web* del gestor que permita llevar a cabo todas las operaciones identificadas como deficiencias de la solución actual más las que en algún momento fueron identificadas y clasificadas como efectivas e importantes.

2.2 Modelo de Dominio

Un modelo del dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las “cosas” que existen o los eventos que suceden en el entorno en el que trabaja el sistema (16).

A continuación se muestra el modelo del dominio correspondiente al módulo Administración de usuarios y grupos:

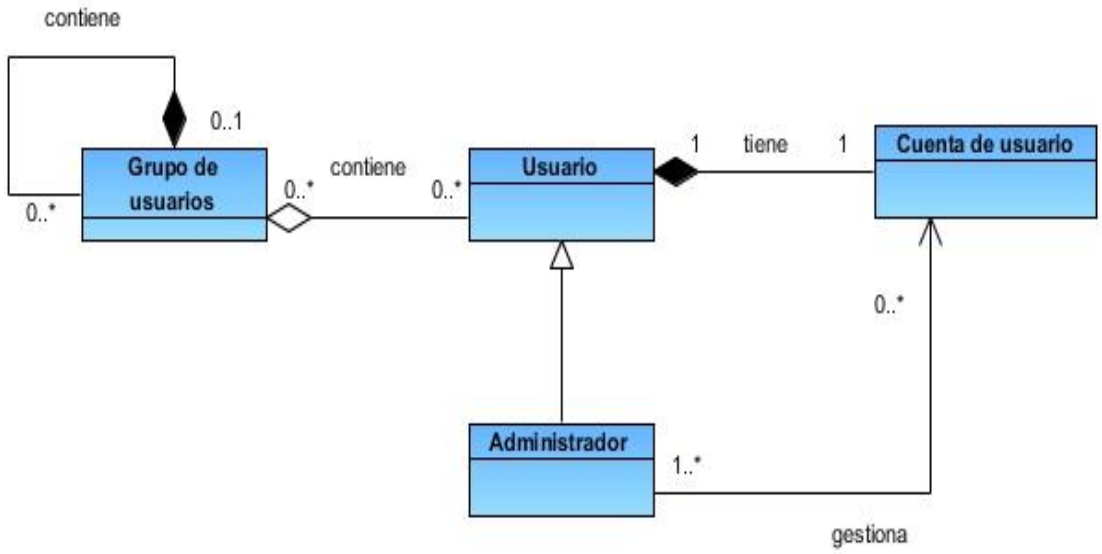


Figura 2 Diagrama de Modelo del dominio

Descripción de las entidades

Usuario: Es la persona o aplicación que interactúa con el negocio.

Administrador: Es un usuario que posee privilegios administrativos en el negocio.

Grupo de usuario: Hace referencia a un conjunto de usuarios que se agrupan con un objetivo específico.

Cuenta de usuarios: Es la entidad que contiene los datos de los usuarios.

Descripción de las relaciones entre las entidades

Grupo de usuario-contiene-Usuario: Un grupo de usuarios, puede estar vacío o formado por uno o más usuarios. De igual forma, un usuario puede pertenecer a tantos grupos como sea necesario o no pertenecer a ningún grupo.
Usuario-tiene-Cuenta de usuario: Un usuario tiene una y solo una cuenta de usuario y una cuenta de usuario puede pertenecer a uno y solo un usuario.
Administrador-es un-Usuario: El administrador es a su vez un usuario por tanto también tiene una cuenta de usuario y puede o no pertenecer a diferentes grupos de usuarios.
Administrador-gestiona-Cuenta de usuario: Una cuenta de usuario puede ser gestionada por uno o varios administradores y un administrador puede gestionar ninguna o muchas cuentas.
Grupo de usuarios-contiene-Grupo de usuarios: Un grupo de usuarios puede estar conformado por cero o más grupos de usuarios y a su vez un grupo de usuarios puede pertenecer como máximo a un grupo de usuarios.

2.3 Especificación de los requisitos del software

Según el estándar 1233 de la IEEE: Guía para el desarrollo de Especificaciones de Requerimientos de Sistemas, un requisito se define como (32):

- condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
- condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente

Se puede concluir que los requisitos de *software* son características y funcionalidades que debe poseer un sistema, están enfocados hacia todo lo que debe hacer el *software*. Los requisitos pueden ser clasificados en requisitos funcionales y no funcionales.

2.3.1 Técnicas para la captura de los requisitos

En el proceso de desarrollo de un sistema, sea o no para la *web*, el equipo de desarrollo se enfrenta al problema de la identificación de requisitos. La definición de las necesidades del sistema es un proceso complejo, pues en él hay que identificar los requisitos que el sistema debe cumplir para satisfacer las necesidades de los usuarios finales y de los clientes. Para realizar este proceso, no existe una única técnica

estandarizada y estructurada que ofrezca un marco de desarrollo que garantice la calidad del resultado.

Existe en cambio un conjunto de técnicas, cuyo uso proponen las diferentes metodologías para el desarrollo de aplicaciones *web*. Se debe tener en cuenta que la selección de las técnicas y el éxito de los resultados que se obtengan, depende en gran medida tanto del equipo de análisis y desarrollo, como de los propios clientes o usuarios que en ella participen (33).

Análisis de sistemas existentes

Mediante el análisis de sistemas existentes es posible estudiar aplicaciones similares a la que se necesita obtener. Una vez que se tiene la concepción del funcionamiento de un *software* similar en cuanto a funcionalidades y características es más sencillo identificar los requisitos del sistema que se necesita implementar.

Durante la investigación se realizó un estudio de aplicaciones similares a la que se necesitaba obtener, en las cuales se observaron los diseños de sus interfaces, las funcionalidades que ofrecen, el grado de dificultad a la hora de interactuar con la aplicación, los colores que emplean, entre otros rasgos importantes que contribuyeran a obtener un producto con la mejor calidad posible.

Tormenta de ideas

Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre. Consiste en la mera acumulación de ideas y/o información sin evaluar las mismas. El grupo de personas que participa en estas reuniones no debe ser muy numeroso (máximo 10 personas), una de ellas debe asumir el rol de moderador de la sesión, pero sin carácter de controlador (33).

Se efectuaron reuniones con el equipo de desarrollo del proyecto eXcriba y a partir de un conjunto de ideas propuestas principalmente por los analistas, se definieron los primeros requisitos con los que debería cumplir el módulo. Se estipuló conveniente una nueva reunión incluyendo algunos integrantes del proyecto para acumular más opiniones y definir una versión final de los requisitos del módulo.

Sketches y Storyboards

Está técnica es frecuentemente usada por los diseñadores gráficos de aplicaciones en el entorno *web*. La misma consiste en representar sobre papel en forma muy esquemática las diferentes interfaces al usuario (*sketches)*. Estos *sketches* pueden ser agrupados y unidos por enlaces dando idea de la estructura de navegación (*Storyboard*) (33). A partir del estudio de sistemas existentes se tomaron ideas para elaborar los primeros prototipos y mostrarlos al equipo de proyecto. Una vez que esto ocurre, el próximo paso es escuchar sus puntos de vista y perfeccionar los prototipos en la medida de lo posible antes de mostrarlos al cliente.

2.3.2 Requerimientos funcionales

Los requerimientos funcionales son declaraciones de los servicios que debe proporcionar el sistema además especifican la manera en que éste debe reaccionar a determinadas entradas y permiten especificar cómo debe comportarse el sistema en situaciones particulares (34). A continuación se enumeran los requisitos funcionales identificados para el módulo:

RF.01- Crear cuenta de usuario: el sistema debe permitir al administrador crear cuentas de usuarios en el sistema.

RF.02- Modificar cuenta de usuario: el sistema debe brindar una funcionalidad que le permita al administrador modificar los datos asociados a las cuentas de usuario, exceptuando el identificador de la cuenta.

RF.03- Eliminar cuenta de usuario: el sistema debe permitirle al administrador eliminar cuentas de usuario.

RF.04- Listar usuarios: el GDA debe permitirle al administrador obtener un listado de todos los usuarios registrados, permitiendo visualizar de cada uno de ellos el nombre, apellidos, identificador de usuario, tipo y estado de la cuenta.

RF.05- Buscar usuario: el GDA debe permitir que el administrador pueda realizar búsquedas de usuarios de acuerdo a determinados criterios asociados fundamentalmente a sus cuentas de usuarios, los criterios de búsqueda deben ser: nombre, apellidos o nombre de usuario.

RF.06- Actualizar contraseña de usuario: el GDA debe brindar una funcionalidad que permita al administrador modificar la clave personal de los usuarios.

RF.07- Crear grupo de usuarios: el GDA debe permitir al administrador crear un grupo de usuarios.

RF.08- Eliminar grupo de usuarios: el GDA debe permitir al administrador eliminar grupos de usuarios. Bajo ningún concepto eliminar un grupo de usuarios puede traducirse en la eliminación de los usuarios que a este pertenecen.

RF.09- Adicionar usuario a un grupo: el GDA debe permitir que el administrador pueda incorporar usuarios previamente registrados a diferentes grupos. Un grupo de usuarios podrá estar conformado por tantos usuarios como se estime y un usuario podrá pertenecer a tantos grupos como sea necesario.

RF.10- Eliminar usuario de un grupo: el sistema debe garantizar que el administrador tengan la capacidad de eliminar usuarios de un grupo determinado. Bajo ninguna circunstancia el sistema debe permitir que la eliminación de un usuario de un grupo determinado implique la eliminación del usuario del sistema.

RF.11- Listar elementos de un grupo: el GDA debe permitir al administrador obtener un listado de los usuarios y subgrupos que pertenecen a un grupo determinado, permitiendo ejecutar filtros o sea, obtener solo los usuarios que se necesiten.

RF.12- Realizar salvas a las cuentas de usuario: el GDA debe permitirle al administrador realizar salvas de las cuentas de usuario existentes.

RF.13- Restaurar cuentas de usuarios: el sistema debe permitir al administrador restaurar cuentas de usuario a partir de una salva previamente realizada.

RF14- Cambiar estado de una cuenta de usuario: el GDA debe brindar una funcionalidad que permita al administrador modificar el estado de una cuenta de usuario, ya sea para ponerla en estado activo como inactivo.

RF.15- Cambiar el tipo de cuenta de usuario: el sistema debe permitirle al administrador modificar el tipo de cuenta de un usuario. Los tipos de cuenta son: administrativa o estándar.

RF.16- Listar grupos de usuarios: el GDA debe permitirle al administrador listar todos los grupos registrados.

RF.17- Obtener de un usuario los grupos a los que pertenece: el GDA debe permitirle al administrador obtener un listado de los grupos de usuarios a los que pertenece un usuario determinado.

RF.18- Realizar salvas a grupos de usuarios: el GDA debe permitirle al administrador realizar salvas de los grupos de usuario existentes.

RF.19- Restaurar grupos de usuarios: el sistema debe permitir al administrador restaurar los grupos de usuarios a partir de una salva previamente realizada.

RF.20- Modificar cuenta personal: el sistema debe permitirle a los usuarios modificar los datos de su cuenta, exceptuando el identificador de usuario, cuota, tipo y estado de cuenta.

RF.21- Modificar clave personal: el sistema debe permitir a los usuarios cambiar su clave personal.

2.3.3 Requerimientos no funcionales

Los requisitos no funcionales imponen restricciones en el diseño o la implementación. Son propiedades o cualidades que el producto debe tener (35). A continuación se muestran los requisitos no funcionales identificados para el módulo:

Requisitos de Usabilidad:

RNF.1- Idioma: Utilizar el idioma español para los mensajes y textos de la interfaz**.**

Restricciones de diseño

RNF.2- La interfaz *web* del módulo deberá implementarse haciendo uso del *framework* JQuery en su versión 1.3.2.

RNF.3- Para la implementación de la lógica de negocio se hará uso del lenguaje de programación PHP en su versión 5.3 y del *framework* CodeIgniter en su versión 1.7.2.

2.3.4 Técnicas de Validación de Requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de estos define realmente el sistema que el usuario necesita o el cliente desea (33).

Matrices de trazabilidad: Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo qué objetivos cubre cada requisito, de esta forma se podrán detectar inconsistencias u objetivos no cubiertos (33). La matriz de trazabilidad permitió comprobar si los casos de uso del sistema atendían todos los requisitos funcionales identificados y así evidenciar la correspondencia entre estos garantizando que todos los requisitos estén reflejados en

los casos de uso y por tanto aplicados en la solución logrando cumplir con las expectativas del cliente (Ver Epígrafe 2.4.3).

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (33). Los prototipos fueron claves en el proceso de diseño del módulo y favorables a la hora de mostrar al cliente los prototipos como una propuesta de lo que pudiese ser la aplicación. El resultado de la utilización de este método permitió la creación de prototipos para el módulo ajustados a las necesidades del cliente.

2.4 Captura de requisitos como casos de uso

Un caso de uso describe la secuencia de interacciones que se producen entre el sistema y los actores del mismo para realizar una determinada función. Los actores son elementos externos (personas u otros sistemas, entre otros) que interactúan con el sistema.

La ventaja esencial de los casos de uso es que resultan muy fáciles de entender para el usuario o cliente, sin embargo carecen de la precisión necesaria cuando no se acompañan con una información textual o detallada (33).

2.4.1 Definición de los actores

A continuación serán descritos los actores del sistema:

Actores	Justificación
Administrador	Es el encargado de realizar todas las acciones relacionadas con la administración de usuarios en el sistema.
Usuario	Es la persona que interactúa con el sistema para realizar determinadas acciones tales como: modificar datos de su cuenta y cambiar su clave personal.

Tabla 1 Definición de los actores del sistema

2.4.2 Diagrama de caso de uso

Figura 3 Diagrama de Casos de Uso

2.4.2 Validación de requerimientos: Matriz de trazabilidad

En este epígrafe se muestra mediante una matriz de trazabilidad como son recogidos todos los requisitos funcionales en los casos de uso identificados.

