



**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS  
CENTRO DE INFORMATIZACIÓN UNIVERSITARIA  
FACULTAD 1**

**Módulo de control de acceso para el Sistema de Gestión de Documentos Históricos Dexcriba.**

Trabajo de diploma para optar por el título de Ingeniero en  
Ciencias Informáticas.

**Autor:** Leidy González Reyes

**Tutores:** Ing. Leodán De los Ángeles Buduén  
Ing. Marlon Jorge Remedios González

**La Habana**

**Mayo 2013**

## Declaración de autoría

Por este medio 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 el presente a los \_\_\_\_ días del mes de \_\_\_\_ del año \_\_\_\_.

---

Firma del autor  
Leidy González Reyes

---

Firma del tutor  
**Ing. Leodán de los Ángeles Buduén**  
González

---

Firma del tutor  
**Ing. Marlon J. Remedios**

**Ing. Leodán De los Ángeles Buduén:** Graduado de Ingeniero en Ciencias Informáticas en el año 2009, con siete años de experiencia en temas de desarrollo de software de gestión de documentos históricos y participación en diferentes eventos relacionados con la materia. En la actualidad labora en el Departamento de Gestión Documental de CENIA de la Universidad de las Ciencias Informáticas (UCI).

**Ing. Marlon Jorge Remedios González:** Graduado como Ingeniero en Ciencias Informáticas en el año 2008 en la Universidad de las Ciencias Informáticas (UCI). Actualmente alcanza la categoría docente de Profesor Asistente. Forma parte de la matrícula de la maestría en informática aplicada de la UCI 3era edición. En el 2011 acreditó una publicación y participación en evento en Informática 2011. Cuenta con participación además en REV 2012 que será publicado en IAOE.

## *Dedicatoria*

*Dedico este trabajo a mi abuela porque sé que ella más que nadie se sentiría súper orgullosa de verme convertida en una profesional, porque a pesar de no estar físicamente, pensar en ella me da las fuerzas para conseguir todo lo que me propongo.*

## *Agradecimientos*

*A mi papá que siempre ha dado todo de sí para verme feliz y hacer todos mis sueños realidad.*

*A mi mamá por comprenderme, por confiar en mí, por darme tanto amor, por ser la mejor madre del mundo. Te quiero mucho.*

*A mi novio por darme su ayuda y apoyo incondicional en todo momento, por darme fuerzas para seguir adelante sin importar los obstáculos, por su comprensión, su cariño y su amor, por estar junto mí cuando más lo necesito.*

*Gracias.*

*A mi hermano por estar siempre a mi lado y brindarme su cariño.*

*A mi abuela, mis tíos y mis primos por apoyarme durante estos 5 años.*

*A mi bebe Balú por ser la alegría de mi casa y cuidar de mis papas cuando no estoy.*

*A mi familia de la UCI Made, Mayi, Ramón, Ramniel, Diana, José Miguel, Daymaris, Yesica, Greysi e Islen con los que he compartido todo lo bueno y lo malo que he vivido en la universidad.*

*A mis amigos del inseparable grupo 2 por todos los momentos lindos que hemos vivido juntos.*

*A mis tutores por su tiempo y dedicación, por su exigencia.*

*A los profes Yoani, Reinier, Alien, Yanet y Yaili por sus oportunas sugerencias.*

*A todas las personas que me aprecian y se preocupan por mí, gracias.*

## **Resumen**

La investigación está encaminada a obtener una implementación de requerimientos de seguridad con el objetivo de crear un módulo correspondiente al Sistema de Gestión de Documentos Históricos DeXcriba, que automatice el proceso de control de acceso a la documentación existente. Esto permite la gestión de usuarios, asignación de roles y privilegios de acceso, estableciendo premisos a nivel de funcionalidad y nivel de organización de la información, incluyendo además un proceso de gestión de trazas y auditoría, para llevar a cabo el control de las operaciones realizadas en el sistema, posibilitando que toda la información que se administre en el sistema sea accedida por el personal autorizado.

Para su cumplimiento se realizó un estudio de homólogos con el fin de investigar cómo es desarrollado el control de acceso a la información en otros sistemas. Además se analizaron varios modelos de control de acceso.

Su desarrollo está basado en tecnologías libres, utilizando PHP 5.3 como lenguaje de programación, PostgreSQL 9.1 como sistema gestor de bases de datos y como metodología de desarrollo de software RUP, generando los artefactos correspondientes para documentar el proceso de desarrollo.

***Palabras clave: control de acceso, rol, usuario, auditoría, nivel de organización.***

# Índice.

<b>INTRODUCCIÓN</b> .....	<b>3</b>
<b>1. CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LA INVESTIGACIÓN</b> .....	<b>9</b>
1.1. INTRODUCCIÓN .....	9
1.2. CONCEPTOS FUNDAMENTALES .....	9
1.2.1. Archivo .....	9
1.2.2. Documento de archivo .....	10
1.2.3. Niveles de organización de los documentos .....	10
1.2.4. Tipos de niveles de organización .....	10
1.2.5. Archivística .....	11
1.2.6. Descripción archivística .....	11
1.2.7. Norma Internacional General de Descripción Archivística ISAD (G) .....	12
1.2.8. Control de acceso .....	13
1.2.9. Identificación .....	14
1.2.10. Autenticación .....	14
1.2.11. Autorización .....	14
1.2.12. Auditoría .....	14
1.3. SEGURIDAD EN APLICACIONES WEB .....	15
1.4. SISTEMAS DE GESTIÓN DE ARCHIVOS HISTÓRICOS .....	17
1.4.1. Portal Pares .....	17
1.4.2. ICA-AtoM .....	17
1.4.3. ArchiVenHIS .....	18
1.4.4. Análisis de las soluciones encontradas .....	19
1.5. MODELOS DE CONTROL DE ACCESO .....	19
1.5.1. Modelo de control de acceso discrecional .....	19
1.5.2. Modelo de control de acceso obligatorio .....	19
1.5.3. Modelo de control de acceso basado en tareas .....	20
1.5.4. Modelo de control de acceso basado en roles .....	20
1.6. MODELO DE AUTENTICACIÓN, AUTORIZACIÓN Y AUDITORÍA (AAA) .....	21
1.6.1. Estructura básica de un sistema AAA .....	23
1.7. TECNOLOGÍAS A UTILIZAR EN LA INVESTIGACIÓN .....	23
1.7.1. Lenguajes .....	23
1.7.2. Marco de trabajo de desarrollo .....	26
1.7.3. Sistema gestor de base de datos .....	28
1.7.4. Servidor web .....	29
1.7.5. Entorno de desarrollo integrado .....	30
1.7.6. Herramientas de modelado .....	30
1.7.7. Metodología de desarrollo de software .....	31
CONCLUSIONES .....	32
<b>2. CAPÍTULO II: CARACTERÍSTICAS DEL MÓDULO</b> .....	<b>33</b>
2.1. INTRODUCCIÓN .....	33
2.2. MODELO DE DOMINIO .....	33
2.2.1. Diagrama de clases del modelo del dominio .....	34
2.3. REQUISITOS FUNCIONALES .....	35

2.4. REQUISITOS NO FUNCIONALES.....	36
2.5. ACTORES DEL SISTEMA.....	37
2.6. DEFINICIÓN DE CASOS DE USO (CU) DEL SISTEMA .....	37
2.6.1. Patrón de Caso de Uso utilizado.....	38
2.6.2. Diagrama de Caso de Uso del Sistema .....	39
2.6.3. Descripción de los Casos de Uso del sistema .....	39
2.7. DISEÑO DEL SISTEMA. ....	43
2.7.1. Diagramas de clases del diseño.....	43
2.8. MODELO DE DATOS. ....	45
2.8.1. Descripción de las tablas del modelo de datos.....	46
2.9. PATRONES. ....	49
CONCLUSIONES.....	50
<b>3. CAPÍTULO III: IMPLEMENTACIÓN Y PRUEBA DEL MÓDULO .....</b>	<b>51</b>
3.1. INTRODUCCIÓN.....	51
3.2. DIAGRAMA DE COMPONENTES. ....	51
3.2.1. Diagrama de componentes del paquete Modelo. ....	51
3.2.2. Diagrama de componentes del paquete Control.....	52
3.2.3. Diagrama de componentes del paquete Vistas.....	53
3.3. DIAGRAMA DE DESPLIEGUE.....	53
3.4. PRUEBAS. ....	54
3.4.1. Pruebas de caja blanca.....	54
3.4.2. Pruebas de caja negra.....	59
CONCLUSIONES.....	65
<b>4. CONCLUSIONES GENERALES.....</b>	<b>66</b>
<b>5. BIBLIOGRAFÍA.....</b>	<b>67</b>
<b>ANEXOS .....</b>	<b>71</b>

## Índice de figuras

<i>Figura 1: Modelo de dominio</i> .....	34
<i>Figura 2: Diagrama de Caso de uso del sistema.</i> .....	39
<i>Figura 4 Modelo de datos.</i> .....	46
<i>Figura 6: Diagrama de componentes</i> .....	51
<i>Figura 7: Diagrama de componentes-Paquete Modelo</i> .....	52
<i>Figura 8: Diagrama de componentes-Paquete Control.</i> .....	52
<i>Figura 9: Diagrama de componentes-Paquete Vistas</i> .....	53
<i>Figura 10: Modelo de despliegue</i> .....	54
<i>Figura 11: Grafo de flujo asociado</i> .....	56