A continuación se enumeran los casos de uso del sistema para facilitar el entendimiento de la tabla que se muestra posteriormente:

CU.1- Listar usuarios

CU.2- Listar grupos de usuarios

CU.3- Gestionar cuentas de usuario

CU.4- Actualizar contraseña

CU.5- Gestionar usuarios de un grupo

CU.6- Cambiar estado de cuenta

CU.7- Restaurar grupo de usuarios

CU.8- Realizar salvas a grupo de usuarios

CU.9- Restaurar cuentas de usuario

CU.10- Obtener grupos de un usuario

CU.11- Cambiar tipo de cuenta

CU.12- Buscar usuarios

CU.13- Gestionar grupos de usuario

CU.14- Realizar salvas de cuentas de usuario

CU.15- Listar elementos de un grupo

CU.16- Modificar clave personal

CU.17- Modificar cuenta personal

	Cu 1	Cu 2	Cu 3	Cu 4	Cu 5	Cu 6	Cu 7	Cu 8	Cu 9	Cu 10	Cu 11	Cu 12	Cu 13	Cu 14	Cu 15	Cu 16	Cu 17
Rf1			x														
Rf2			x														
Rf3			x														
Rf4	x																
Rf5												x					
Rf6				x													
Rf7													x				
Rf8													x				
Rf9					x												
Rf10					x												
Rf11															x		
Rf12														x			
Rf13									x								
Rf14						x											
Rf15											x						
Rf16		x															
Rf17										x							
Rf18								x									
Rf19							x										
Rf20																	x
Rf21																x	

Tabla 2 Matriz de trazabilidad

Como se puede observar en la tabla anterior cada requisito identificado ha sido cubierto por al menos un caso de uso del sistema, lo que significa que la especificación de caso de uso propuesta satisface todas las necesidades del cliente.

2.4.3 Descripción de los casos de uso

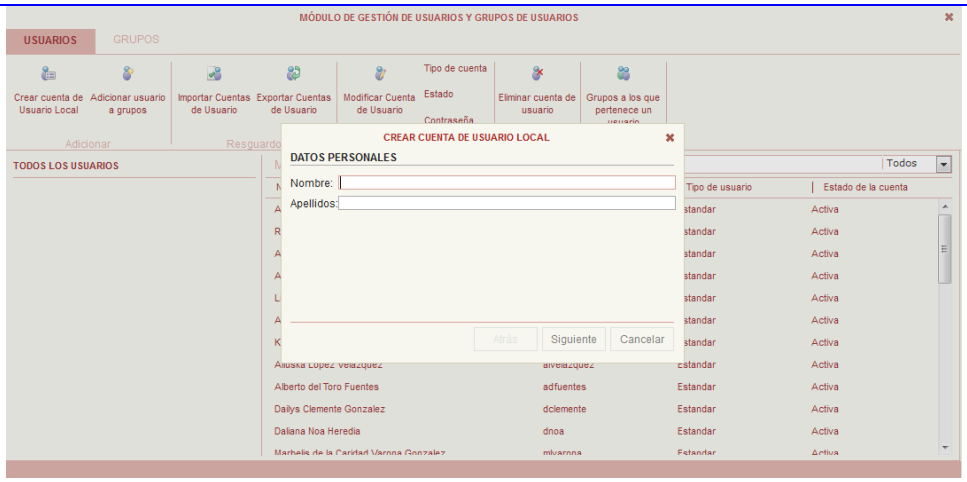
A continuación se expone la descripción del caso de uso Gestionar cuentas de usuario identificado como uno de los casos de uso arquitectónicamente significativos en el sistema. El resto de las descripciones podrá consultarlas en el expediente de proyecto correspondiente al módulo.

Objetivo	Gestionar cuentas de usuario	
Actores	Administrador	
Resumen	El caso de uso comienza cuando el administrador desea realizar alguna de estas acciones: Crear, Modificar o Eliminar cuentas de usuario. El caso de uso termina cuando realiza alguna de estas acciones.	
Complejidad	Alta	
Prioridad	Media	
Precondiciones	El administrador debe estar autenticado en el sistema. Para las secciones eliminar y modificar debe existir al menos una cuenta registrada en el sistema.	
Postcondiciones	Se almacenan los datos de la cuenta creada o modificada y en cuanto a la cuenta eliminada desaparece de manera permanente del sistema.	
Flujo de eventos		
Sección 1: “Crear cuenta de usuario”		
	Actor	**Sistema**
1.	El administrador accede a la opción crear cuentas de usuario.	2. El sistema muestra una nueva interfaz con los campos requeridos para crear la cuenta.
3.	El administrador completa los campos y accede a la opción “Crear”.	4. El sistema verifica que los campos hayan sido completados de manera correcta.
		5. El sistema notifica al administrador que la cuenta ha sido creada satisfactoriamente. Termina el caso de uso.
Flujos alternos		
Nº Evento “Datos incompletos”		
	Actor	**Sistema**
4.1		El sistema muestra un mensaje

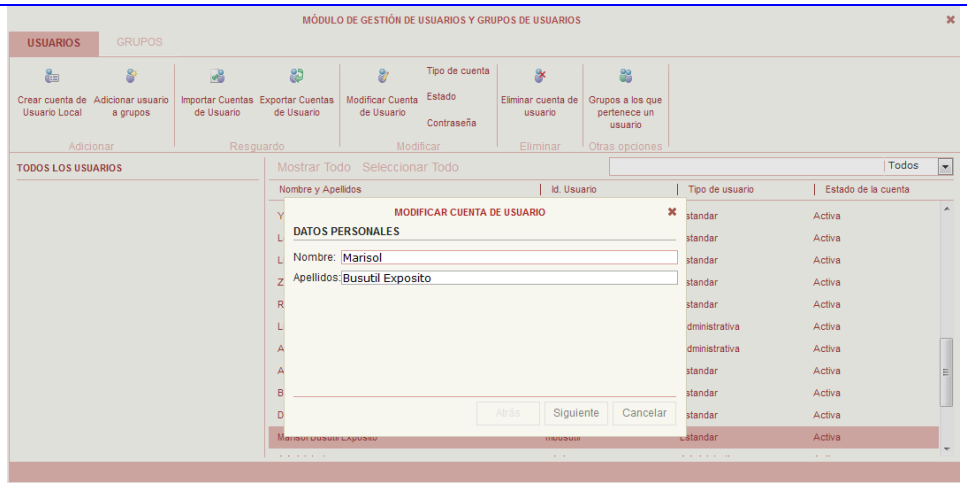
		indicando que debe completar todos los campos para crear la cuenta.
Nº Evento "Datos incorrectos"		
4.2		El sistema muestra un mensaje indicando que ha introducido tipos de datos incorrectos.
Sección 2: "Modificar cuenta de usuario"		
	Actor	**Sistema**
1.	El administrador selecciona el o los usuarios al que desea modificar y selecciona la opción Modificar cuenta.	2. El sistema muestra una interfaz con todos los campos modificables de la cuenta.
3.	El administrador modifica los datos que desea y accede al botón "Modificar".	4. El sistema verifica los datos modificados en caso que estén todos los campos llenos y con los caracteres indicados para cada campo, el sistema notifica que la cuenta ha sido modificada satisfactoriamente.
Flujos alternos		
Nº Evento "Datos incompletos"		
	Actor	**Sistema**
3.1		El sistema muestra un mensaje indicando que debe completar todos los campos para poder modificar la cuenta.
Nº Evento "Datos incorrectos"		
3.2		El sistema muestra un mensaje indicando que ha introducido tipos de datos incorrectos.
Sección 3: "Eliminar cuenta de usuario"		
	Actor	**Sistema**
1.	El administrador selecciona la cuenta o las cuentas que desea eliminar y accede a la opción eliminar.	2. El sistema muestra una interfaz preguntando si el administrador está seguro de eliminar la o las cuentas.
3.	El administrador accede a la opción "Aceptar".	4. El sistema elimina la cuenta de manera automática en el sistema y muestra un mensaje informando al

		administrador que la cuenta ha sido eliminada de manera satisfactoria.
Nº Evento "No eliminar cuenta"		
3.1	El administrador accede a la opción "Cancelar".	3.2 El sistema cancela la operación.
Nº Evento "Seleccionar al menos un usuario"		
3.3	El administrador no ha seleccionado ningún usuario para eliminar.	3.4 El sistema muestra un mensaje indicando que debe seleccionar al menos un usuario para poder eliminar.
Nº Evento "Selección de la cuenta de administrador"		
3.5	El administrador selecciona su propia cuenta para eliminarla.	3.6 El sistema muestra un mensaje advirtiendo que no puede eliminar su propia cuenta.
Relaciones	**CU Incluidos**	
	CU Extendidos	
Requisitos no funcionales	**RNF1, RNF2**	
Asuntos pendientes		

Prototipo de interfaz: "Crear cuenta de usuario"



Prototipo de interfaz: "Modificar cuenta de usuario"



Prototipo de interfaz: “Eliminar cuenta de usuario”

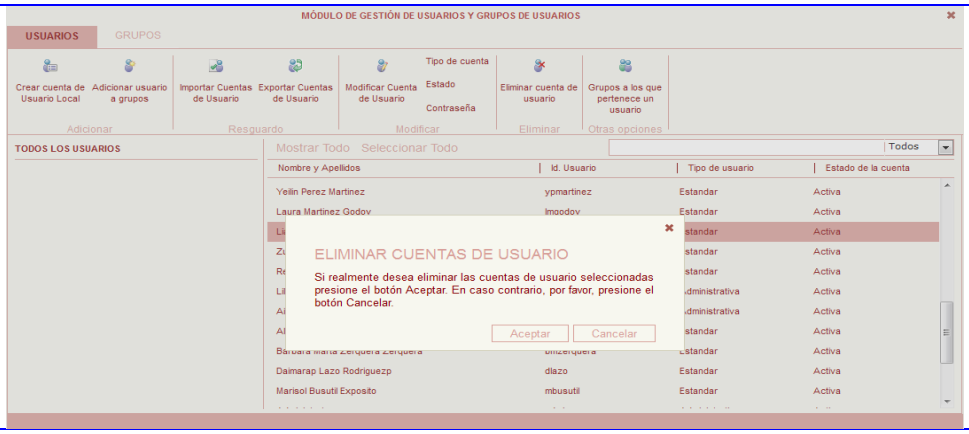


Tabla 3 Descripción del caso de uso Gestionar cuentas de usuarios

Capítulo 3. Diseño de la solución

En este capítulo serán desarrollados los aspectos relacionados con el diseño de la solución, se representa y describe la arquitectura del sistema eXcriba y sus módulos, así como la descripción de los componentes que la conforman y los patrones de diseño utilizados. Se refleja además los diagramas de clases del diseño utilizando estereotipos *web* y los diagramas de colaboración correspondientes al caso de uso gestionar cuentas de usuario.

3.1 Arquitectura del sistema

La arquitectura de *software* es la organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros, con el entorno y los principios que orientan su diseño y evolución (36).

El objetivo de la arquitectura de *software* es aportar elementos que ayuden a la toma de decisiones y, al mismo tiempo, proporcionar conceptos y un lenguaje común que permita la comunicación entre los equipos que participen en un proyecto. Para lograrlo la arquitectura de *software* construye abstracciones, materializándolas en forma de diagramas comentados (37).

La arquitectura basada en capas es un miembro de la familia de estilos de llamada y retorno, esta arquitectura se define como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior (39).

En la arquitectura multicapa las capas inferiores brindan servicios a las capas superiores (independientemente del nivel en que se encuentren). En esta arquitectura una capa solamente debe utilizar lo que la interfaz de la o las capas inferiores le brindan, de este modo se pueden intercambiar las capas respetando la interfaz, que viene a ser como un “contrato entre capas” (38).

Para el diseño del módulo Administración de usuarios y grupos se utilizó una arquitectura basada en capas o n-capas, por ser la arquitectura a la que responde el GDA eXcriba.

3.1.1 Análisis general de la Arquitectura del GDA eXcriba

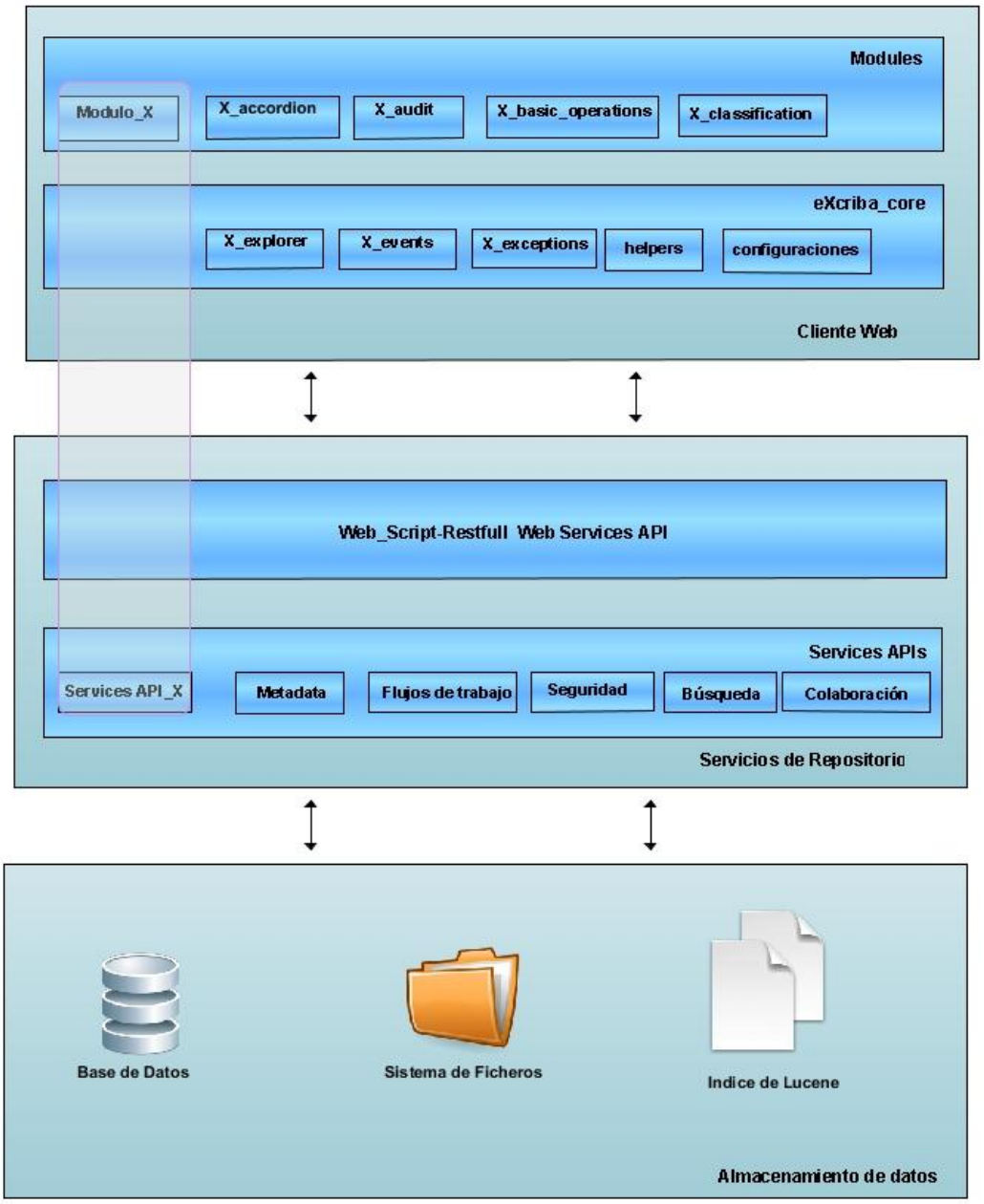


Figura 4 Vista de la arquitectura de eXcriba

Como se puede apreciar en la Figura 4 el GDA eXcriba se compone de tres capas arquitectónicas fundamentales (Cliente *Web*, Servicios de Repositorio, Almacenamiento

de Datos) que interactúan entre sí durante la realización de cada una de las funcionalidades que brinda el sistema.

La capa superior, “Cliente *Web*” se compone de un conjunto de módulos que dan soporte a las diferentes operaciones o funcionalidades con las que interactúan los diferentes usuarios del sistema. Su principal componente es el módulo “eXcriba-core” el cual brinda un conjunto de servicios, interfaces, configuraciones y funcionalidades genéricas de las cuales dependen (o pueden ser usadas) por el resto de los módulos que conforman la aplicación *web*.

La capa de servicios de acceso al repositorio “Servicios de Repositorio”, a través del *framework* WebScript provee un API completa de objetos de repositorio y servicios basados en el estilo arquitectónico REST. Estos servicios juegan un papel fundamental en la incorporación de las funcionalidades al sistema al tiempo que garantizan que desde el cliente *web* se pueda acceder a los datos que se almacenan en el repositorio de contenido.

La capa de almacenamiento de datos, la más baja de la arquitectura, constituye precisamente los cimientos sobre los cuales se construye o desarrolla el resto de la aplicación, esta capa se compone de la base de datos en la cual se almacena de manera estructurada toda la información referente a los usuarios y demás objetos de negocio.

3.1.1.1 Relación entre los componentes

En la Figura 4 se puede apreciar que cuando se implementa un nuevo módulo, Módulo X por ejemplo, este pasa a formar parte del subsistema “*modules*” dentro de la primera capa, “eXcriba *web client*” (cliente *web* eXcriba).

Por lo general, estos módulos hacen uso de algunas de las funcionalidades o configuraciones que provee el núcleo (eXcriba-*core*). Para acceder a los datos del repositorio se hace uso de los servicios que se encuentran disponibles en el subsistema “*services*” dentro de la capa “*Repository Services*”, servicios que en la mayoría de los casos hacen uso de las APIs que provee Alfresco para el desarrollo de determinadas funcionalidades.

3.1.2. Diseño de la arquitectura del módulo

El módulo para la gestión de usuarios y grupos responde a una arquitectura basada en capas, capa de presentación, capa de aplicación y capa de acceso a datos con la que se interactúa para obtener los datos que serán procesados por dicho módulo.

Capa de Presentación: esta capa está compuesta por los recursos básicos que serán tratados o procesados, generalmente por el cliente (navegador *web*) para generar interfaces que serán mostradas al usuario. Además en esta capa se encuentran las páginas HTML, hojas de estilo y ficheros JavaScript que se ejecutan del lado del cliente. En la capa de presentación el módulo Administración de usuarios y grupos hace uso del *framework* jQuery para el procesamiento de las interfaces de usuario a través del lenguaje JavaScript.

Capa de Aplicación: agrupa la lógica del negocio que se ejecuta del lado del servidor (en este caso Apache). En función del módulo que sea tratado, en esta capa se validan los datos que provienen del cliente, se ejecutan ciertas operaciones con dichos datos y delega el envío de los mismos al repositorio a través de las capas inferiores, así mismo, en determinados casos prepara los datos obtenidos de las capas inferiores para conformar las vistas a mostrar.

Para el desarrollo del módulo se hace uso en esta capa del subsistema de Control de Acceso, el cual es utilizado para validar la sesión de un usuario previamente autenticado, el subsistema de Gestión de Acciones y el de Gestión de Eventos interactúan entre sí para llevar a cabo las acciones y eventos que se desencadenan en el sistema una vez que el usuario ejecuta determinada funcionalidad y el subsistema de Comunicación con la Capa de Acceso a datos posibilita hacer la llamada a los servicios que se ofrecen en la misma.

Capa de Acceso al Repositorio: en esta capa se encuentran las interfaces encargadas de la comunicación con el repositorio de Alfresco permitiendo interactuar con los servicios en el mismo, dichos servicios se explicarán en detalle en apartados siguientes. El subsistema Servicios para la Administración de usuarios y grupos contiene un conjunto de funcionalidades que serán ofrecidas al usuario a través del módulo.

3.1.3 Relación entre los componentes del módulo

Como se puede observar, en la figura 5 existe una relación entre cada una de las capas representadas por una flecha bidireccional. La capa de presentación solicita datos a la capa de aplicación, la misma devuelve dichos datos o dado el caso construye interfaces que han sido requeridas al igual que los datos. En el caso de la capa de aplicación solicita de igual forma datos a la capa de acceso a datos y esta última los devuelve a la capa superior. Es de esta forma que se ve la relación entre las diferentes capas y la intención de la flecha bidireccional que las relaciona.

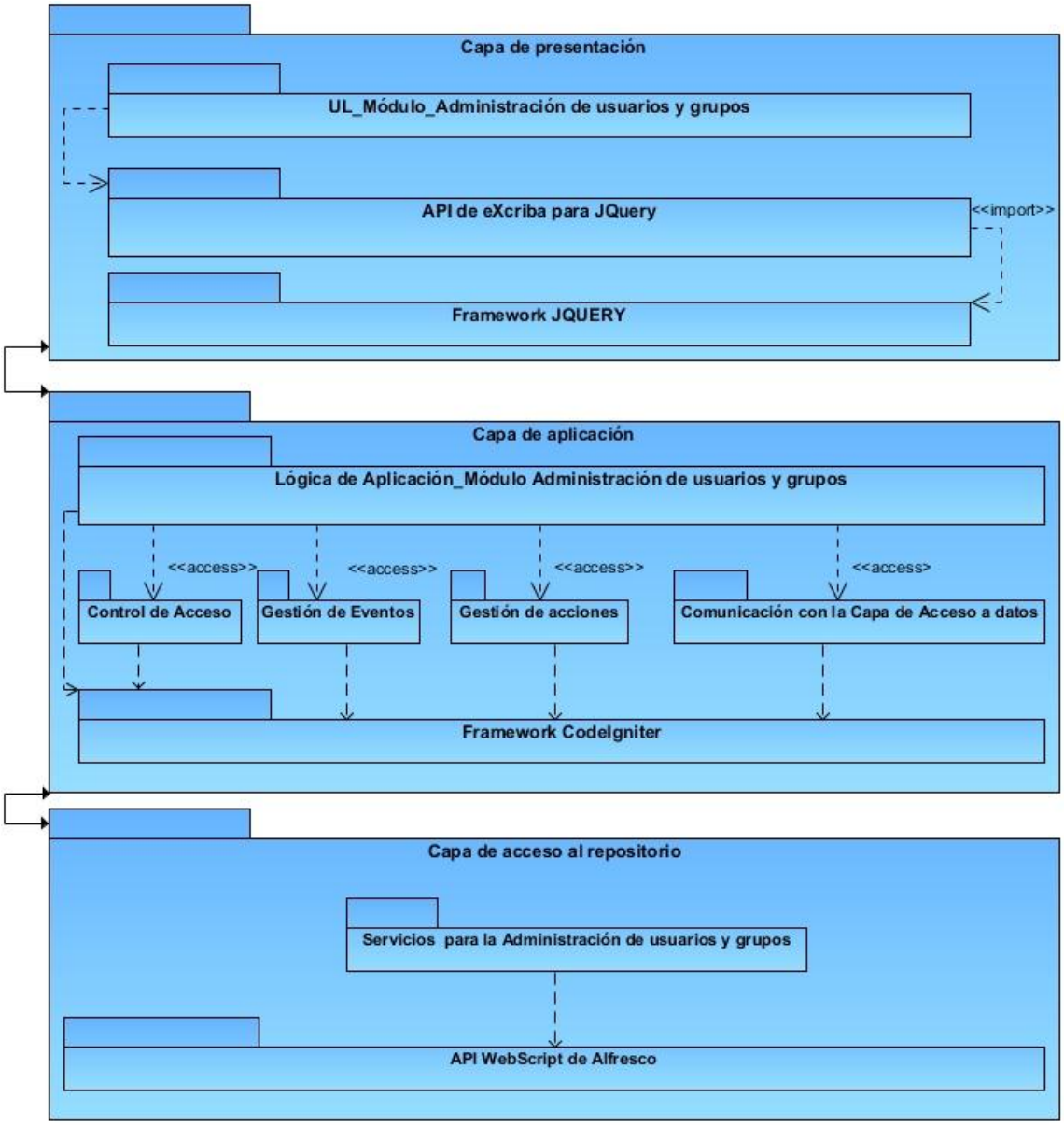


Figura 5 Vista en subsistemas de la arquitectura del módulo Administración de usuarios y grupos

Como se puede observar, en la figura 5 existe una relación entre cada una de las capas representadas por una flecha bidireccional. La capa de presentación solicita datos a la capa de aplicación, esta última devuelve dichos datos, para ello solicita de igual forma datos a la capa de acceso a datos y esta última los devuelve a la capa superior. Es de

esta forma que se ve la relación entre las diferentes capas y la intención de la flecha bidireccional que las relaciona.

3.2 Patrones de diseño

Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas (17). En la presente investigación se utilizaron los patrones GRASP, los cuales serán referenciados en este epígrafe. GRASP es un acrónimo que significa General *Responsibility Asignment Software Patterns* (Patrones Generales de Software para Asignar Responsabilidades), donde una responsabilidad es un método en los artefactos UML, por lo que los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones. Esta familia está compuesta por nueve patrones: experto, creador, alta cohesión, bajo acoplamiento, controlador, polimorfismo, fabricación pura, indirección y no hables con extraños (17). A continuación serán explicados de manera sintetizada los patrones que fueron utilizados para el diseño del módulo.

3.2.1 Experto

El patrón Experto es el que se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad, nótese que el cumplimiento de una responsabilidad requiere a menudo información distribuida en varias clases de objetos por lo que pueden existir varios expertos parciales que colaboren con la tarea (17).

La clase Gestionar_Usuarios es el experto en la creación de cuentas de usuario pues la misma contiene la información suficiente para la creación de una cuenta así como el método asociado a dicha funcionalidad. La clase Crear_Cuenta_js es un experto parcial en este caso pues suministra datos a la clase Gestionar_Usuarios para la creación de cuentas de usuario.

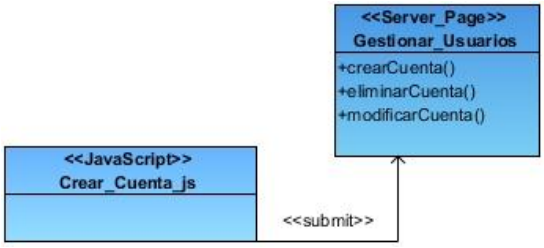


Figura 6 Ejemplo del patrón Experto

3.2.2 Controlador

Un controlador es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema y define para ello el método de su operación. La mayor parte de los sistemas reciben eventos de entrada externa, los cuales generalmente incluyen una interfaz gráfica para el usuario operado por una persona, para manejar estos eventos de entrada se eligen los controladores (17).

En este caso la clase Gestionar_Usuarios contiene casi todos los métodos correspondientes a las funcionalidades que ofrece el sistema, algunas de ellas son:

- crear, modificar y eliminar cuentas de usuario
- crear y eliminar grupos de usuarios
- exportar e Importar cuentas de usuario
- obtener los grupos a los que pertenece un usuario

Figura 7 Ejemplo del patrón Controlador

3.3 Diseño del caso de uso gestionar cuentas de usuario

Durante el diseño del caso de uso gestionar cuentas de usuario se identificaron y definieron las clases del diseño necesarias para llevar a cabo las funcionalidades

asociadas al mismo. Además se refleja mediante diagramas de clases del diseño y diagramas de interacción las relaciones entre las clases. Asimismo se describen tanto las clases como los servicios que deben implementarse o usarse para la completa realización del caso de uso.

3.3.1 Diagrama de clases del diseño

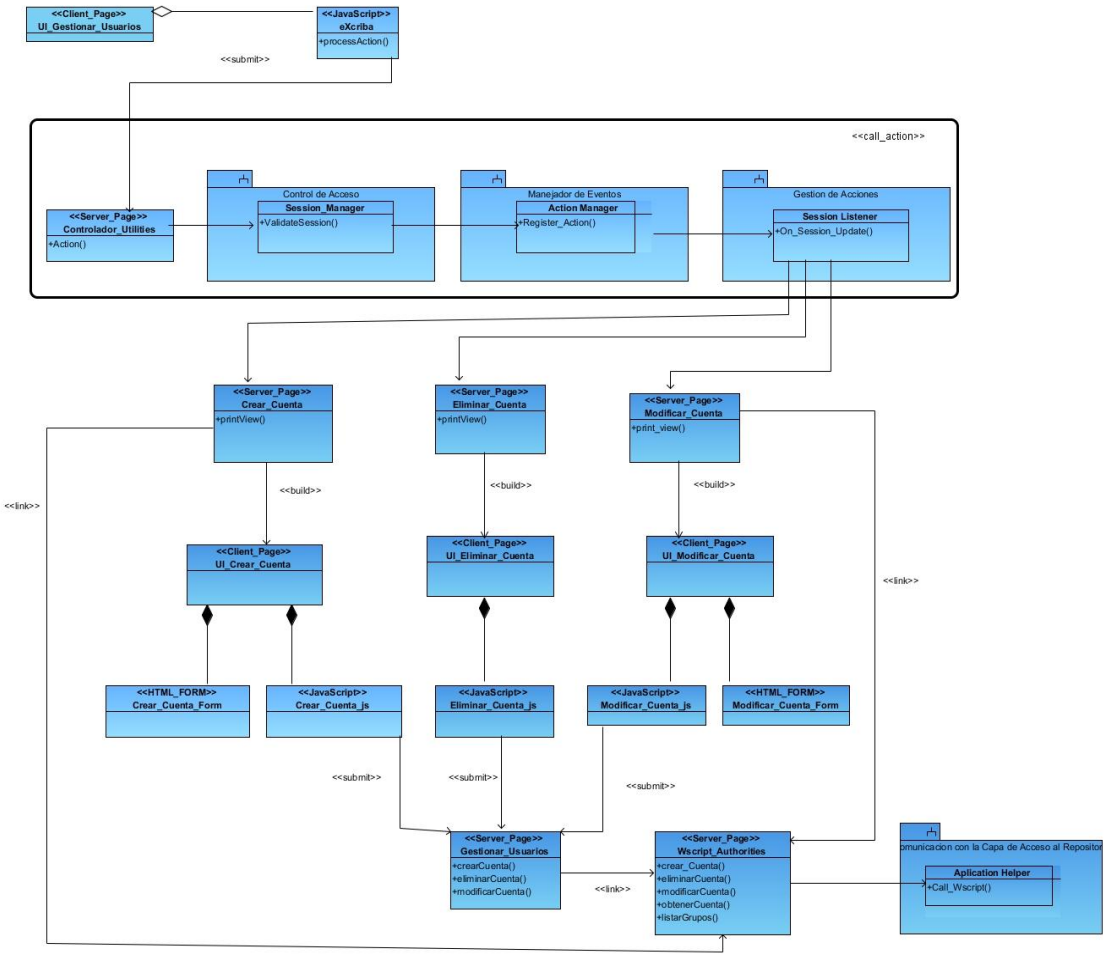


Figura 8 Diagrama de clases del caso de uso Gestionar cuentas de usuario

3.3.2 Descripción de las clases del diseño

A continuación se exponen en forma de tablas las principales clases que intervienen en el caso de uso Gestionar cuentas de usuario, así como las funcionalidades identificadas como exclusivamente necesarias para la realización del mismo.