## Índice de tablas

<i>Tabla 1: Actores del sistema</i>	37
<i>Tabla 2: Descripción del caso de uso: Asignar funcionalidad a rol</i>	40
<i>Tabla 3: Descripción del caso de uso: Gestionar rol</i>	43
<i>Tabla 4: Diagramas de clases del diseño</i>	45
<i>Tabla 5: Modelo de datos entidad: usuario</i>	47
<i>Tabla 6: Modelo de datos entidad: traza</i>	48
<i>Tabla 7: Modelo de datos entidad: funcionalidad</i>	48
<i>Tabla 8: Modelo de datos entidad: rol</i>	48
<i>Tabla 9: Modelo de datos entidad: url</i>	49
<i>Tabla 10: Modelo de datos entidad: nivel_organización</i>	49
<i>Tabla 11: Caminos básicos</i>	57
<i>Tabla 12: Casos de prueba de caja blanca</i>	59
<i>Tabla 13: Casos de pruebas caja negra</i>	63
<i>Tabla 14: Resultados de las pruebas</i>	64
<i>Tabla 15: Descripción detallada de CU: Autenticar usuario</i>	71
<i>Tabla 16: Descripción detallada de CU: Gestionar usuario</i>	71
<i>Tabla 17: Descripción detallada de CU: Buscar usuario</i>	71
<i>Tabla 18: Descripción detallada de CU: Buscar rol</i>	72
<i>Tabla 19: Descripción detallada de CU: Asignar rol a usuario</i>	72
<i>Tabla 20: Descripción detallada de CU: Gestionar traza</i>	72
<i>Tabla 21: Descripción detallada de CU: Búsqueda traza sencilla</i>	73
<i>Tabla 22: Descripción detallada de CU: Búsqueda traza avanzada</i>	73
<i>Tabla 23: Descripción detallada de CU: Gestionar funcionalidad</i>	73
<i>Tabla 24: Descripción detallada de CU: Buscar funcionalidad</i>	74
<i>Tabla 25: Descripción detallada de CU: Registrar usuario</i>	74
<i>Tabla 26: Asignar nivel de organización a rol</i>	74
<i>Tabla 27: Modelos de datos entidad: sexo</i>	75
<i>Tabla 28: Modelo de datos entidad: usuario_rol</i>	75
<i>Tabla 29: Modelo de datos entidad: funcionalidad_rol</i>	76
<i>Tabla 30: Modelo de datos entidad: rol_nivel_organización</i>	76

## Introducción.

La historia de la evolución humana ha podido construirse gracias a las salvaguardas de la información en disímiles soportes a través del tiempo. Desde siglos remotos el hombre sintió la necesidad de dejar constancia de sus momentos vividos creando así una guía para el desarrollo de futuras generaciones, para ello utilizaba todos los medios que la naturaleza le brindaba.

Los espacios físicos destinados al almacenamiento de los documentos a lo largo de la historia han tomado diferentes nombres en dependencia de las culturas y de las funciones para los cuales fueron creados. En la actualidad los lugares destinados a guardar estos documentos se conocen con el nombre de archivos.(1)

Desde el surgimiento de los archivos hasta la actualidad han sido clasificados de acuerdo a su nivel de importancia con el objetivo de restringir el acceso del público en general. Con el fin de evitar que ocurran pérdidas notables o una manipulación en beneficio propio de quien los usurpe, se hace necesario que exista una jerarquía en cuanto a los permisos para su acceso.

Los archivos juegan un papel esencial para la gestión documental, pues garantizan el tratamiento de los documentos a lo largo de toda su vida. La gestión documental es un área de la administración general que se encarga de garantizar la economía y la eficiencia en la creación, mantenimiento, uso y disposición de los documentos administrativos durante todo su ciclo de vida.(1)

Una adecuada gestión documental puede reportar a las organizaciones ventajas en términos de ahorro de costos y agilización de la tramitación de contratos y documentos. La gestión documental según la archivística integrada se divide en cinco etapas: la primera se conoce como Génesis, la segunda Archivo de Gestión, la tercera Archivo Central, la cuarta Archivo Intermedio y la quinta Archivo Histórico. Esta última tiene como misión atesorar, organizar, custodiar y conservar la documentación de valor permanente que constituye el patrimonio documental de una nación o territorio.(2)

Producto del incremento del volumen de documentación y el desarrollo de las tecnologías de la información y las comunicaciones (TIC), surgieron los Sistemas de Gestión de Archivos Históricos, una solución que contribuye a la conservación y difusión de la memoria histórica resguardada por instituciones de archivo.

Una aplicación de gestión de archivos históricos debe garantizar que los documentos tengan el nivel de seguridad adecuado, muchos de ellos contienen información sensible por lo cual no deberían ser accedidos más que por el personal autorizado. Por esto, se hace necesario establecer una jerarquía en cuanto a los permisos de acceso que pueden surgir de acuerdo al tipo y nivel de importancia de la misma. Para garantizar estas condiciones es importante tener implementado un adecuado mecanismo de autorización, que administre lo mejor posible el flujo de documentación, automatizando los procesos de identificación y autenticación de las personas. Incluyendo además un proceso de trazabilidad y auditoría, para llevar a cabo un control de las operaciones en que se involucran los usuarios con el objetivo de lograr la delimitación de niveles de acceso a la información para los usuarios del sistema. La documentación existente en los sistemas de gestión de archivos históricos es descrita bajo la norma ISAD (G), esta plantea el uso de niveles de organización para la información, por ello se hace necesario también establecer el control de acceso teniendo en cuenta estos niveles.

Cuba no se encuentra aislada de estos avances tecnológicos, es por esto que en el año 2002 se creó la Universidad de las Ciencias Informáticas (UCI) que entre sus objetivos incluye formar a un profesional altamente calificado, informatizar el país y desarrollar la industria del software para contribuir al desarrollo económico del mismo. Para cumplir con estos objetivos, la misma cuenta con diferentes centros de producción dentro de los cuales se encuentra el Centro de Informatización Universitaria (CENIA), y en este un departamento de gestión documental con varias líneas de desarrollo encargadas del desarrollo de software para facilitar el tratamiento que se le da a los archivos a lo largo de su vida.

Durante el año 2007 se desarrolla el sistema de gestión de documentos históricos ArchiVenHIS, con el fin de proporcionarle al Archivo General de la Nación (AGN) de la República Bolivariana

de Venezuela una solución, desarrollada bajo tecnologías libres y estándares abiertos, que contribuye a la preservación y difusión del fondo documental bajo su custodia.

ArchiVenHIS fue concebido como una aplicación orientada a resolver las necesidades específicas del Archivo General de la República Bolivariana de Venezuela, para cubrir la necesidad de administración de la información más urgente, y después ir avanzando sobre otras áreas. De esta manera ArchiVenHIS toma la estructura de los departamentos y roles existentes en cada una de las áreas, pero no puede ser utilizado en otro entorno de trabajo, quedando obsoleto fuera de las fronteras del AGN, esto lo hace un software a la medida, puesto que no puede ser adaptable o extensible para ser usado en cualquier otra institución. Además cuenta con un sistema de control de acceso desarrollado solo a nivel de funcionalidades, es decir, a pesar de que sus documentos están descritos siguiendo la Norma Internacional General de Descripción Archivística ISAD (G), el sistema no brinda acceso seguro según los niveles de organización de la información, por ello actualmente su uso es solo exclusivamente para personal administrativo.

Con el objetivo de solventar las limitantes encontradas en ArchiVenHIS, se hace necesario crear un software genérico que tribute a la conservación y difusión de la documentación existente en un archivo histórico cumpliendo con las siguientes características:

- ✓ Sus módulos puedan ser agregados según los requerimientos de cada organización y no esté limitado a una estructura estática.
- ✓ Establezca el control de acceso a la información, no solo a nivel de funcionalidad, sino también, teniendo en cuenta el nivel de organización de la documentación.

La capacidad de proveer semejante servicio constituye un importante valor agregado para los sistemas de gestión de archivos históricos.

Por lo anteriormente expuesto se plantea como **problema a resolver**: ¿Cómo delimitar el acceso a las funcionalidades y niveles de organización de la información del Sistema de Gestión de Documentos Históricos Dexcriba?

Definiéndose como **objeto de estudio** de esta investigación el control de acceso para sistemas informáticos.

De acuerdo con la problemática planteada se propone como **objetivo general**: Desarrollar un módulo para el Sistema de Gestión de Documentos Históricos Dexcriba, que permita la gestión de roles, usuarios y la asociación de funcionalidades a estos, delimitando además, el acceso a los niveles de organización de la información mediante tecnologías libres y estándares abiertos.

Definiéndose como **campo de acción** el control de acceso para sistemas de gestión de documentos de archivo histórico.

Con el propósito de cumplir con el objetivo general, se plantean las siguientes tareas de la investigación:

- ✓ Análisis bibliográfico relacionado con el control de acceso a sistemas para el desarrollo del módulo incluyendo: conceptos fundamentales, modelos de control de acceso, entre otros aspectos.
- ✓ Análisis y diseño del módulo a través de la especificación de los requisitos funcionales y no funcionales, para luego transformarlos en los productos de trabajo que especifican el diseño del software que se va a desarrollar.
- ✓ Implementación del módulo para desarrollar la solución del control de acceso al Sistema de Gestión de Documentos Históricos Dexcriba.
- ✓ Ejecución de las pruebas del desarrollador con el objetivo de validar el funcionamiento de las funcionalidades que componen el módulo.
- ✓ Integración del sistema para verificar si el módulo de control de acceso al Sistema de Gestión de Documentos Históricos Dexcriba, además de cumplir los requisitos que se plantearon se integra correctamente a la arquitectura del sistema general.

### **Los métodos teóricos utilizados son:**

**Modelación:** Se utilizó para representar el sistema a través del modelado de entidades, actores, componentes, entre otros elementos, con el fin de lograr una mejor comprensión del proceso de control de acceso, modelando el mismo de manera general, además se describió paso a paso cómo está estructurado el sistema según los requerimientos definidos, obteniendo así una panorámica del creciente flujo de actividades existentes en dicho proceso.

**Análisis histórico-lógico:** Permitió estudiar el proceso evolutivo por el cual han transitado las tecnologías relacionadas con los sistemas de controles de acceso realizando un análisis riguroso de sus antecedentes.

**Análisis de contenido:** Posibilitó el estudio de la bibliografía concretando aquella de interés para la investigación con el propósito de determinar el sistema de conceptos y fundamentos relacionados con el tema.

### **El método empírico utilizado es:**

**Observación:** Condujo el estudio del estado del arte permitiendo realizar un análisis sistémico, selectivo y objetivo de los principales procesos de control de acceso que en la actualidad se aplican a sistemas informáticos.

### **Justificación de la investigación**

La investigación provee al Sistema de Gestión de Documentos Históricos DEXCRIBA de un módulo que permite gestionar los roles y usuarios con el objetivo de limitar el acceso a documentación reservada o útil para un conjunto restringido de personas. Debido a esto, no se permitirá al usuario gestionar documentos a los cuáles no tiene definido el acceso, garantizando así que solo la persona autorizada con los permisos requeridos pueda ejecutar cualquier acción sobre la documentación.

**El presente trabajo de diploma se ha estructurado de la siguiente forma:**

**Capítulo I: Fundamentación teórica de la investigación:** En este capítulo se abordan aspectos generales del estudio del estado del arte, conceptos claves dentro del proceso de la archivística y el control de acceso a la información que fundamentan la investigación, así como la metodología, tecnologías y herramientas empleadas en la desarrollo del módulo.

**Capítulo II: Características del módulo:** Se describen de manera general los procesos involucrados en el campo de acción. Se presenta una descripción general de la propuesta de solución y su funcionamiento. Además se especifican los requisitos de software, diagramas de modelo de dominio y casos de uso del sistema. Se expone a través de un conjunto de artefactos la solución que se propone para el problema en cuestión.

**Capítulo III: Implementación y prueba del módulo:** Se representan los elementos físicos necesarios para el despliegue de la aplicación, empleando para ello un diagrama de despliegue. Se efectúa además la validación y prueba de la solución de acuerdo a los requisitos que debe cumplir para garantizar una calidad óptima, a través de las pruebas de caja negra y caja blanca.

# **CAPITULO I: Fundamentación teórica de la investigación.**

## ***Introducción***

En el presente capítulo se exponen los fundamentos teóricos que sustentan la base de la investigación para el desarrollo del módulo. Además, se realiza una breve descripción de las distintas herramientas, lenguajes y tecnologías que serán utilizadas para el desarrollo del mismo especificándose también la metodología de desarrollo de software a utilizar.

## ***Conceptos fundamentales***

### **Archivo**

Según la visión de Mayra Mena Mugica *“el archivo no es más que el reflejo natural y la plasmación en sus documentos de las actividades y tareas de una entidad determinada”*. (1)

Antonia Heredia define un archivo como *“uno o más conjuntos de documentos, sea cual sea su fecha, su forma y soporte material, acumulados en un proceso natural por una persona o institución pública o privada en el transcurso de su gestión, conservados, respetando aquel orden, para servir como testimonio e información para la persona o institución que los produce, para los ciudadanos o para servir de fuentes de historia”*. (2)

En el 2011 Cruz Mundet ofrece una definición un poco más actualizada, definiendo el archivo como *“un sistema corporativo de gestión que contribuye de manera efectiva, mediante una metodología propia, a la definición de los procesos de producción administrativa garantizando la correcta creación de los documentos, su tratamiento, conservación, acceso y comunicación”*. (3)

De manera general un archivo puede definirse como un conjunto de documentos, sin importar su naturaleza, soporte o fecha de creación, pueden ser acumulados por una persona o institución, y sirven como información o testimonio de sus creadores. Así mismo se entiende también por



archivo la institución cultural donde se reúnen, conservan, ordenan y difunden conjuntos de documentos útiles en la investigación, la cultura, la información y la gestión administrativa.

## **Documento de archivo**

Son definidos como *“el testimonio material de un hecho o acto realizado en el ejercicio de sus funciones por personas físicas o jurídicas, públicas o privadas, de acuerdo con las características de tipo material y formal”* según lo expresado por Mayra Mena Mugica. (1)

En 1984 la definición incorporada en la Ley de Archivos de Andalucía plantea que *“un documento es toda expresión en lenguaje oral o escrito, natural o codificado, recogida en cualquier tipo de soporte material, así como cualquier otra expresión gráfica que constituya testimonio de funciones y actividades sociales del hombre y de los grupos humanos, con exclusión de las obras de creación y de investigación editadas, y de las que por su índole, forman parte del patrimonio bibliográfico, así como las expresiones aisladas de naturaleza arqueológica o artística”*. (4)

De esta forma puede decirse que un documento de archivo es producido o recibido por una persona o institución en el cumplimiento de sus funciones y se genera en el curso de su gestión o actividad para el cumplimiento de sus objetivos, además de ser conservado con el objetivo de utilizarse como prueba o fuente de información.

## **Niveles de organización de los documentos**

Los niveles de organización son una estructura jerárquica previamente creada que hace referencia a las divisiones documentales a las que se aplica la descripción, las cuales están divididas en subdivisiones documentales según la preferencia de la institución.

## **Tipos de niveles de organización**

Clasificación que se asigna a los niveles de organización, para crear unidad de documentos, que tienen una forma particular, o como consecuencia de relaciones derivada de su producción, recepción o utilización.

## **Archivística**

La archivera española Antonia Heredia Herrera, la define como *“la ciencia de los archivos, no de los documentos, aunque en última instancia estos sean el producto integrante de aquellos. Como tal se ocupará de la creación, historia, organización y el servicio de los mismos a la Administración y a la Historia, en definitiva a la Sociedad”*. (2)

En el manual de archivística José Ramón Cruz Mundet plantea que la definición correcta de la archivística es la dada por el Consejo Internacional de Archivos (CIA) donde se define como *el estudio teórico y práctico de los principios, procedimientos y problemas concernientes a las funciones de los archivos*. (3)

Según el Diccionario de Terminología Archivística es *“la disciplina que trata de los aspectos teóricos y prácticos (tipología, organización, funcionamiento, planificación, etcétera) de los archivos y el tratamiento archivístico de sus fondos documentales”*. (5)

En conclusión, puede definirse la archivística como la ciencia que estudia la naturaleza de los archivos, se encarga de su organización, origen y de los medios para su utilización.

### **Descripción archivística**

*“La descripción archivística es el medio utilizado por el archivero para obtener información contenida en los documentos y ofrecerla a los interesados en ella”*. (5) Se realiza con dos objetivos fundamentales: dar información a los usuarios y facilitar el trabajo al archivero. La descripción de los documentos debe ser concreta, breve y contener las ideas básicas; tanto los factores internos como los externos deben ser descritos.

Las descripciones son realizadas por los archivistas a través de instrumentos descriptivos como guías, inventarios, índices y catálogos. Un instrumento descriptivo que ha ganado gran aceptación en el campo archivístico es la Norma Internacional General de Descripción Archivística ISAD (G).

## **Norma Internacional General de Descripción Archivística ISAD (G).**

La Norma Internacional General de Descripción Archivística ISAD (G) por sus siglas en inglés (International Standard for Archival Description-General), es una normativa internacional para la descripción de documentos. (6)

ISAD (G) se concibe como una guía general para la elaboración de descripciones archivísticas, donde la define como la creación de una representación precisa de los fondos y sus partes componentes mediante el proceso de captar, componer y organizar la información que sirve para identificar el material archivístico y explicar el contexto y sistemas archivísticos que lo produjeron. Contiene reglas generales para la descripción que se pueden aplicar a cualquier tipo documental y soporte físico de la documentación, excepto para materiales especiales. No se concibe como una norma exclusiva sino que ha de ser utilizada conjuntamente con las respectivas normas nacionales.

Las cinco unidades de descripción (documento o conjunto de documentos objeto de descripción) básicas según la norma ISAD (G) son:

- ✓ Fondo.
- ✓ Subfondo.
- ✓ Serie.
- ✓ Expediente.
- ✓ Unidad documental.