Nótese que no se describen las clases que son comunes para las tres secciones (crear, eliminar y modificar cuentas de usuario) como es el caso de las clases controladoras Gestionar_Usuarios, Wscript_Authorities y la clase interfaz UI_Gestionar_Usuarios.

Nombre: UI_Gestionar_Usuarios
Tipo de clase: Interfaz de usuario
Descripción: Es la interfaz que se le muestra inicialmente al usuario mediante la cual el mismo puede acceder a la funcionalidad que necesite.

Tabla 4 Descripción de la clase UI_Gestionar_Usuarios

Nombre: Crear_Cuenta	
Tipo de clase: página servidora	
Descripción: Es la clase que construye la interfaz correspondiente a la funcionalidad crear cuenta de usuario.	
Responsabilidades:	
Nombre:	Descripción:
printView()	Construye y genera la vista donde el usuario podrá visualizar los campos necesarios para crear una cuenta.

Tabla 5 Descripción de la clase Crear_Cuenta

Nombre: Eliminar _Cuenta	
Tipo de clase: página servidora	
Descripción: Es la clase que construye la interfaz correspondiente a la funcionalidad eliminar cuenta de usuario.	
Responsabilidades:	
Nombre:	Descripción:
printView()	Construye y genera una interfaz de confirmación para eliminar una cuenta.

Tabla 6 Descripción de la clase Eliminar_Cuenta

Nombre: Modificar _Cuenta	
Tipo de clase: página servidora	
Descripción: Es la clase que construye la interfaz correspondiente a la funcionalidad modificar cuenta de usuario.	
Responsabilidades:	
Nombre:	Descripción:
printView()	Construye y genera la vista donde el usuario podrá modificar los datos asociados a la cuenta de usuario.

Tabla 7 Descripción de la clase Modificar_Cuenta

Nombre: Gestionar _Usuarios	
Tipo de clase: controladora	
Descripción: Contiene los métodos necesarios para interactuar con la capa de Acceso al Repositorio. Su función es actuar como la interfaz a través de la cual se comunica la capa de aplicación/negocio (específicamente las funcionalidades del módulo Administración de usuarios y grupos) con la de Acceso al Repositorio.	
Responsabilidades:	
Nombre:	Descripción:
crearCuenta	Coordina las operaciones necesarias que deben llevarse a cabo para crear una cuenta.
eliminarCuenta	Coordina las operaciones necesarias para la eliminación de una cuenta.
modificarCuenta	Coordina las operaciones necesarias que deben llevarse a cabo para modificar una cuenta.

Tabla 8 Descripción de la clase Gestionar_Usuarios

Nombre: *Wscript_Authorities*	
Tipo de clase: controladora	
Descripción: Recibe los datos necesarios para hacer las llamadas a los servicios.	
Responsabilidades:	
Nombre:	Descripción:
crearCuenta	Se encarga de hacer la llamada al servicio crear cuenta de usuario.
eliminarCuenta	Se encarga de hacer la llamada al servicio crear eliminar cuenta de usuario.
modificarCuenta	Se encarga de hacer la llamada al servicio modificar cuenta de usuario.

Tabla 9 Descripción de la clase Wscript_Authorities

Para la realización del caso de uso gestionar cuentas de usuario se propone la implementación de los servicios que se describen en las tablas que aparecen a continuación:

Servicio: Crear cuenta de usuario	
URL:	/cu/uci/excriba/*authorities*/person/{*userName*}
Método HTTP:	POST
Lenguaje de implementación:	JavaScript
Parámetros o argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
userName: identificador del usuario a crear	mbusutil
person: objeto en formato json que contiene los datos de la personales del usuario al cual se le está creando la cuenta	{“propiedad”: “valor”, “propiedad”:“valor”, ….}
user: objeto en formato json que contiene los datos asociados a la cuenta de usuario	{“propiedad”: “valor”, “propiedad”:“valor”, ….}
groups: objeto en formato json que contiene el listado de los grupos en los que se desea insertar el nuevo usuario	[“GROUP_A”, “GROUP_B”]
Parámetros o argumentos de salida	
errores: almacena los mensajes de error que ocurren durante la ejecución del servicio	“Debe especificar un nombre de usuario” “Ya existe el usuario mbusutil”
resultado: contiene el resultado del servicio	“OK”
advertencias: almacena mensajes que indican que una acción no pudo realizarse	“No existe el grupo GROUP_A”

pero que no afecta el correcto funcionamiento del servicio	

Tabla 10 Descripción del servicio “Crear cuenta de usuario”

Servicio: Eliminar cuenta de usuario	
URL:	/cu/uci/excriba/authorities/person/{userName}
Método HTTP:	DELETE
Lenguaje de implementación:	JavaScript
Parámetros o argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
userName: identificador del usuario a eliminar	mbusutil
userNames: objeto en formato json que contiene el listado de los usuarios que serán eliminados	[“mbusutil”, “relejalde”, “mlvarona”]
Parámetros o argumentos de salida	
errores: almacena los mensajes de error que ocurren durante la ejecución del servicio	“Debe especificar el usuario que desea eliminar”
resultado: recoge el resultado del servicio	“OK”
advertencias: almacena mensajes que indican que una acción no pudo realizarse pero que no afecta el correcto funcionamiento del servicio.	“El usuario que desea eliminar no existe”

Tabla 11 Descripción del servicio “Eliminar cuenta de usuario”

Servicio: Obtener cuenta	
URL:	/cu/uci/excriba/authorities/person/{userName}
Método HTTP:	GET
Lenguaje de implementación:	JavaScript
Parámetros o argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
username: identificador del usuario a eliminar	mbusutil
Parámetros o argumentos de salida	
respuesta: objeto en formato json que contiene los datos de la cuenta que se quiere obtener	{ “propiedad”: “valor”, “propiedad”:“valor”, ….}

Tabla 12 Descripción del servicio “Obtener cuenta de usuario”

Servicio: Modificar cuenta	
URL:	/cu/uci/excriba/authorities/person/{userName}

Método HTTP:	PUT
Lenguaje de implementación:	JavaScript
Parámetros o argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
username: identificador del usuario a eliminar	mbusutil
Parámetros o argumentos de salida	
respuesta: objeto en formato json que contiene los datos de la cuenta que se quiere obtener	{“propiedad”: “valor”, “propiedad”:“valor”, ….}

Tabla 13 Descripción del servicio “Modificar cuenta de usuario

3.3.3 Diagramas de colaboración del diseño

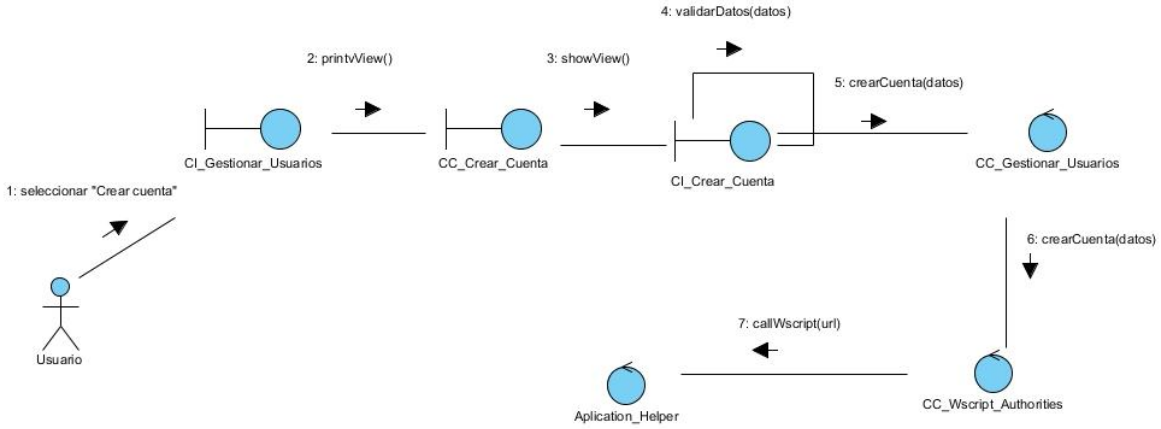


Figura 9 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Crear cuentas de usuario”

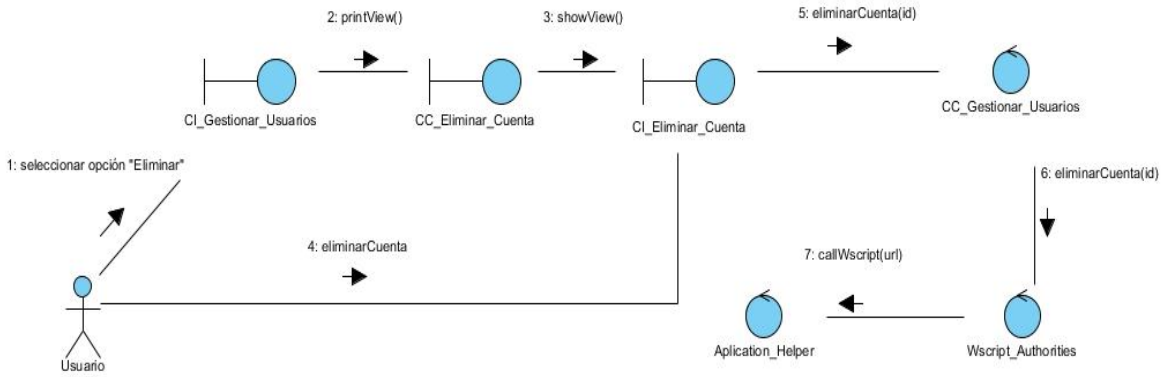


Figura 10 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Eliminar cuentas de usuario”

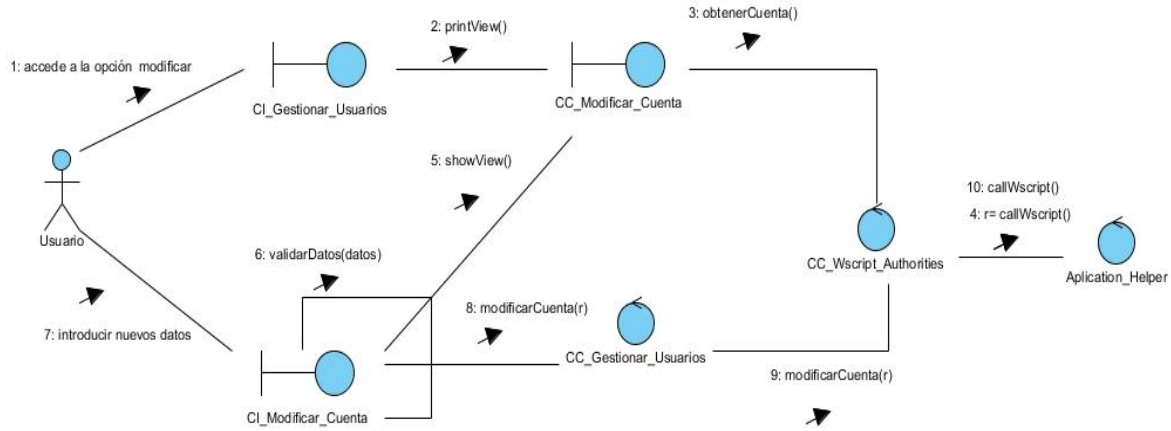


Figura 11 Diagrama de colaboración del caso de uso Gestionar cuentas de usuario, sección: “Modificar cuentas de usuario”

Para consultar los restantes artefactos que se generaron para el caso de uso gestionar grupos consultar Anexo E, el resto de los diagramas los podrá consultar en el expediente de proyecto correspondiente al módulo.

Capítulo 4. Implementación y Prueba de la Solución

Este capítulo describe cómo los elementos del modelo de diseño son implementados en términos de componentes y la manera en que se organizan de acuerdo con los nodos referidos en el modelo de despliegue. Se expone el nivel, tipo de prueba y técnica a los que se someterá el módulo.

4.1 Implementación del módulo

La implementación de un sistema parte con el resultado del diseño y se construye el sistema en términos de componentes, es decir ficheros de código, ejecutables, entre otros. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto. El propósito principal de la implementación es desarrollar la arquitectura y el sistema como un todo (40).

4.1.1 Diagrama de componentes

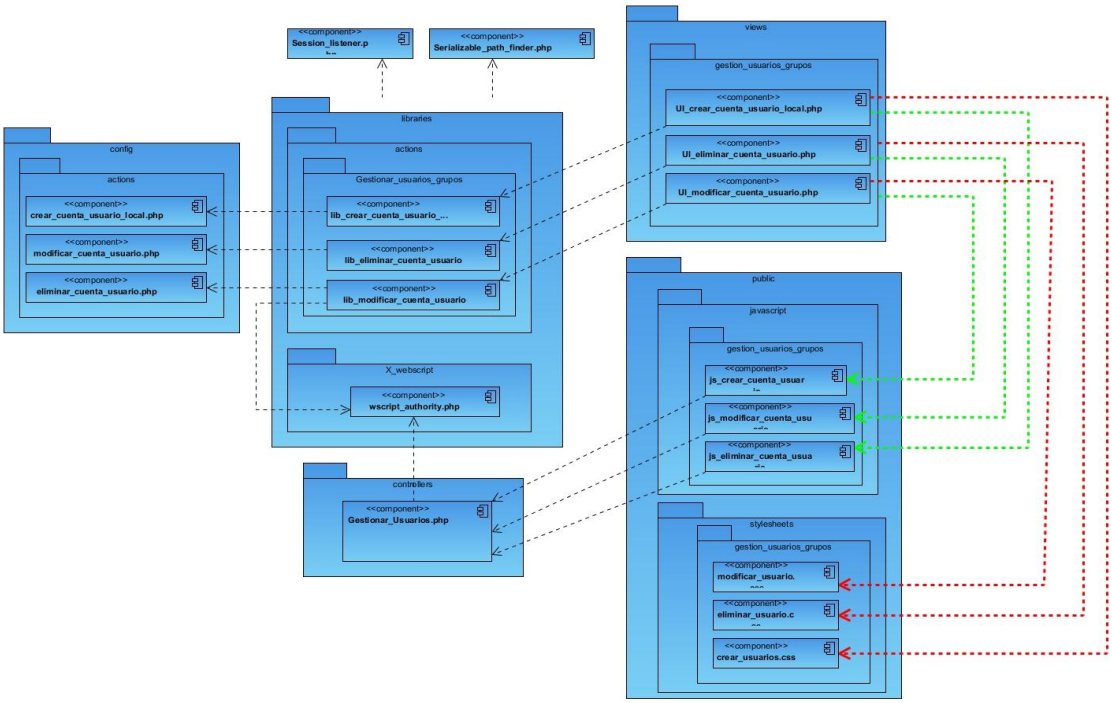


Figura 12 Diagrama de componentes correspondiente al caso de uso Gestionar cuentas de usuario

En el flujo de implementación se modelaron los diagramas de componentes para estructurar el modelo de implementación en términos de subsistemas y mostrar las relaciones entre dichos elementos. De manera general su emplearon para describir los elementos físicos del sistema y sus relaciones.