En cuanto a la profundidad en la descripción, establece 26 elementos descriptivos, de los cuales señala 6 como esenciales: código de referencia, título, productor, fechas, volumen de la unidad de descripción y nivel de descripción. El resto dependerá de la naturaleza de la unidad de

descripción. Los 26 elementos que componen la norma se estructuran en 7 áreas de información descriptiva: (6)

**Área de Identificación:** Contiene la información esencial para identificar la unidad de descripción.

**Área de Contexto:** Contiene la información relativa al origen y custodia de la unidad de descripción.

**Área de Contenido y Estructura:** Contiene la información relativa al objeto y organización de la unidad de descripción.

**Área de Acceso y Utilización:** Contiene la información relativa a la accesibilidad de la unidad de descripción.

**Área de Documentación Asociada:** Contiene la información relativa a aquellos documentos que tienen una relación significativa con la unidad de descripción.

**Área de Notas:** Contiene información especial y aquella otra que no ha podido incluirse en ninguna de las demás áreas.

**Área de control de la información:** Contiene la información relativa al cómo, cuándo y quién ha elaborado la descripción archivística.

### **Control de acceso**

El control de acceso implica quién tiene acceso a sistemas informáticos específicos y recursos en un momento dado. El concepto de control de acceso consta de tres pasos. Estos pasos son: identificación, autenticación y autorización. Con el uso de estos tres principios un administrador del sistema puede controlar qué recursos están disponibles para los usuarios de un sistema. (7)

## **Identificación**

La identificación se refiere a cosas como nombre de usuario y tarjeta de identificación. Es el medio por el cual un usuario del sistema se identifica. Este paso se realiza generalmente al iniciar sesión. (7)

## **Autenticación**

La autenticación es el segundo paso del proceso de control de acceso. Contraseñas, reconocimiento de voz, y escáneres biométricos son métodos comunes de autenticación. El objetivo de la autenticación es verificar la identidad del usuario del sistema. (7)

## **Autorización**

La autorización se produce después de que un usuario del sistema se autentica y luego es autorizado a utilizar el sistema. El usuario está generalmente sólo autorizado a usar una porción de los recursos del sistema en función de su papel en la organización. (7)

## **Auditoría**

La auditoría es la recopilación de datos sobre el uso de los recursos del sistema. Los datos de auditoría proporcionan un registro de los eventos del sistema relacionados con la seguridad. La auditoría correcta comienza con dos funciones de seguridad: identificación y autenticación. En cada inicio de sesión, después de que un usuario proporciona un nombre de usuario y la autenticación se realiza correctamente, se genera un ID<sup>1</sup> de usuario de auditoría único e inmutable y se lo asocia con el usuario, y se genera un ID de sesión de auditoría único y se lo asocia con el proceso del usuario. El ID de sesión de auditoría es heredado por cada proceso que se inicia durante esa sesión de inicio de sesión. Cuando un usuario cambia a otro usuario, a todas las acciones del usuario se les realiza un seguimiento con el mismo ID de usuario de auditoría. (8)

---

<sup>1</sup> Elemento que identifica al usuario unívocamente.

## ***Seguridad en aplicaciones web***

En el mundo de hoy, debido al alto grado de interconectividad existente entre estaciones de trabajo y otros dispositivos electrónicos; la Web se ha convertido en un importante medio de comunicación e intercambio de información, que ofrece un acceso abierto a un conjunto de datos que explícitamente se hacen públicos. Por tanto, es necesario limitar el acceso a informaciones reservadas o útiles para un conjunto restringido de personas. Otro aspecto que cobra especial importancia es la seguridad de la información que se intercambia en la Web, que cada vez está más expuesta, a personas que incurren en delitos informáticos.

La principal vulnerabilidad de la información en una aplicación web, (9) se presenta en que la información viaja a través de la red, lo que implica que puede ser sustraída por cualquier persona, no obstante, existe una serie de vulnerabilidades como la falsificación de identidad, el robo de sesiones autenticadas, la falsificación o modificación de los mensajes, entre otras. Una aplicación web que se considere segura, debe contemplar mecanismos que garanticen que efectivamente, ninguna persona ajena al sistema pueda modificar, obtener o falsificar datos de la misma.

El proyecto OWASP por sus siglas en inglés (*The Open Web Application Security Project*), (10) tiene como objetivo ofrecer una metodología, de libre acceso y utilización, que puede ser utilizada como material de referencia por parte de los arquitectos de software, desarrolladores, fabricantes y profesionales de la seguridad, involucrados en el diseño, desarrollo, despliegue y verificación de la seguridad de las aplicaciones y servicios Web.

Según esta guía, los principios básicos de seguridad que debe cumplir cualquier aplicación o servicio web son: (10)

- ✓ Validación de la entrada y salida de información: La entrada y salida de información es el principal mecanismo que dispone un atacante para enviar código malicioso contra el sistema. Por tanto, siempre debe verificarse que cualquier dato entrante o saliente, es apropiado y en el formato que se espera. Las características de estos datos deben estar predefinidas y deben verificarse en todas las ocasiones.

- ✓ Diseños simples: Los mecanismos de seguridad deben diseñarse para que sean lo más sencillo posible, huyendo de elementos que compliquen excesivamente el trabajo a los usuarios. Si los pasos necesarios para proteger de forma adecuada una función o módulo son muy complejos, la probabilidad de que estos no se ejecuten de la forma esperada será muy elevada.
- ✓ Utilización y reutilización de componentes de confianza: Debe potenciarse el principio de la reutilización en todos los casos en que sea posible. Por tanto, cuando exista un componente que resuelva un problema determinado correctamente, lo más adecuado es reutilizarlo.
- ✓ Defensa en profundidad: Nunca confiar en que un componente realizará su función de forma permanente y ante cualquier situación. Es necesario disponer de los mecanismos de seguridad suficientes para que, cuando un componente del sistema falle ante un determinado evento, otros sean capaces de detectarlo.
- ✓ Proveer servicios de autenticación segura a las aplicaciones web: Las aplicaciones deberán poseer reglas o mecanismos de autenticación donde los certificados de seguridad establecidos puedan poseer variables que se modifiquen durante el tiempo en que la sesión se encuentre activa.
- ✓ Verificación de privilegios: Los sistemas deben diseñarse para que funcionen con los menos privilegios posibles. Igualmente, es importante que los procesos únicamente dispongan de los privilegios necesarios para desarrollar su función, de forma que queden compartimentados.
- ✓ Ofrecer la mínima información: Ante una situación de error o una validación negativa, los mecanismos de seguridad deben diseñarse para que faciliten la mínima información posible. De la misma forma, estos mecanismos deben estar diseñados para que una vez denegada una operación, cualquier operación posterior sea igualmente denegada.

- ✓ Manejo de errores y auditoría: Las soluciones implementadas deben poseer un fuerte proceso de trazabilidad y control de las transacciones en las que se involucren los usuarios, para saber en todo momento, quién realizó cada una de estas y qué información sufrió cualquier tipo de modificación.



Considerando estos elementos se hace necesario cada día más, el desarrollo de aplicaciones o componentes de software especializados en el control de acceso a las aplicaciones web y la gestión de sus requerimientos de seguridad. Se debe considerar además, que el desarrollo de aplicaciones web ha alcanzado un auge sin precedentes pues las soluciones se orientan a fines y propósitos cada vez más especializados. Esto a su vez, implica que las mismas manipulen información y datos de elevada sensibilidad, la cual podría encontrarse sin la debida protección, en estados corruptos e inconsistentes, perdiendo así los principios básicos de la información, es decir, la confidencialidad, integridad y disponibilidad.

### ***Sistemas de gestión de archivos históricos.***

#### **Portal Pares**



Se trata de una plataforma común que integra las bases de datos de varios archivos de carácter histórico nacional. Pares ofrece un acceso libre y gratuito, a cualquier persona interesada en acceder a los documentos con imágenes digitalizadas de los Archivos Españoles. Además implementa la norma

internacional ISAD (G). (11)

Garantiza la seguridad de la información teniendo en cuenta los niveles de organización aunque debe adecuarse mejor a sus usuarios en lo referente a información con el fin de orientar al usuario para que se sienta cómodo en la utilización del portal.

#### **ICA-AtoM**



Es un software creado para la empresa canadiense *Artefactual Systems* bajo patrocinio del Consejo Internacional de Archivos (ICA) y de Unesco<sup>2</sup>. Utiliza una capa de abstracción de bases de datos que lo hace ser compatible con PostgreSQL, SQLite, SQLServer, Oracle, pero ha sido utilizado MySQL para su desarrollo. Utiliza el lenguaje de programación PHP para gestionar las peticiones y respuestas entre los clientes web y la lógica de la aplicación. (12)

La descripción de materiales se basa en los estándares del ICA ISAD(G), ISAAR(CPF)<sup>3</sup>, ISDF<sup>4</sup> e ISDIAH<sup>5</sup>, aunque también se pueden crear las descripciones usando otros esquemas como las reglas de descripción archivísticas canadienses(RAD) por sus siglas en inglés *Rules for Archival Description*. El control de acceso se establece teniendo en cuenta el nivel de organización de la información partiendo del uso de la norma ISAD (G).

## ArchiVenHIS

En el año 2007 surge el Sistema de Gestión de Documentos Históricos (SGDH) ArchiVenHIS, como un sistema orientado a resolver las necesidades más crecientes de almacenamiento digital de la información en el Archivo General de la Nación de la República Bolivariana de Venezuela (AGN), con las especificaciones particulares de esta institución y el propósito de ofrecer a dicha



institución una herramienta desarrollada sobre tecnologías libres y estándares abiertos que automatiza el proceso de descripción archivística. Dicha herramienta facilita el proceso de descripción de los fondos documentales bajo la custodia de la institución según la norma ISAD (G). Actualmente solo puede ser de uso

administrativo puesto que tiene implementado un control de acceso dirigido solamente a garantizar la seguridad a nivel de funcionalidad sin tener en cuenta el nivel de organización de la información.