Descripción del diagrama de componentes

El módulo *x-authority-manager* contiene todos los componentes agrupados en dos paquetes fundamentales: php y resource. El paquete php conteniene todos los componentes que se procesan del lado del servidor mientras que el paquete resource contiene los componentes que se procesan del lado del cliente.

Dentro del paquete php se encuentra el componente config, donde se definen las configuraciones para ejecutar las acciones o funcionalidades que brinda el módulo, en el componente libraries se encuentran las clases encargadas de construir las vistas asociadas a cada una de las funcionalidades del módulo (paquete actions) y además las clases encargadas de hacer las llamadas a los servicios *web* (paquete x-webscript).

Además en el paquete php se encuentran el componente views, que contiene las interfaces visuales que se mostrarán al usuario en el navegador y el componente controllers que contiene la clase GestionarUsuarios la cual coordina la ejecución de las diferentes operaciones que se pueden llevar a cabo en el módulo.

En el componente resource se encuentran los paquetes images, javascript y css. En images se encuentran las imágenes utilizadas para decorar las interfaces del módulo, javascript contiene los script que serán ejecutados en el cliente (navegador *web*) durante la interacción del usuario con las interfaces que brinda el módulo y el componente css contiene las hojas de estilo que se le aplican a las interfaces que serán mostradas al usuario.

4.1.2 Diagrama de despliegue

Los diagramas de despliegue describen la topología de un sistema así como los elementos de *hardware* y *software* que ejecuta cada uno de ellos, además representan los nodos y sus relaciones. Los nodos son objetos físicos en tiempo de ejecución que representan un recurso computacional, generalmente con memoria y capacidad de

procesamiento, estos son conectados por asociaciones de comunicación tales como enlaces de red y conexiones TCP/IP, entre otros.

Para cada componente de un diagrama es necesario documentar las características técnicas requeridas para la ejecución del mismo (41).

A continuación se muestra representado una manera en que puede ser representada la distribución de los componentes de eXcriba incluyendo los componentes del módulo.

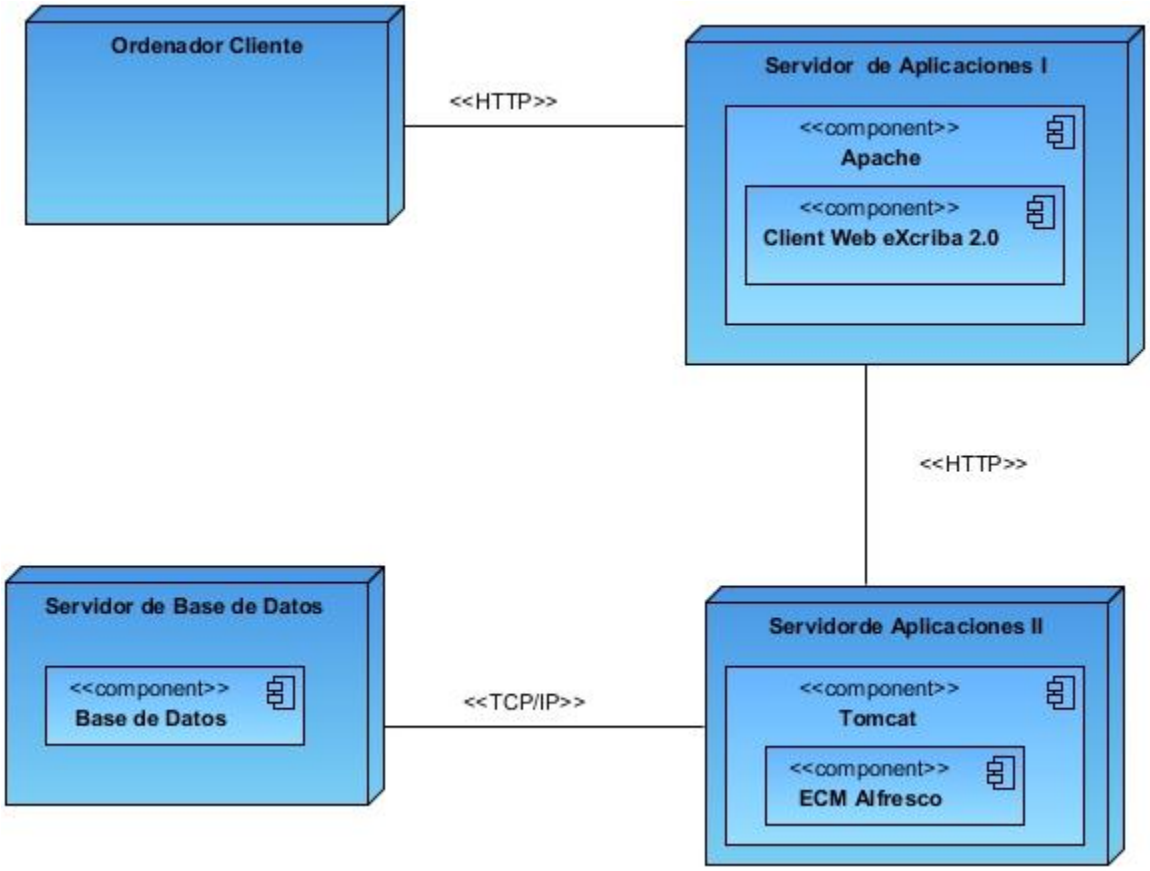


Figura 13 Diagrama de despliegue

Descripción de los nodos

- Nodo **Ordenador Cliente:** en este nodo físico estará alojado el navegador *web* (Internet Explorer, Mozilla Firefox, Chrome, entre otros) con el que interactúa el usuario para acceder a las interfaces del módulo.
- Nodo **Servidor de Aplicaciones I:** en este nodo es desplegado el cliente *web* de eXcriba en un servidor Apache con PHP 5.3 o superior y debe estar activado el módulo php_curl, usado en el contexto del módulo para el acceso a los servicios. Para el caso particular de este módulo, en este nodo se encuentran los componentes correspondientes a la capa de negocio mostrados en el diagrama de despliegue.
- Nodo **Servidor de Aplicaciones II:** en este nodo está desplegado el servidor de aplicaciones Tomcat en el cual se despliega el ECM Alfresco, repositorio de contenidos y servicios del GDA eXcriba. En este nodo se encuentran desplegados los servicios que serán usados para brindar funcionalidades al módulo. En este nodo debe estar instalada la máquina virtual de java.
- Nodo **Servidor de Base de Datos:** es donde se encuentra la Base de Datos que utiliza el ECM Alfresco para almacenar de forma estructurada información asociada con la gestión de los contenidos y otros objetos de negocio como son los datos de los usuarios y los grupos.

Descripción de las relaciones entre los nodos

Inicialmente el usuario interactúa con el cliente *web* del GDA eXcriba a través de la comunicación que se establece mediante el protocolo HTTP, entre su ordenador y el Servidor *de* Aplicaciones I, luego este último establece una conexión con el Servidor de Aplicaciones II mediante el protocolo HTTP para obtener los datos solicitados anteriormente por el usuario en su interacción con la aplicación, para ello el Servidor de Aplicaciones II mediante el protocolo TCP/IP accede a la información almacenada en la Base de Datos dando respuesta a la petición realizada por este servidor.

4.2 Prueba

La calidad es un factor importante en el proceso de desarrollo de *software* y el instrumento justo para garantizar dicha calidad son las pruebas de *software*. La prueba

de *software* es aquella actividad encaminada a evaluar las capacidades de un programa o sistema así como comprobar si cumple con los resultados esperados (42).

Nivel de Sistema

El módulo Administración de usuarios y grupos será probado en el nivel de sistema pues este tiene como propósito ejercitar profundamente el sistema para verificar que se han integrado adecuadamente todos los elementos del mismo (*hardware*, otro *software*) y que realizan las funciones adecuadas. Concretamente se quería comprobar que se cumplen los requisitos funcionales establecidos.

Para la generación de casos de prueba de sistema se utilizan técnicas de caja negra. Este tipo de pruebas se suelen hacer inicialmente en el entorno del desarrollador, denominadas Pruebas Alfa, y seguidamente en el entorno del cliente denominadas Pruebas Beta (53). Estas pruebas finalizan cuando se han cumplido los meses o semanas estimadas para realizarlas y se han hallado N errores.

Prueba de caja negra

Las pruebas de caja negra, también denominadas pruebas de comportamiento se centran en los requisitos funcionales del *software* (18) y se llevan a cabo sobre la interfaz del sistema, obviando el comportamiento interno y la estructura del programa. El objetivo de esta prueba es verificar que el sistema cumpla con los requisitos funcionales establecidos por el cliente. En la prueba de caja negra los casos de prueba pretenden demostrar que las funciones del *software* son operativas, que la entrada se acepta de forma adecuada, que se produce una salida correcta y que la integridad de la información externa se mantiene.

Con el objetivo de demostrar que el módulo cumple con los requisitos que han sido definidos por el cliente reflejados en el (epígrafe 2.3.2), se utiliza las pruebas de caja negra para mejorar la calidad de dicho módulo al descubrir sobre su interfaz la mayor cantidad posible de errores y en base a estos, proceder a su corrección.

Las pruebas de caja negra han sido aplicadas para detectar (18):

- funciones incorrectas o faltantes
- errores de interfaz
- errores de comportamiento o desempeño

Técnica de partición equivalente

Para desarrollar las pruebas de caja negra es posible aplicar varias técnicas, entre ellas:

- técnica de la partición de equivalencia: esta técnica divide el campo de entrada en clases de datos que tienden a ejercitar determinadas funciones del *software*
- técnica del análisis de valores límites: esta técnica prueba la habilidad del programa para manejar datos que se encuentran en los límites aceptables
- técnica de grafos de causa-efecto: es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones

La técnica de partición de equivalencia es considerada una de las técnicas más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el *software*, esta técnica se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (18).

Para la realización de las pruebas del módulo Administración de usuarios y grupos se generó el artefacto “Diseño de casos de pruebas basado en casos de uso”. A continuación se muestra como ejemplo el caso de prueba realizado al caso de uso gestionar cuentas de usuario, especificando para ello las condiciones que deben cumplirse para su ejecución, la respuesta emitida por el sistema para cada escenario expuesto y los pasos fundamentales a seguir para ejecutar la acción.

Escenario	**Respuesta del sistema**
EC 1.1 Crear cuenta de forma correcta	Se crea una nueva cuenta de usuario en el sistema
EC 1.2 Campos incompletos	Se notifica al administrador que debe completar todos los campos obligatorios para poder crear la cuenta.
EC 1.3 Campos incorrectos	Se notifica al administrador que ha introducido tipos de datos incorrectos en un campo o varios, ejemplo: en un campo nombre ha introducido números.

EC 1.4 Cuenta existente	Se le notifica al administrador que la cuenta que intenta crear ya existe pues ya ha sido registrado un usuario con ese nombre de usuario.

Tabla 14 CP1_CU Gestionar cuentas de usuario. Crear cuentas de usuario

Sección: Modificar cuentas de usuario		
Escenario	**Respuesta del sistema**	**Flujo central**
EC 1.1 Modificar cuenta satisfactoriamente	Se guarda la cuenta con las modificaciones.	• Autenticarse en el sistema • Seleccionar la cuenta que desea modificar. • Seleccionar la opción Modificar cuenta de usuario local. • Completar los campos que desea modificar. • Presionar el botón modificar para que la cuenta sea guardada con las modificaciones.
EC 1.2 Campos incompletos	Se notifica al administrador que no debe dejar campos en blanco para poder modificar la cuenta.	
EC 1.3 Campos incorrectos	Se notifica al administrador que ha introducido tipos de datos incorrectos en un campo o varios, ejemplo: en un campo nombre ha introducido números.	
EC 1.4 Selección previa de una cuenta	Se le notifica al administrador debe seleccionar una cuenta para poder modificarla.	

Tabla 15 CP1_CU Gestionar cuentas de usuario. Modificar cuentas de usuario

Escenario	**Respuesta del sistema**
EC 1.1 Eliminar cuenta	Se elimina la cuenta del sistema.
EC 1.2 Selección de la cuenta a eliminar	Se notifica al administrador que debe seleccionar previamente la cuenta que desea eliminar.
EC 1.3 Eliminar cuenta	Se notifica al administrador que no puede eliminar su

propia	propia cuenta.
EC 1.4 Eliminar cuenta en uso	El administrador elimina la cuenta independientemente si esta está siendo usada o no.

Tabla 16 CP1_CU Gestionar cuentas de usuario. Eliminar cuentas de usuario

Para consultar el resto de los casos de prueba diseñados para cada funcionalidad del sistema, consultar el expediente de proyecto correspondiente al módulo “Administración de usuarios y grupos”.

4.2.1 Resultados de las pruebas realizadas

La realización de pruebas al módulo permitió detectar varias no conformidades en las dos primeras iteraciones siendo estas resueltas, dichas no conformidades fueron resueltas durante la propia iteración. Con la conclusión de esta fase de pruebas, fue posible comprobar que el módulo cumple con las especificaciones que se trazaron en los requisitos definidos por el cliente en el capítulo dos. A continuación se muestra la relación de no conformidades (detectadas y resueltas) por iteración:

Iteración	NC detectadas	Asociadas a	NC resueltas
I	16	Errores de interfaz y de funcionalidades	16
II	9	Errores de interfaz y de validación	9
III	0	-	-

Tabla 17 Resultado de las pruebas

Como se puede apreciar en la tabla anterior se realizaron tres iteraciones de pruebas. A lo largo de la primera iteración se detectaron 16 no conformidades asociadas a errores de interfaz y de funcionalidades incorrectas de las cuales fueron resueltas todas en la propia iteración. En la segunda iteración fueron detectadas 9 no conformidades asociadas a errores de interfaz y validación siendo satisfactoriamente solucionadas, en una tercera y última iteración no fueron detectadas no conformidades por lo que la aplicación mostró un perfecto funcionamiento.