---

<sup>2</sup> Organización de las Naciones Unidas para la Educación, la Ciencia y la Cultura. Se fundó el 16 de noviembre de 1945 con el objetivo de contribuir a la paz y a la seguridad en el mundo mediante la educación, la ciencia, la cultura y las comunicaciones.

<sup>3</sup> Norma internacional sobre los registros de autoridad de archivos relativos a entidades, personas y familias.

<sup>4</sup> Norma internacional para la descripción de funciones, proporciona una orientación para la elaboración de descripciones de funciones de instituciones asociadas con la producción y la conservación de documentos de archivo.

<sup>5</sup> Norma internacional para describir instituciones que custodian fondos de archivo.

## **Análisis de las soluciones encontradas**

Estos sistemas informáticos existentes en el mercado mundial y nacional presentan características comunes, entre las que se pueden mencionar el uso de la norma ISAD (G) para la descripción de los documentos. Luego del estudio realizado se determina, que ninguna de estas variantes soluciona el problema planteado, puesto que en algunos casos la tecnología que se utiliza es de punta y por tanto no puede ser utilizada, y para otros simplemente no existe el control de acceso con las características que debe cumplir el que se desea implementar. Todo esto ratifica la necesidad de incorporar al sistema de gestión de documentos históricos Describa el módulo de control de acceso.

## ***Modelos de control de acceso***

### **Modelo de control de acceso discrecional**

En el modelo de control de acceso discrecional (*Discretionary Access Control, DAC*) es el usuario quien decide cómo proteger el sistema, mediante controles de acceso impuestos por el sistema, es decir, el propietario del objeto es autorizado a permitir u otorgar permisos para este objeto a otros usuarios. Los sujetos pueden ser usuarios, grupos o procesos.

El modelo DAC brinda al usuario el beneficio de la flexibilidad del modelo, debido a esto es demasiado débil para controlar el acceso a los recursos de forma efectiva. El DAC para realizar operaciones sobre los recursos utiliza las Listas de Control de Acceso(ACLs), que son listas de usuarios y grupos con sus permisos específicos. DAC es apropiado en ambientes donde la compartición de información es más importante que su protección. (13)

### **Modelo de control de acceso obligatorio**

En el modelo de control de acceso obligatorio, por sus siglas en inglés (*Mandatory Access Control, MAC*) es el sistema quien protege los recursos, la autorización para que un sujeto

acceda a un objeto depende de los niveles de seguridad que tengan, ya que el administrador es el que indica qué permiso de seguridad tiene el sujeto y el nivel de sensibilidad del objeto. (13)

El modelo de seguridad MAC describe niveles de seguridad a través de un conjunto de clasificaciones discretas que siguen el modelo de clasificación de la información militar donde la confidencialidad de la información es lo más relevante (Desclasificado, Confidencial, Secreto y Sumamente Secreto); esto se conoce como política de seguridad multinivel. Un importante objetivo del modelo MAC es controlar el flujo de información en orden de asegurar su confidencialidad y su integridad, objetivo no alcanzado por los modelos DAC. Sin embargo las políticas que establece este modelo no proporcionan soluciones factibles dado que es demasiado rígido.

### **Modelo de control de acceso basado en tareas**

En el modelo de control de acceso basado en tareas, por sus siglas en inglés (*Task Based Access Control*) permite controlar el acceso en entornos representados por flujos de trabajo. Da soporte para el modelado de las tareas, la monitorización y gestión del proceso de autorización, garantizando el control de acceso mediante etapas de autorización. La activación y desactivación de los permisos de los estados de protección (son los que definen todos los permisos que pueden ser activados por la etapa de autorización) de forma dinámica en tiempo de ejecución, además permite monitorizar el uso de los permisos en tiempo de ejecución. En el modelo TBAC la especificación de políticas de seguridad es compleja, también la gestión, delegación y revocación de los privilegios son muy primitivas. (13)

### **Modelo de control de acceso basado en roles**

El principal objetivo del modelo de control de acceso basado en roles, por sus siglas en inglés (*Role Based Access Control, RBAC*) es prevenir que los usuarios tengan libre acceso a la información de la organización. (13)

La NIST<sup>6</sup> por sus siglas en inglés (*National Institute of Standard and Technology*) define de manera clara y organizada las características de la tecnología RBAC, no solo reconociendo el beneficio potencial de ésta sino que ha propuesto un estándar. El modelo de referencia de RBAC propuesto consta de cuatro componentes: usuarios, roles, operaciones y recursos. En este modelo los usuarios son asignados a uno o varios roles mientras que los permisos y privilegios se asignan a estos roles, es decir, los permisos de acceso están asociados a los roles. Este modelo permite la construcción jerárquica de las políticas de acceso por herencia o especialización. RBAC de forma general describe un grupo de usuarios que pueden estar actuando bajo un conjunto de roles y realizando operaciones en las que utilizan un conjunto de objetos como recursos.

Este modelo incluye un grupo de sesiones donde cada sesión es la relación entre un usuario y un subconjunto de roles que son activados en dicha sesión. Las decisiones de acceso se realizan sobre el conjunto de permisos asociados con los roles activos de cada sesión. Para que un usuario pueda ejercer los permisos para los cuales está autorizado, es necesario que el sistema le haya activado los roles que tienen asociados tales permisos.

En comparación con los otros 3 modelos anteriores, este último RBAC, es el idóneo para encaminar la solución propuesta por la presente investigación, pues trabaja de forma centralizada y elimina algunas abstracciones que tienen otros modelos. Esto se debe principalmente a la forma jerárquica de su funcionamiento, siendo más fácil gestionar el control de acceso a un sistema asignando roles a los usuarios.

### ***Modelo de Autenticación, Autorización y Auditoría (AAA)***

El modelo AAA se definió como un marco de funcionamiento común para gestionar las operaciones de Autenticación, Autorización y Auditoría en un escenario de control de acceso a la red, de modo que fuese compatible con las distintas tecnologías existentes, extensible, y que

---

<sup>6</sup> Instituto Nacional de Estándares y Tecnología

permitiera su gestión entre diferentes dominios. Es importante comprender qué se entiende en el modelo AAA por Autenticación, Autorización y Auditoría, para identificar correctamente los requerimientos que debe cumplir un buen sistema de este tipo: (14)

- ✓ Autenticación: Cuando un usuario o cliente requiere acceso a la red, debe presentar información personal que permita establecer de manera unívoca su identidad dentro del entorno. El usuario podrá presentar cualquier tipo de credencial de identidad (nombre de usuario y contraseña, certificado de identidad secreto o compartido), usando para ello algunos de los mecanismos de control de acceso. El servidor AAA y concretamente su servicio de autenticación, deberá obtener dichas credenciales y de forma interna (mediante bases de datos, políticas de autenticación o servicios auxiliares), determinar si ese usuario es realmente quien dice ser.
- ✓ Autorización: El proceso de autorización deberá determinar si el usuario, una vez autenticado (si es necesario), tiene o no derecho a hacer uso del servicio solicitado. Para ello, la decisión de autorización puede tomarse en función de diferentes credenciales del usuario. Por ejemplo, la autorización puede otorgarse simplemente comprobando que el usuario ha sido autenticado correctamente, aunque en entornos más especializados puede otorgarse en base a credenciales de autorización, como pueden ser sentencias de atributos que identifican el rol o papel que juega el usuario dentro de la organización. En este último caso, la manera en que el servicio AAA recupera esta información de autorización, depende del escenario concreto de acceso, ya que puede ser presentada directamente por el usuario o recuperada desde repositorios bien conocidos.
- ✓ Auditoría: Existen entornos donde se requiere llevar un control detallado de las operaciones que realizan los usuarios de los sistemas, por ejemplo, en un entorno de alta seguridad donde toda la información operacional debe ser almacenada. En este tipo de entornos, cualquier información relativa a una sesión de usuario (acceso al servicio, tipo de servicio obtenido, cantidad de tráfico enviado o recibido, tiempo de conexión), debe

almacenarse para que después pueda ser usada apropiadamente durante el proceso de auditoría.

### **Estructura básica de un sistema AAA.**

El modelo incorpora en su estructura básica un servidor genérico AAA que provee un conjunto de interfaces específicas para los procesos que este describe, de modo que puedan ser utilizadas por un conjunto de aplicaciones, componentes o servicios que necesiten los requerimientos que se especifican en el modelo. La arquitectura básica de un escenario AAA está compuesta principalmente por tres elementos: el cliente; el servidor AAA, que es el responsable de gestionar los procesos de autenticación y autorización de los usuarios; y los proveedores de servicios, que son las entidades que ofrecen los servicios finales y que pueden o no, formar parte de la infraestructura AAA. (14)

En sentido general, puede decirse que el modelo simplemente describe cómo deben llevarse a cabo los procesos descritos a modo de guía, por ello no especifica en ningún momento, los elementos funcionales que deben formar parte de la solución que lo tendrá como base teórica conceptual.

### ***Tecnologías a utilizar en la investigación.***

La selección de los lenguajes, las tecnologías y metodología utilizada para la implementación del módulo de control de acceso para el Sistema de Gestión de Documentos Históricos Dexcriba fueron establecidas por acuerdos realizados en el proyecto.

### **Lenguajes.**

Un lenguaje de programación es un idioma artificial diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. (15)

### **PHP 5.3**

PHP por sus siglas en inglés (*PHP Hypertext Pre-processor*), es un lenguaje interpretado de alto rendimiento, independiente de plataforma. Es un lenguaje de programación del lado del servidor gratuito, con una gran biblioteca de funciones y abundante documentación. Está caracterizado por ser un lenguaje que soporta la programación orientada a objetos y tiene capacidad de conexión con la mayoría de los sistemas gestores de bases de datos. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos. Tiene la capacidad de expandir su potencial utilizando módulos. (16)

### **HTML 4.0.1**

El Lenguaje de Marcas de Hipertext, por siglas en inglés (*Hypertext Markup Language*), es el lenguaje para describir la estructura de las páginas Web. El HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>). HTML ofrece a los autores los medios para:

- ✓ Publicar documentos en línea con encabezados, textos, tablas, listas, fotos.
- ✓ Diseñar formularios para realizar transacciones con servicios remotos, para su uso en la búsqueda de información, hacer reservas y pedir productos.
- ✓ Incluir hojas de cálculo, videoclips, clips de sonido y otras aplicaciones directamente en sus documentos.

La última especificación oficial de HTML se publicó el 24 de diciembre de 1999 y se denomina HTML 4.0.1. Se trata de una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas. (17)

### **JavaScript**

JavaScript es un lenguaje de programación interpretado. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador aunque existe una forma de JavaScript del lado del servidor. Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model <sup>7</sup>(DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML. (18)

## **XML**

XML (lenguaje de marcas extensible), por sus siglas en inglés (*eXtensible Markup Language*) es un lenguaje de marcas desarrollado por W3C<sup>8</sup>, acrónimo de (*World Wide Web Consortium*). Deriva del lenguaje SGML<sup>9</sup> (Estándar de Lenguaje de Marcado Generalizado) y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones se deben comunicar entre sí o integrar información.

XML no ha nacido sólo para su aplicación para Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (19)

---

<sup>7</sup> Es esencialmente una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

<sup>8</sup> Es una comunidad internacional que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

<sup>9</sup> Consiste en un sistema para la organización y etiquetado de documentos.



## CSS 2.1

CSS (Hojas de Estilo en Cascada), por sus siglas en inglés (*Cascading Style Sheets*) es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos. (20)

Esta especificación define las Hojas de Estilo en Cascada, Nivel 2 revisión 1. CSS 2.1 se basa en CSS 2, y este a su vez en CSS 1. Corrige algunos errores en CSS, el más importante es una nueva definición de la “altura/anchura” de los elementos con posición absoluta. (21)

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Internet Explorer	Trident	Completo desde la versión 6.0	Completo desde la versión 8.0	Prácticamente nulo
Firefox	Gecko	Completo	Casi completo	Selectores, pseudo-clases y algunas propiedades
Safari	WebKit	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo	Casi completo	Todos los selectores, pseudo-clases y muchas propiedades

## Marco de trabajo de desarrollo.

El término marco de trabajo, se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un *framework* se puede considerar como una aplicación genérica incompleta y configurable a la que puede añadirse las últimas piezas para construir una aplicación concreta. (22)

### **JQuery 1.8.3**

Jquery es un *framework* (marco de trabajo) de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, permitiendo manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX<sup>10</sup> al sistema.

JQuery consiste en un único fichero JavaScript que contiene las funcionalidades comunes de DOM, eventos, efectos y AJAX. La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. (23)

### **JQuery UI 1.9.2**

JQuery UI (Interfaz de usuario) por sus siglas en inglés (*Jquery User Interface*) es una biblioteca de componentes para el framework jQuery que le añaden un conjunto de *plug-ins*<sup>11</sup>, *widgets*<sup>12</sup> y efectos visuales para la creación de aplicaciones web.

Es un complemento que permite implementar componentes diversos para generar interfaces de usuario en páginas web, además de otras funcionalidades básicas para crear aplicaciones web enriquecidas. Como su propio nombre indica, está basado en el popular framework JavaScript. (24)

### **CodeIgniter 2.1.3**

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su principal objetivo es brindarle la facilidad a los usuarios de desarrollar proyectos con mayor rapidez de lo que alcanzaría si lo tuvieran que escribir desde cero, proporcionándole una serie de librerías para tareas comúnmente necesarias como: acceder a una base de datos, enviar email, validar datos de un formulario, mantener sesiones, manipular imágenes, entre otras.

---

<sup>10</sup> Acrónimo para Asynchronous JavaScript + XML, en realidad no es una tecnología sino la combinación de muchas tecnologías.

<sup>11</sup> Complemento, conector o extensión

<sup>12</sup> Componente reutilizable de la interfaz de usuario.

Es importante destacar que este framework implementa el patrón arquitectónico Modelo-Vista-Controlador (MVC), que es un estándar de programación que puede ser utilizado tanto para la realización de sitios web como para programas tradicionales. Además, muchas de sus utilidades y modos de funcionamiento son opcionales, lo que hace que el usuario goce de una mayor libertad a la hora de desarrollar aplicación web. (25)

### **Sistema gestor de base de datos.**

Un sistema Gestor de Base de Datos es un software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. (26)

### **PostgreSQL 9.1**

PostgreSQL es un Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos, derivado de Postres, desarrollado en la Universidad de California, en el Departamento de Ciencias de la Computación de Berkeley. Es un gestor de bases de datos de código abierto, brinda un control de concurrencia multi-versión que permite trabajar con grandes volúmenes de datos; soporta gran parte de la sintaxis SQL y cuenta con un extenso grupo de enlaces con lenguajes de programación.

Posee características significativas del motor de datos, entre las que se pueden incluir las su consultas, los valores por defecto, las restricciones a valores en los campos (*constraints*) y los disparadores (*triggers*). Ofrece funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, conversión de tipos y entrada de enteros binarios y hexadecimales.

El código fuente se encuentra disponible para todos sin costo alguno. Está disponible para 34 plataformas con la última versión estable. Es totalmente compatible con ACID (*acrónimo de*

*Atomicity, Consistency, Isolation and Durability*; en español: Atomicidad, Consistencia, Aislamiento y Durabilidad).

Debido a la liberación de la licencia, PostgreSQL se puede usar, modificar y distribuir de forma gratuita para cualquier fin, ya sea privado, comercial o académico. (26)

### **Servidor web.**

Un servidor web es un programa que sirve datos en forma de Páginas Web, hipertextos o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. La comunicación de estos datos entre cliente y servidor se hace por medio un protocolo, concretamente del protocolo HTTP. Con esto, un servidor Web se mantiene a la espera de peticiones HTTP, que son ejecutadas por un cliente HTTP; lo que suele conocerse como un Navegador Web. (27)

### **Apache 2.2**

Está diseñado para ser un Servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular.

Este diseño permite a los administradores de Sitios Web elegir qué características van a ser incluidas en el servidor seleccionando qué módulos se van a cargar, ya sea al compilar o al ejecutar el servidor.

Además, es gratuito, y de código abierto, así que puede decirse que corre sobre cualquier plataforma. Apache es una muestra, al igual que el sistema operativo Linux, de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar. (27)

## **Entorno de desarrollo integrado**

Un entorno de desarrollo (IDE) es un entorno de programación que ha sido empaquetado como una aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDE proveen un marco de trabajo de fácil uso para la mayoría de los lenguajes de programación. (28)

### **Netbeans 7.3**

Netbeans es un proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento y con cerca de cien socios en todo el mundo. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Tiene soporte para varios lenguajes incluyendo PHP, JavaScript, HTML y CSS. (29)

Dentro de las características de Netbeans se pueden encontrar las siguientes:

- ✓ Posee una integración completa en términos de administración básica y avanzada de MySQL.
- ✓ El depurado de las aplicaciones es más sencillo.
- ✓ Autocompleta código de los lenguajes que soporta.
- ✓ Compatibilidad total con PHP 5.3.
- ✓ Es multiplataforma.

## **Herramientas de modelado.**

### **Visual Paradigm 8.0**

Visual Paradigm para UML es una herramienta de modelado profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite construir todos los tipos de

diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (30)

### **Lenguaje Unificado de Modelado 2.1 (UML)**

UML (Lenguaje Unificado de Modelado) por sus siglas en inglés (*Unified Modeling Language*) es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el ciclo de desarrollo de un sistema de software. UML ofrece un estándar para describir modelos, incluyendo aspectos conceptuales tales como procesos de negocio y funciones de sistema, además de aspectos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reutilizables. UML no es una guía para realizar el análisis y diseño orientado a objetos, es decir, no es un proceso. UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

El Lenguaje Unificado de Modelado prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, y describe la semántica esencial de lo que estos diagramas y símbolos significan. (30)

### **Metodología de desarrollo de software**

Surge ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental para desarrollar un producto (31). Se utiliza para el desarrollo del módulo la metodología RUP según lo establece el Programa de mejora que la UCI lleva a cabo para alcanzar una certificación internacional del nivel 2 de CMMI<sup>13</sup>, con el objetivo de lograr productos y servicios de alta calidad y aumentar la productividad.