Conclusiones

Como resultado de la investigación realizada se desarrolló un módulo para la gestión de usuarios y grupos desde la interfaz *web* del GDA eXcriba, concluyendo:

- el análisis realizado a diferentes sistemas de gestión documental permitió identificar la necesidad de desarrollar un módulo para la administración de usuarios y grupos desde la propia interfaz *web* de eXcriba, así como un conjunto de funcionalidades que deberían ser incorporadas al mismo
- el análisis de la arquitectura y la aplicación de patrones de diseño permitieron comprender los aspectos relacionados con los requisitos del sistema, descomponer los trabajos de implementación en partes más manejables creando así una entrada apropiada y un punto de partida para la implementación
- la implementación del módulo para la administración de usuarios y grupos constituye el punto de partida para el control de acceso en el Gestor de Documentos Administrativos eXcriba
- la aplicación de pruebas de caja negra a la propuesta de solución propiciaron la correcta validación de sus funcionalidades considerándose lista para ser desplegada en un entorno real

Recomendaciones

Se recomienda:

- Incorporar al módulo la funcionalidad Importar y Exportar cuentas de usuario desde el directorio activo LDAP.
- Incorporar la sincronización automática con el repositorio.

Referencias Bibliográficas

1. InformaticaHoy Team´s. Que son los sistemas de gestión documental. [En línea]. INFORMATICAHOY. 2007. [Consultado el: 16 de noviembre de 2012.] Disponible en: http://www.informatica-hoy.com.ar/aprender-informatica/Que-son-los-sistemas-de-gestion-empresarial.php.
2. Anon. Concepto de usuario [En línea].Definición abc tu diccionario hecho fácil. 2007-2013. [Citado el: 18 de marzo de 2013.] Disponible en:http://www.definicionabc.com/tecnologia/usuario.php
3. Mastermagazine team´s. Definición de password. [En línea]. Mastermagazine. 2007. [Consultado el: 18 de marzo de 2013.] Disponible en: http://www.mastermagazine.info/termino/6239.php.
4. Red Hat, Inc. Concepto de grupos de usuario. [En línea]. Red Hat Enterprise Linux 4, Introducción a la administración de sistemas. 2005. [Consultado el: 15 de enero de 2013.] Disponible en:http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/ch-acctsgrps.html.
5. Microsoft., Diccionario de informática e Internet de. *Diccionario de informática e Internet de Microsoft.* España: McGranw-Hill Interamericana, 2000. 0-7356-0615-3.
6. PricewaterhouseCoopers y PwC team´s. Administración de identidades y accesos. [En línea]. PWC 2011-2012. [Consultado el: 16 de septiembre de 2012.] Disponible en:http://www.pwc.com/co/es/administracion-de-identidades-y-accesos/index.jhtml.
7. MONTOYA, JOSÉ A. y RESTREPO, R. Gestión de identidades y control de acceso desde una perspectiva organizacional*.* Ing. USBMed*,* 2012*,* Vol. 3: p. 23-34.
8. RED HAT, INC. Nombre de usuario. [En línea]. Red Hat Enterprise Linux 4, Introducción a la administración de sistemas. 2005. [Consultado el: 20 de enero de 2013.] Disponible en: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/ch-acctsgrps.html.
9. Desarrollado por el Centro Nacional de Cálculo. Políticas para cuentas de usuario del sistema. [En línea]. Sistema Institucional de Seguimiento y Actualización para Egresados SISAE. [Consultado el: 23 de enero de 2013.] Disponible en: http://sistemas.cenac.ipn.mx/SISAE/politicas/CuentasUsuario.do;jsessionid=9055CB4D1495AB980795632E620113DC.

10. *Oracle® Fusion Middleware User's Guide for Oracle Identity Manager.* Oracle user guide Developer team´s. 2013.
11. NUXEO team´s. Administración de usuarios y grupos. [En línea] .NUXEO/ Content Management Platform for Business Aplications. 2013. [Citado el: 25 de enero de 2013.] Disponible en:http://www.nuxeo.com/en.
12. Knowledgetree team´s. Administración de usuarios y grupos. [En línea]. Knowledgetree. 2013. [Citado el: 5 de febrero de 2013.] Disponible en: www.Knowledgetree.com.
13. OpenKM team´s. Administración de usuarios y grupos. [En línea] .OpenKM.com 2011. [Citado el: 11 de febrero de 2013.] Disponible en:http://www.openkm.com/es/.
14. Alfresco Software Inc. Administración de usuarios y grupos. [En línea] .Alfresco 2013. [Citado el: 6 de febrero de 2013.] disponible en: http://wiki.alfresco.com/wiki/Alfresco_Explorer.
15. Shari, Munwar. *Alfresco Enterprise Content Management Implementation.* 2006.
16. JACOBSON, IVAR. *El Proceso Unificado de Desarrollo de Software.* La Habana: Ediciones Félix Varela, 2004.
17. LARMAN, CRAIG. *UML Y PATRONES: Introducción al análisis y diseño orientado a objetos.* Hispanoamericana, S.A: Prentice Hall, 1999. ISBN: 9701702611.
18. PRESSMAN, ROGER S. Ingeniería de Software, un enfoque práctico. Quinta edicion. Madrid, McGraw-Hill Companies, 2002.
19. FIELDING ROY, THOMAS. Architectural Styles and the Design of Network-based Software Architectures. Tesis doctoral, University of California, California, 2000.
20. RESTful Web services team´s.. [En línea] RESTful Web services: The basics [Citado en: noviembre de 2008]. Disponible en: http://www.ibm.com/developerworks/webservices/library/ws-restful/.
21. Alfresco team´s. Web_Scripts. [En línea] .Alfresco. 2013. [Citado el: 20 de diciembre de 2012.] Disponible en: http://wiki.alfresco.com/wiki/.
22. Freemarker team´s. Definición Freemarker. [En línea] FreeMarker: java template engine library – overview. 2013. [Citado Abril 2012]. Disponible en: http://freemarker.org/.
23. Json team´s. Definición de Json. [En línea] Introducing Json .2013. [Citado el: 18 de enero de 2013.] Disponible en: http://www.json.org/json-es.html.

24. DAMIÁN PEREZ VALDÉS. ¿Qué es JavaScript?. [En línea]. Maestros de la Web .2007.]. Disponible en: http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/

25. The jQuery Fundation. JavaScript . [En línea] .The jQuery Fundation, JavaScript library.2012. [Citado en: Mayo, 2013] Disponible en :http://jquery.com/.

26. Facultad de Informática de la Universidad de Las Palmas de Gran Canaria, ¿Que es html?. Conceptos básicos, 2012. 16 de mayo de 2013. Disponible en: http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/concepto.htm#html

27. W3C. Hojas de Estilo. [En linea]. W3C.es. 2013. [Citado en: mayo de 2013] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo.

28. SAETHER, S. Manual de php. [En línea] [Citado en: noviembre de 2002]. Disponible en: http://www.calitae.com/manuales/manual-php.pdf.

29. CodeIgniter team´s. Marco de trabajo Codeigniter. [En línea] .CodeIgniter.2012. [citado en: abril de 2013] Disponible en: http://codeigniter.com/.

30. Wobreference team´s. JavaScript (ServerSide). [En línea] Webreference.2013. [Citado el: 19 de marzo de 2013.] Disponible en: http://www.webreference.com/programming/javascript/rg37/index.html.

31. Ecured team´s. Eclipse,_entorno_de_desarrollo_integrado. [En línea] .Conocimiento con todos y para todos EcuRed .2010. Disponible en:http://www.ecured.cu/index.php/

32. IEEE. IEEE:Guide for Developing Requirements Specification. [En línea] 1998. ISBN: 1-55937-716-X.

33. *Ingeniería de Requisitos en Aplicaciones para la Web- Un estudio comparativo.* Escalona, Maria Jose y Koch, Nora. Universidad de Sevilla: s.n., 2002.

34. Grupo de Investigación eumed.net (SEJ-309).Requisitos funcionales. [En línea] Eumed.net [Citado el: 6 de 3 de 2013.] Disponible en: http://www.eumed.net/libros-gratis/2010b/698/Requisitos%20funcionales.htm.

35. Jacobson Ivar, Booch Grady, Rumbaugh James. *El Proceso Unificado de Desarrollo de Software.* Varela, Felix.*La habana.* 2004. págs. 125-163, 187-202.

36. IEEE, Computer Society.*recommended practice for architectural description of software intensive systems.* s.l. : Ieee std 1471-2000.

37. Desarrolloweb.com Teams. [En línea] Desarrolloweb.com Guiarte Multimedia S.L., 1999. [Citado el: 7 de 3 de 2013.] Disponible en: http://www.desarrolloweb.com/articulos/1622.php.

38. Dr. Roy Thomas Fielding. *Architectural Styles and the Design of Network-based Software Architectures.* sexta edición. s.l.: Addison Wesley, 2000. ISBN 0-599-87118-0.

39. David Garlan and Mary Shaw. *An Introduction to Software Architecture.* Carnegie Mellon University: s.n., 1994. 9810215940.

40. Santiago, Mellano, Cecilia, Molina y Pranelli, Germán. *Sistema para la Administración de Programas y Proyectos de Investigación coordinados por la Secretaría de Ciencia y Técnica.* PhD thesis, Universidad Nacional de Río Cuarto : s.n., 2010.

41. Ecured Team´s. Diagrama de despliegue. [En línea]. Conocimiento con todos y para todos Ecured .2010. [Citado el: 10 de abril de 2013.] Disponible en: http://www.ecured.cu/index.php/Diagrama_de_despliegue.

42. William, C. Hetzel. *The Complete Guide to Software Testing.* second. s.l. : John Wiley and Sons Canada, 1992. pág. 296 pp. 0471565679.

43. ISSI_CAMY, Lázaro. *JavaScript.* Madrid : Ediciones Anaya Multimedia, 2002. 84-415-1384-8.

44. Web Scripts team´s. Web Scripts . [En línea] Web Scripts [citado en: febrero de 2011]. Disponible en: http://wiki.alfresco.com/wiki/Web_Scripts.

45. EcuRed team´s. Visual Paradigm. [En línea] Conocimiento con todos y para todos Ecured. 2011-2012.[citado en: abril de 2013] Disponible en: http://www.ecured.cu/index.php/Visual_Paradigm.

46. the jquery fundation. Jquery. [En línea] The jQuery Foundation, JavaScript library. 2012. Disponible en:http://jquery.com/.

47. Microsoft team´s. Diagrama de componentes. [En línea] .msdn. 2013. [Citado el: 10 de abril de 2013.] Disponible en: http://msdn.microsoft.com/es-es/library/dd409390.aspx.

48. EcuRed team´s. Control de acceso. [En línea] Conocimiento con todos y para todos EcuRed 2010. [Citado el: 17 de enero de 2013.] Disponible en: http://www.ecured.cu/index.php/Control_de_acceso.

49. Society, IEEE Computer. *SWEBOK, Software Engineering Body of Knowledge.* s.l. : Group Managing Editor, 2004. 0-7695-2330-7.

50. JURISTO NATALIA, MORENO ANA M. MORENO, VEGAS SIRA .Técnicas de evaluación de software. 2005. [26 de mayo de 2013] Versión 12.0. Disponible en: http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_2/Comun/Tecnicas_de_evaluacion_de_software_Jurisco-Moreno.pdf

Bibliografía

1. InformaticaHoy Team´s. Que son los sistemas de gestión documental. [En línea]. INFORMATICAHOY. 2007. [Consultado el: 16 de noviembre de 2012.] Disponible en: http://www.informatica-hoy.com.ar/aprender-informatica/Que-son-los-sistemas-de-gestion-empresarial.php.

2. Anon. Concepto de usuario [En línea].Definición abc tu diccionario hecho fácil. 2007-2013. [Citado el: 18 de marzo de 2013.] Disponible en:http://www.definicionabc.com/tecnologia/usuario.php

3. Mastermagazine team´s. Definición de password. [En línea]. Mastermagazine. 2007. [Consultado el: 18 de marzo de 2013.] Disponible en: http://www.mastermagazine.info/termino/6239.php.

4. WordReference team´s. Definición de rol. [En línea] WordReference.com| Online language dictionaries 2011-2012. [Citado el: 14 de enero de 2013.] Disponible en: http://www.wordreference.com/definicion/rol.

5. Sobre Conceptos team´s. Concepto de permiso. [En línea] Sobre Conceptos porque todo termino tiene su origen .2007-2012. [Citado el: 14 de enero de 2013.] http://sobreconceptos.com/permiso.

6. Red Hat, Inc. Concepto de grupos de usuario. [En línea]. Red Hat Enterprise Linux 4, Introducción a la administración de sistemas. 2005. [Consultado el: 15 de enero de 2013.] Disponible en:http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/ch-acctsgrps.html.

7. Microsoft., Diccionario de informática e Internet de. *Diccionario de informática e Internet de Microsoft.* España: McGranw-Hill Interamericana, 2000. 0-7356-0615-3.

8. PricewaterhouseCoopers y PwC team´s. Administración de identidades y accesos. [En línea] PWC 2011-2012. [Citado el: 16 de septiembre de 2012.] Disponible en: http://www.pwc.com/co/es/administracion-de-identidades-y-accesos/index.jhtml.

9. MONTOYA, JOSÉ A. y RESTREPO, R. Gestión de identidades y control de acceso desde una perspectiva organizacional. Ing. USBMed*, 2012,* Vol. 3: p. 23-34.

10. RED HAT, INC. Nombre de usuario. [En línea]. Red Hat Enterprise Linux 4, Introducción a la administración de sistemas. 2005. [Consultado el: 20 de enero de

2013.] Disponible en: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-isa-es-4/ch-acctsgrps.html.

11. Desarrollado por el Centro Nacional de Cálculo. Políticas para cuentas de usuario del sistema. [En línea]. Sistema Institucional de Seguimiento y Atualización para Egresados SISAE. [Consultado el: 23 de enero de 2013.] Disponible en: http://sistemas.cenac.ipn.mx/SISAE/politicas/CuentasUsuario.do;jsessionid=9055CB4D1495AB980795632E620113DC.

12. *Oracle® Fusion Middleware User's Guide for Oracle Identity Manager.* Oracle user guide Developer team´s. 2013.