### **Proceso Unificado de Desarrollo de Software (RUP)**

RUP unifica los mejores procesos de metodologías anteriores, es orientado a objetos y está preparado para desarrollar grandes y complejos proyectos. Se caracteriza por ser iterativo e

---

<sup>13</sup> Modelo de referencia para el crecimiento de capacidades y madurez, que se enfoca tanto en procesos de administración como de ingeniería de sistemas y software.

incremental, dirigido por casos de uso y centrado en la arquitectura. Las actividades se agrupan en grupos lógicos, definiendo 9 flujos de trabajo principales que guían el desarrollo del software a través de las 4 fases: Inicio, Elaboración, Construcción y Transición, que componen el ciclo de vida del proyecto. En su modelación define un grupo de elementos que determinan quién, qué, cómo y cuándo se realizarán las actividades: (32)

- ✓ Trabajadores (“quién”): definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.
- ✓ Actividades (“cómo”): son tareas que tienen un propósito claro, son realizadas por los trabajadores.
- ✓ Artefactos (“qué”): productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.
- ✓ Flujo de actividades (“cuándo”): secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.

## **Conclusiones**

Luego de investigar acerca de los fundamentos teóricos que constituyen la base para la concepción, diseño e implementación de la solución propuesta se concluye que, ninguno de los sistemas de gestión de documentos de archivos históricos estudiados da respuesta al problema planteado por ello se decide desarrollar una nueva solución. Además será utilizado el modelo de control de acceso basado en roles siguiendo la propuesta de principios básicos de seguridad que propone el proyecto OWASP.

## **Capítulo II: Características del módulo**

### ***Introducción***

En este capítulo se realiza una descripción de la solución propuesta a través del modelo de dominio y una descripción de los procesos que serán objeto de automatización proporcionando un mejor entendimiento del sistema. También son especificados los requisitos funcionales y no funcionales que debe cumplir el módulo, así como la descripción y representación de los casos de uso del sistema. Se diseña el modelo de datos y los diagramas de clases del diseño mostrando además los patrones utilizados para ello.

### ***Modelo de dominio***

Un Modelo Conceptual o Modelo de Dominio, constituye una representación visual para el usuario de los conceptos u objetos significativos del mundo real para un problema o área de interés. Representa conceptos del mundo real, no de los componentes de software, mediante clases conceptuales del dominio del problema, encargándose de capturar los tipos más importantes de objetos y eventos que suceden en el entorno. (33)

La creación de un modelo del dominio posibilita identificar conceptos asociados con el entorno en que se desarrolla la solución y mostrar las relaciones entre ellos. En el siguiente modelo se muestran las relaciones que existen entre los principales conceptos que componen el Sistema de Control de Acceso, destacándose los aspectos organizativos que deben ser tomados en cuenta para su desarrollo.

En el presente trabajo de diploma se realiza el modelo de dominio debido ya que no existe un negocio definido, por lo cual, no se pueden determinar los procesos y roles del proceso de negocio, haciéndose engorrosa y poco exacta la descripción de los mismos.



## Diagrama de clases del modelo del dominio

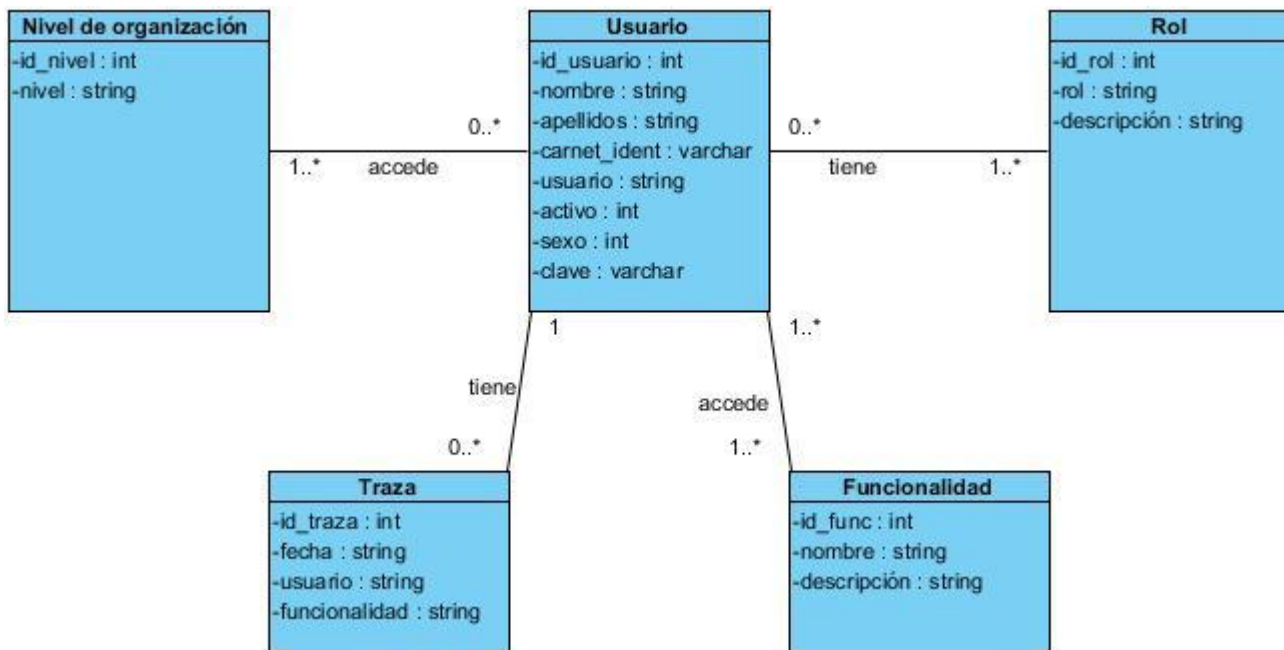


Figura 1: Modelo de dominio

A continuación se describen las clases del modelo de dominio, para su mejor comprensión:

**Usuario:** Persona que una vez registrada en el sistema en correspondencia al rol que tenga asignado, puede acceder a las funcionalidades que está autorizado.

**Rol:** Función que cumple determinado usuario en el sistema.

**Traza:** Acción realizada por determinado usuario sobre la documentación en el sistema.

**Funcionalidad:** Funcionalidades del sistema.

**Nivel de organización:** Niveles de organización definidos por la norma ISAD (G).

## ***Requisitos funcionales***

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que se mantienen invariables sin importar con qué propiedades o cualidades se relacionen. (33)

**RF1** Autenticar Usuario: debe permitir al usuario autenticarse en el sistema.

**RF2** Registrar usuario: debe permitir al Administrador del sistema registrar usuarios.

**RF3** Listar usuario: debe permitir al Administrador listar los usuarios registrados en el sistema.

**RF4** Eliminar usuario: debe permitir al Administrador del sistema eliminar usuarios.

**RF5** Modificar usuario: debe permitir al Administrador modificar los datos deseados de los usuarios registrados en el sistema.

**RF6** Buscar usuario: debe permitir al Administrador del sistema buscar usuario.

**RF7** Crear rol: debe permitir al Administrador del sistema crear roles.

**RF8** Modificar rol: debe permitir al Administrador del sistema modificar roles.

**RF9** Eliminar rol: debe permitir al Administrador del sistema eliminar roles.

**RF10** Listar rol: debe permitir al Administrador del sistema listar roles.

**RF11** Buscar rol: debe permitir al Administrador del sistema buscar roles.

**RF12** Asignar usuario a rol: debe permitir al Administrador del sistema asignar usuario a rol.

**RF13** Asignar funcionalidad a rol: debe permitir al Administrador del sistema asignar funcionalidades a roles.

**RF14** Registrar traza: debe permitir registrar las trazas originadas por cada usuario.

**RF15** Eliminar traza: debe permitir eliminar trazas.

**RF16** Listar traza: debe permitir listar las trazas.

**RF17** Buscar traza sencillo: debe permitir realizar una búsqueda de traza de forma sencilla.

**RF18** Buscar traza avanzado: debe permitir realizar una búsqueda de traza de forma avanzada.

**RF19** Crear funcionalidad: debe permitir al administrador crear las funcionalidades del sistema.

**RF20** Modificar funcionalidad: debe permitir al administrador modificar las funcionalidades del sistema.

**RF21** Listar funcionalidad: debe permitir al administrador listar las funcionalidades del sistema.

**RF22** Eliminar funcionalidad: debe permitir al administrador eliminar la funcionalidad que desee del sistema.

**RF23** Buscar funcionalidad: debe permitir al administrador buscar dentro de las funcionalidades del sistema la que desee consultar.

**RF24** Registrar usuario: debe permitir al usuario registrarse en el sistema en caso de que no exista en la base de datos.

**RF25** Asignar nivel de organización a rol: debe permitir al Administrador del sistema asignar niveles de organización a roles.

### ***Requisitos no funcionales***

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales.

(33)

Los requisitos no funcionales de la aplicación han sido agrupados por categorías y se muestran a continuación:

#### **Software**

- ✓ **RNF1.** Servidor de bases de datos PostgreSQL 9.1 o superior.
- ✓ **RNF2.** Servidor web Apache 2.2 o superior.
- ✓ **RNF3.** Navegador web: Mozilla Firefox 11.0 o superior, Google Chrome 17.0 o superior .