13. NUXEO team´s. Administración de usuarios y grupos. [En línea] .NUXEO/ Content Management Platform for Business Aplications. 2013. [Citado el: 25 de enero de 2013.] Disponible en:http://www.nuxeo.com/en.

14. Knowledgetree team´s. Administración de usuarios y grupos. [En línea]. Knowledgetree. 2013. [Citado el: 5 de febrero de 2013.] Disponible en: www.Knowledgetree.com.

15. OpenKM team´s. Administración de usuarios y grupos. [En línea] .OpenKM.com 2011. [Citado el: 11 de febrero de 2013.] Disponible en:http://www.openkm.com/es/.

16. Alfresco Software Inc. Administración de usuarios y grupos. [En línea] .Alfresco 2013. [Citado el: 6 de febrero de 2013.] disponible en: http://wiki.alfresco.com/wiki/Alfresco_Explorer.

17. Shari, Munwar. *Alfresco Enterprise Content Management Implementation.* 2006.

18. JACOBSON, IVAR. *El Proceso Unificado de Desarrollo de Software.* La Habana: Ediciones Félix Varela, 2004.

19. LARMAN, CRAIG. *UML Y PATRONES: Introducción al análisis y diseño orientado a objetos.* Hispanoamericana, S.A: Prentice Hall, 1999. ISBN: 9701702611.

20. PRESSMAN, ROGER S. Ingeniería de Software, un enfoque práctico. Quinta edicion. Madrid, McGraw-Hill Companies, 2002.

21. FIELDING ROY, THOMAS. Architectural Styles and the Design of Network-based Software Architectures. Tesis doctoral, University of California, California, 2000.

22. RESTful Web services team´s.. [En línea] RESTful Web services: The basics [Citado en: noviembre de 2008]. Disponible en: http://www.ibm.com/developerworks/webservices/library/ws-restful/.

23. Alfresco team´s. Web_Scripts. [En línea] .Alfresco. 2013. [Citado el: 20 de diciembre de 2012.] Disponible en: http://wiki.alfresco.com/wiki/.

24. Freemarker team´s. Definición Freemarker. [En línea] FreeMarker: java template engine library – overview. 2013. [Citado Abril 2012]. Disponible en: http://freemarker.org/.

25. Json team´s. Definición de Json. [En línea] Introducing Json .2013. [Citado el: 18 de enero de 2013.] Disponible en: http://www.json.org/json-es.html.

26. BRADENBAUGH, JERRY. *Aplicaciones JavaScript.* Madrid: Anaya, 2000. ISBN:84-41 5-1 070-9.

27. The jQuery Fundation. JavaScript . [En línea] .The jQuery Fundation, JavaScript library.2012. [Citado en: Mayo, 2013] Disponible en :http://jquery.com/.

28. Facultad de Informática de la Universidad de Las Palmas de Gran Canaria, ¿Que es html?. Conceptos básicos, 2012. 16 de mayo de 2013. Disponible en: http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/concepto.htm#html

29. W3C. Hojas de Estilo. [En linea]. W3C.es. 2013. [Citado en: mayo de 2013] Disponible en: http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo.

30. SAETHER, S. Manual de php. [En línea] [Citado en: noviembre de 2002]. Disponible en: http://www.calitae.com/manuales/manual-php.pdf.

31. CodeIgniter team´s. Marco de trabajo Codeigniter. [En línea] .CodeIgniter.2012. [citado en: abril de 2013] Disponible en: http://codeigniter.com/.

32. Wobreference team´s. JavaScript (ServerSide). [En línea] Webreference.2013. [Citado el: 19 de marzo de 2013.] Disponible en: http://www.webreference.com/programming/javascript/rg37/index.html.

33. ALVAREZ, MIGUEL ANGEL. Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. desarrolloweb.com, año.2013 Disponible en: http://www.desarrolloweb.com/articulos/1178.php.

34. Ecured team´s. Eclipse,_entorno_de_desarrollo_integrado. [En línea] .Conocimiento con todos y para todos EcuRed .2010. Disponible en:http://www.ecured.cu/index.php/

35. IEEE. IEEE:Guide for Developing Requirements Specification. [En línea] 1998. ISBN: 1-55937-716-X.

36. *Ingeniería de Requisitos en Aplicaciones para la Web- Un estudio comparativo.* Escalona, Maria Jose y Koch, Nora. Universidad de Sevilla: s.n., 2002.

37. Grupo de Investigación eumed.net (SEJ-309).Requisitos funcionales. [En línea] Eumed.net [Citado el: 6 de 3 de 2013.] Disponible en: http://www.eumed.net/libros-gratis/2010b/698/Requisitos%20funcionales.htm.

38.JACOBSON IVAR, BOOCH GRADY, RUMBAUGH JAMES. *El Proceso Unificado de Desarrollo de Software.* Varela, Felix.*La habana.* 2004. págs. 125-163, 187-202.

39. IEEE, Computer Society.*recommended practice for architectural description of software intensive systems.* s.l. : Ieee std 1471-2000.

40. Desarrolloweb.com Teams. [En línea] Desarrolloweb.com Guiarte Multimedia S.L., 1999. [Citado el: 7 de 3 de 2013.] Disponible en: http://www.desarrolloweb.com/articulos/1622.php.

41. DR. ROY THOMAS FIELDING. *Architectural Styles and the Design of Network-based Software Architectures.* sexta edición. s.l. : Addison Wesley, 2000. ISBN 0-599-87118-0.

42. David Garlan and Mary Shaw. *An Introduction to Software Architecture.* Carnegie Mellon University : s.n., 1994. 9810215940.

43. Santiago, Mellano, Cecilia, Molina y Pranelli, Germán. *Sistema para la Administración de Programas y Proyectos de Investigación coordinados por la Secretaría de Ciencia y Técnica.* PhD thesis, Universidad Nacional de Río Cuarto : s.n., 2010.

44. Ecured Team´s. Diagrama de despliegue. [En línea]. Conocimiento con todos y para todos Ecured .2010. [Citado el: 10 de abril de 2013.] Disponible en: http://www.ecured.cu/index.php/Diagrama_de_despliegue.

45. William, C. Hetzel. *The Complete Guide to Software Testing.* second. s.l. : John Wiley and Sons Canada, 1992. pág. 296 pp. 0471565679.

46. ISSI_CAMY, Lázaro. *JavaScript.* Madrid : Ediciones Anaya Multimedia, 2002. 84-415-1384-8.

47. Web Scripts team´s. Web Scripts . [En línea] Web Scripts [citado en: febrero de 2011]. Disponible en: http://wiki.alfresco.com/wiki/Web_Scripts.

48. EcuRed team´s. Visual Paradigm. [En línea] Conocimiento con todos y para todos Ecured. 2011-2012.[citado en: abril de 2013] Disponible en: http://www.ecured.cu/index.php/Visual_Paradigm.

49. the jquery fundation. Jquery. [En línea] The jQuery Foundation, JavaScript library. 2012. Disponible en:http://jquery.com/.

50. Microsoft team´s. Diagrama de componentes. [En línea] .msdn. 2013. [Citado el: 10 de abril de 2013.] Disponible en: http://msdn.microsoft.com/es-es/library/dd409390.aspx.

51. EcuRed team´s. Control de acceso. [En línea] Conocimiento con todos y para todos EcuRed 2010. [Citado el: 17 de enero de 2013.] Disponible en: http://www.ecured.cu/index.php/Control_de_acceso.

52. Society, IEEE Computer. *SWEBOK, Software Engineering Body of Knowledge.* s.l. : Group Managing Editor, 2004. 0-7695-2330-7.

53. JURISTO NATALIA, MORENO ANA M. MORENO, VEGAS SIRA .Técnicas de evaluación de software. 2005. [26 de mayo de 2013] Versión 12.0. Disponible en: http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_2/Comun/Tecnicas_de_evaluacion_de_software_Jurisco-Moreno.pdf54. Dirección de Teleformación. El Proceso Unificado de desarrollo. [en línea], Entorno Virtual de Aprendizaje. 2013. [25 de mayo de 2013]. Disponible en:http://eva.uci.cu/mod/resource/view.php?id=8500&subdir=/El_Proceso_Unificado_de_Desarrollo

Glosario de Términos

***Framework*:** un *framework*, es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros *software* para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Keylogger: es un tipo de *software* específico que se encarga de registrar las pulsaciones que se realizan en el teclado, para posteriormente memorizarlas en un fichero o enviarlas a través de internet. Suele usarse como *software* de tipo demonio, permitiendo que otros usuarios tengan acceso a contraseñas importantes, como los números de una tarjeta de crédito, u otro tipo de información privada que se quiera obtener.

Mouselogger: es un *software* que registra los movimientos y pulsaciones de botón de tu ratón, lo que te permite automatizar ciertas tareas repetitivas. Los registros pueden ser enviados por correo a otros usuarios en aras de propagar dicha información.

Multiusuario: se le llama multiusuario a la característica de un sistema operativo o programa que permite proveer servicio y procesamiento a múltiples usuarios simultáneamente, estrictamente es pseudo-simultáneo (tanto en paralelismo real como simulado).

Nodo: un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Un Nodo es un elemento de hardware o *software*.

Segregación de funciones: consiste en la separación de funciones ya sea en un sistema o institución.

Anexos

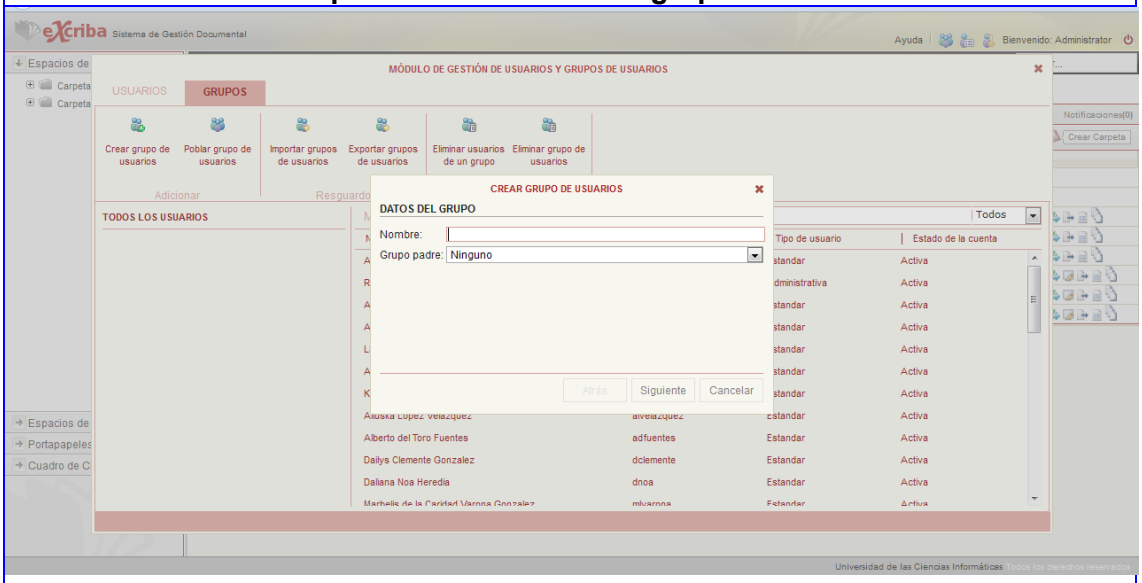
Anexo A

Descripción del caso de uso gestionar grupo de usuarios

Objetivo	Gestionar grupos de usuarios	
Actores	Administrador	
Resumen	El caso de uso comienza cuando el administrador desea gestionar un grupo de usuario. Para ello puede crear o eliminar los mismos. El caso de uso termina cuando realiza alguna de estas acciones	
Complejidad	Alta	
Prioridad	Media	
Precondiciones	El administrador debe estar autenticado en el sistema, para la sección eliminar debe existir al menos un grupo en el sistema.	
Postcondiciones	Se almacenan los datos del grupo creado y en cuanto al grupo eliminado desaparece por completo del sistema.	
Flujo de eventos		
Flujo básico: "Crear grupo de usuario"		
	Actor	**Sistema**
1.	El administrador accede a la opción "Crear grupo"	2. El sistema muestra una nueva interfaz con los campos requeridos para crear un grupo.
3.	El administrador completa los datos y accede al botón "Crear"	4. El sistema crea el grupo y notifica que el grupo ha sido creado satisfactoriamente
Flujos alternos		
Evento "Datos incompletos"		
	Actor	**Sistema**
3.1		El sistema muestra un mensaje indicando que debe completar todos los campos para crear el grupo.
	Evento "Datos incorrectos"	
3.2		El sistema muestra un mensaje indicando que ha

		introducido caracteres incorrectos.
Sección 1: "Eliminar grupo de usuario"		
Flujo básico "Eliminar grupo de usuario"		
	Actor	**Sistema**
4	El administrador selecciona los grupos que desea eliminar y accede a la opción "Eliminar grupos"	5. El sistema muestra un mensaje preguntando si desea eliminar los elementos seleccionados de manera permanente
6	El administrador accede al botón "Aceptar".	7. El sistema ejecuta la acción y notifica al administrador que el grupo ha sido eliminado satisfactoriamente.
	Flujos alternos	
Evento "Denegar eliminar grupo		**"**
1.	El administrador accede a la opción "Cancelar".	2. El sistema no elimina el grupo y cierra la interfaz.
Relaciones	**CU Incluidos**	
	CU Extendidos	
Requisitos no funcionales	**RNF1, RNF2**	
Asuntos pendientes		

Prototipo de interfaz : “crear grupo de usuario”



Prototipo de interfaz : “eliminar grupo de usuario”

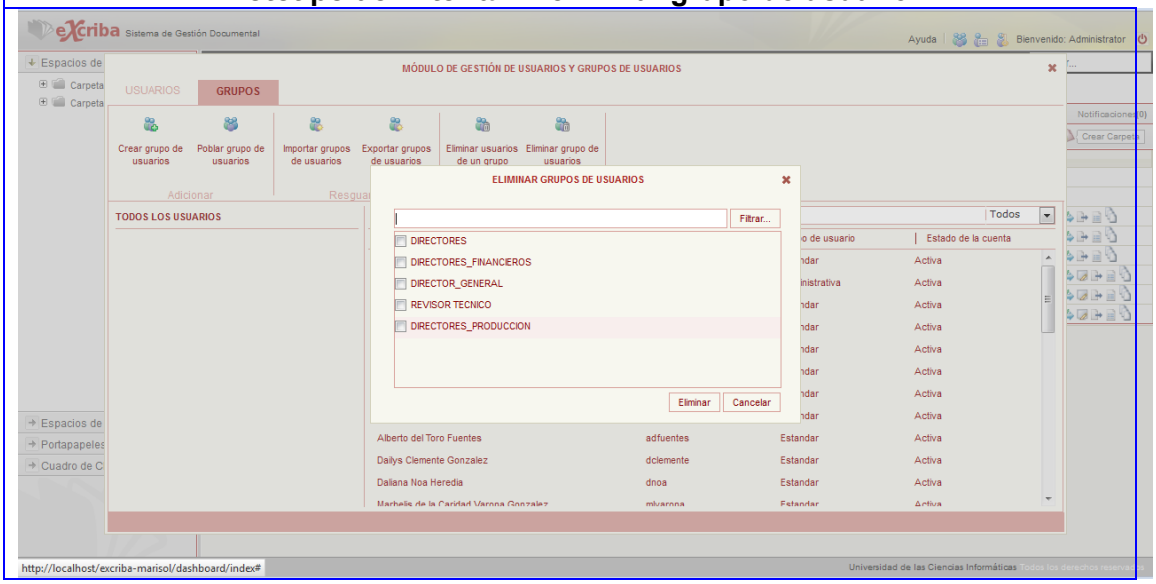


Tabla 18 Especificación del caso de uso gestionar grupo de usuarios.

Anexo B
Diagramas de clases del diseño

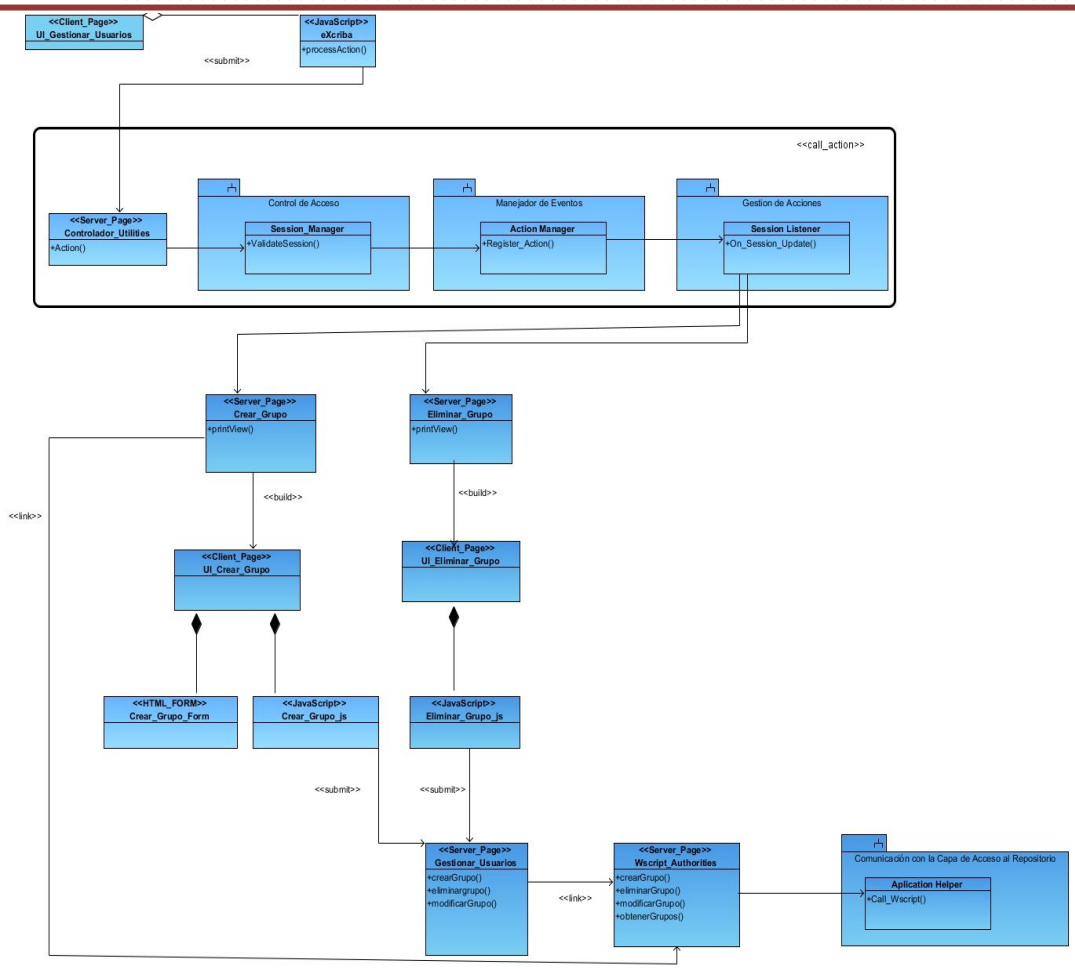


Figura 14 Diagrama de clases del caso de uso Gestionar grupos de usuario

Anexo C
Descripciones de las clases del diseño

Nombre: UI_Gestionar_Usuarios
Tipo de clase: Interfaz
Descripción: Es la interfaz que se le muestra inicialmente al usuario mediante la cual el mismo puede acceder a la funcionalidad que necesite.

Tabla 19 Descripción de la clase UI_Gestionar_Usuarios

Nombre: Crear _Grupo
Tipo de clase: página servidora
Descripción: Es la clase que construye la interfaz correspondiente a la funcionalidad que

el usuario haya seleccionado.	
Responsabilidades:	
Nombre:	Descripción:
printView()	Construye y genera la vista donde el usuario podrá visualizar los campos pertinentes para la creación de un grupo.

Tabla 20 Descripción de la clase Crear_Grupo

Nombre: Eliminar _Grupo	
Tipo de clase: página servidora	
Descripción: Es la clase que construye la interfaz correspondiente a la funcionalidad que el usuario requiera en este caso, "Eliminar grupo".	
Responsabilidades:	
Nombre:	Descripción:
printView()	Construye y genera una interfaz de confirmación para eliminar un grupo.

Tabla 21 Descripción de la clase Eliminar_Grupo

Nombre: Gestionar _Usuarios	
Tipo de clase: controladora	
Descripción: Contiene los métodos necesarios para interactuar con la capa de Acceso al Repositorio. Su función es actuar como la interfaz a través de la cual se comunica la capa de aplicación/negocio (específicamente las funcionalidades del módulo Administración de usuarios y grupos) con la de Acceso al Repositorio.	
Responsabilidades:	
Nombre:	Descripción:
crearGrupo	Coordina las operaciones necesarias que deben llevarse a cabo para crear un grupo,

	además recibe los datos del cliente los procesa y envía a la capa de acceso a datos y posteriormente o envía al cliente.
eliminarGrupo	Coordina las operaciones necesarias para la eliminación de un grupo.

Tabla 22Descripción de la clase Gestionar_Usuarios

Nombre: Wscript_Authorities	
Tipo de clase: controladora	
Descripción: Recibe los datos necesarios para hacer las llamadas a los servicios.	
Responsabilidades:	
Nombre:	Descripción:
crearGrupo	Tiene como función la creación de un grupo.
eliminarGrupo	Tiene como función la eliminación un grupo.

Tabla 23 Descripción de la clase Wscript_Authorities

Anexo D
Descripción de los servicios

Servicio: crear grupo	
URL:	/cu/uci/excriba/authorities/group/
Método HTTP:	POST
Lenguaje de implementación del controlador:	JavaScript
Parámetros o Argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
groupName: nombre para el grupo que se desea crear	Profesores
Parámetros o Argumentos de salida	
errores: almacena los mensajes de error que ocurren durante la ejecución del servicio	“Debe especificar un nombre del grupo” “Ocurrió un error mientras se creaba el grupo. Por favor inténtelo más tarde”

resultado: recoge el resultado del servicio	“OK”
advertencias: almacena mensajes que indican que una acción no pudo realizarse pero que no afecta el correcto funcionamiento del servicio	“Ya existe un grupo con este nombre”

Tabla 24 Descripción del servicio “Crear grupo de usuario”

Servicio: Eliminar cuenta de usuario	
URL:	/cu/uci/excriba/authorities/group/</url>
Método HTTP:	DELETE
Lenguaje de implementación del controlador:	JavaScript
Parámetros o Argumentos de entrada	
Nombre y Descripción:	**Formato/Ejemplo**
Parámetros o Argumentos de salida	
errores: almacena los mensajes de error que ocurren durante la ejecución del servicio	“Debe especificar el grupo que desea eliminar”
resultado: recoge el resultado del servicio	“OK”

Tabla 25 Descripción del servicio “Eliminar grupo de usuario”

Anexo E

Diagramas de colaboración

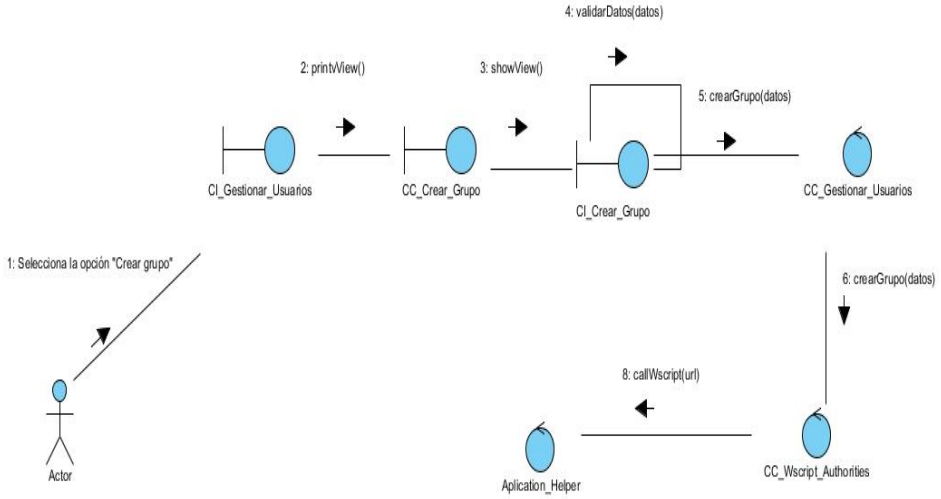


Figura 15 Diagrama de colaboración del caso de uso Gestionar Grupo de usuarios, sección Crear grupo

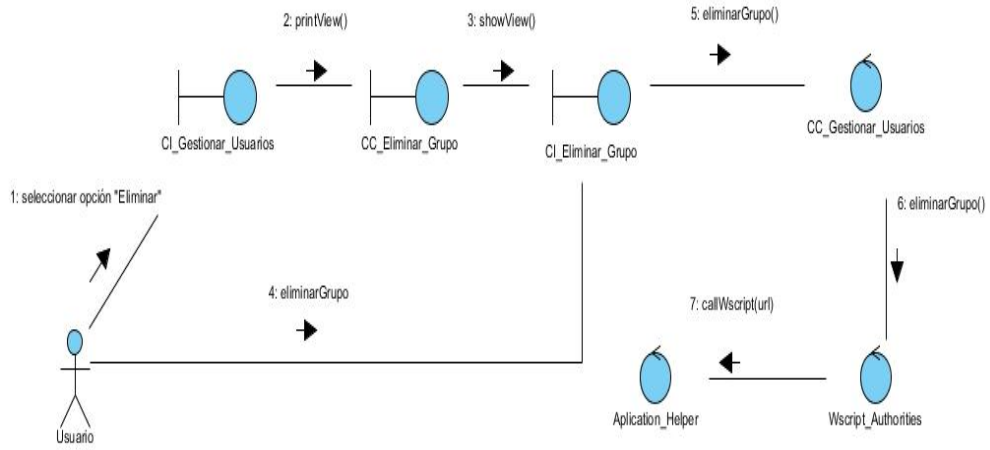


Figura 16 Diagrama de colaboración del caso de uso Gestionar Grupo de usuarios, sección Eliminar grupo

Anexo F
Diagrama de componentes

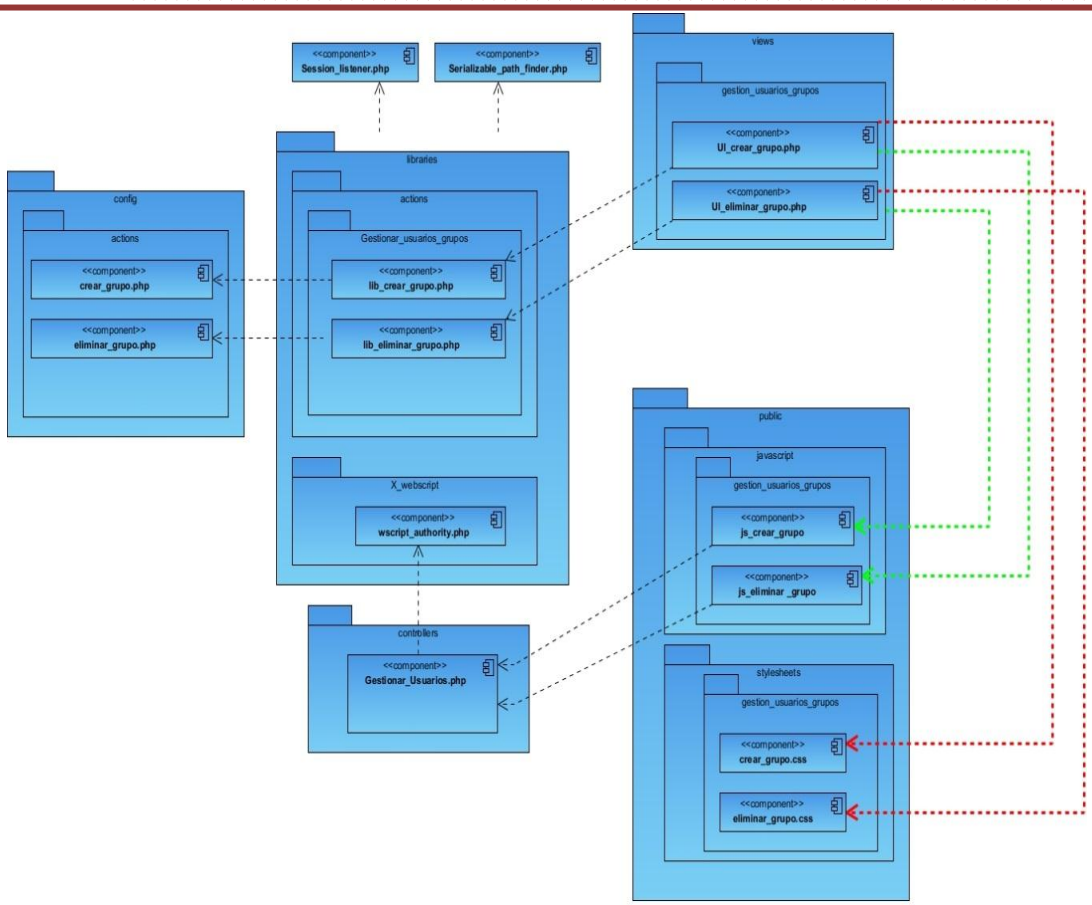


Figura 17 Diagrama de componentes del caso de uso Gestionar grupos de usuario