#### **Soporte**

- ✓ **RNF4.** Los componentes de software que integran la solución se organizarán de forma modular.

#### **Hardware**

- ✓ **RNF5-** Servidor de base de datos y servidor web con las siguientes prestaciones:

RAM: 1 GB.

Procesador: dos núcleos a 2.4 GHZ +.

### Diseño

- ✓ **RNF6**- Lenguaje de programación: PHP.
- ✓ **RNF7**- *Framework* de desarrollo: CodeIgniter 2.1.3.
- ✓ **RNF8**- Entorno de desarrollo integrado: Netbeans.

### Actores del sistema

Los actores del sistema representan el rol que juega una o varias personas, un equipo o un sistema automatizado, son parte del sistema y pueden intercambiar información con él o ser recipientes pasivos de información. (33)

En la tabla que aparece a continuación se definen los actores que interactúan con el sistema.

Actores	Justificación
Administrador del sistema	El administrador es el encargado de gestionar los usuarios internos del sistema. Además de garantizar los roles y los permisos que a estos se asignen.
Usuario externo	El usuario externo podrá realizar búsquedas sobre los documentos incorporados al sistema, puede visualizar las imágenes asociadas a ellos o visualizar su descripción.

Tabla 1: Actores del sistema

### Definición de Casos de Uso (CU) del Sistema

Los casos de uso son artefactos narrativos que describen bajo la forma de acciones y reacciones el comportamiento del sistema desde el punto de vista del usuario. Cada CU proporciona uno o

más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. (33)

### **Patrón de Caso de Uso utilizado**

CRUD por sus siglas en inglés (*Creating, Reading, Updating, Deleting*)

El patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

**CRUD Completo:** Este patrón modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico tales como creación, lectura, actualización y eliminación.

En este sistema se hace uso de dicho patrón en varios casos de usos, entre ellos pueden encontrarse:

- Gestionar usuario: compuesto por las funciones de crear, modificar, eliminar y listar usuario.
- Gestionar rol: compuesto por las funciones de crear, modificar, eliminar y listar rol.
- Gestionar funcionalidad: compuesto por las funciones de crear, modificar, eliminar y listar funcionalidad.

## Diagrama de Caso de Uso del Sistema.

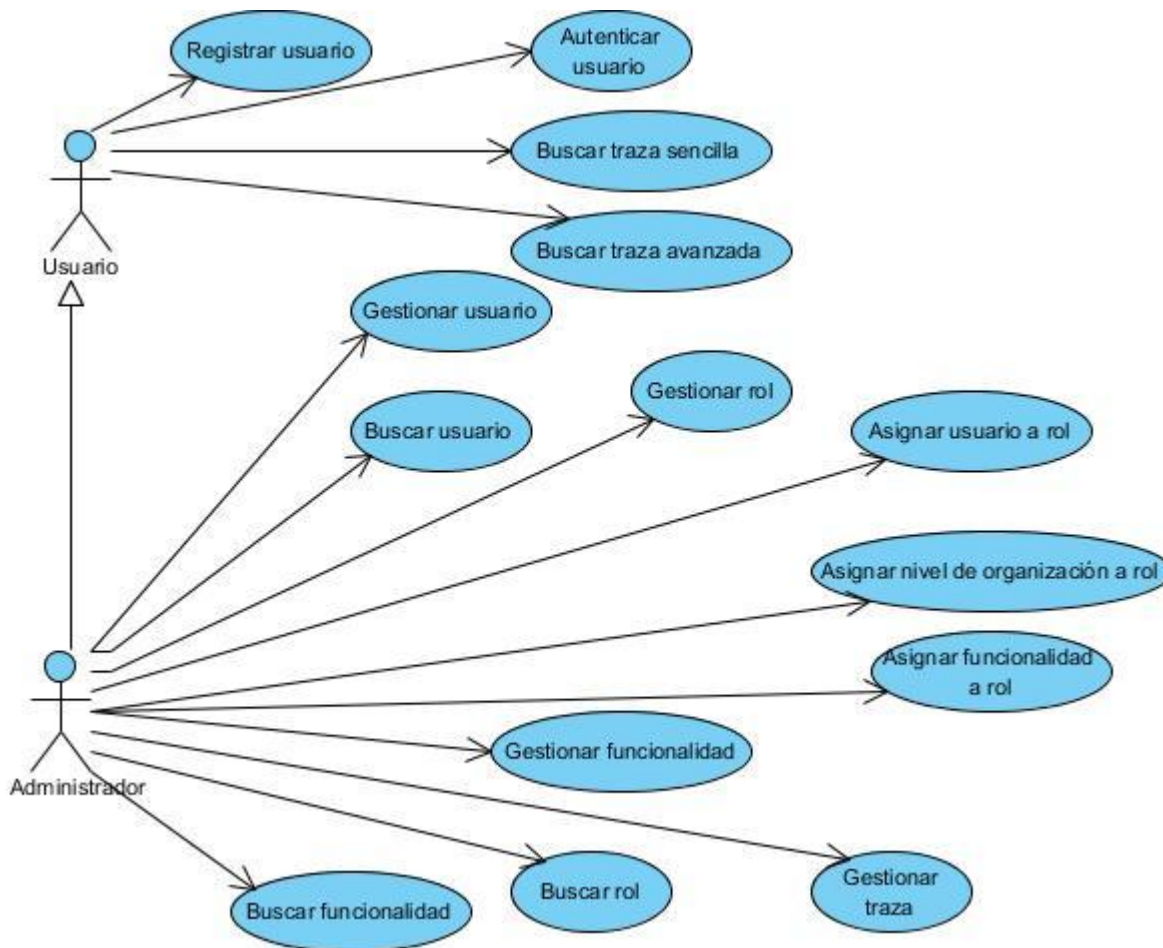


Figura 2: Diagrama de Caso de uso del sistema.

## Descripción de los Casos de Uso del sistema.

A continuación se describen 2 de los casos de uso críticos del módulo, el resto de las descripciones pueden ser consultadas en la sección de anexos.

<b>Objetivo</b>	Permite asignar una funcionalidad a determinado rol.	
<b>Actores</b>	Administrador.	
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a asignar funcionalidades a los roles registrados en el sistema y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema.	
<b>Pos condiciones</b>	Una vez que se hayan realizado satisfactoriamente las acciones llevadas a cabo por el usuario, el sistema almacena toda la información en la base de datos.	
<b>Flujo de eventos</b>		
<b>Flujo básico " Asignar funcionalidad a rol".</b>		
	<b>Actor</b>	<b>Sistema</b>
	1.-El usuario accede a la opción Gestionar rol.	2.- El sistema muestra una interfaz con todos los roles que se encuentran en el sistema.
	3.-El usuario selecciona el rol que desea asignarle la funcionalidad(es).	4.- El sistema muestra las funcionalidades que no se le han asignado a dicho rol, y otro listado con las que ya tiene asignadas.
	5.- El usuario selecciona la(s) funcionalidad(es) que desea asignar.	6.- El sistema muestra el mensaje de éxito "Se ha asignado correctamente".

Tabla 2: Descripción del caso de uso: Asignar funcionalidad a rol

<b>Objetivo</b>	Permite adicionar, eliminar, modificar y listar roles al sistema.	
<b>Actores</b>	Administrador.	
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a Gestionar rol, puede adicionar, listar, modificar y eliminar los roles y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.	
<b>Complejidad</b>	Alta	
<b>Prioridad</b>	Alta	
<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema.	
<b>Pos condiciones</b>	Una vez que se hayan realizado satisfactoriamente las acciones llevadas a cabo por el usuario, el sistema almacena toda la información en la base de datos.	
<b>Flujo de eventos</b>		
<b>Flujo básico " Adicionar rol".</b>		
	<b>Actor</b>	<b>Sistema</b>
	1.- El caso de uso inicia cuando el actor accede a la opción Gestionar rol.	2.- El sistema muestra una interfaz con todos los roles que se encuentran en el sistema.
	3.- El actor selecciona la opción Adicionar.	4.- El sistema le muestra un formulario con todos los campos especificados para adicionar un rol.
	5.- El actor oprime el botón Registrar.	6.- El sistema valida los datos y guarda la información, mostrando un mensaje de éxito. (Ver flujo alterno 1).
<b>Flujo alterno 1</b>		



<b>Actor</b>	<b>Sistema</b>
1.- El actor no introduce todos los datos del formulario al adicionar un rol.	2.- El sistema muestra un mensaje de error " El rol no puede estar vacío ".
<b>Sección 2: " Eliminar rol".</b>	
<b>Actor</b>	<b>Sistema</b>
1.- El caso de uso inicia cuando el actor accede a la opción Gestionar rol.	2.- El sistema muestra una interfaz con todos los roles que se encuentran en el sistema.
3.- El actor selecciona el o los roles que desea eliminar y la opción Eliminar.	4.- El sistema valida y guarda los datos mostrando un mensaje de éxito. (Ver flujo alternativo 2).
<b>Flujo alternativo 2</b>	
<b>Actor</b>	<b>Sistema</b>
1.- El actor selecciona la opción Eliminar sin seleccionar los elementos.	2.- El sistema muestra un mensaje de error " Debe seleccionar al menos un elemento ".
<b>Sección 3: " Modificar rol".</b>	
<b>Actor</b>	<b>Sistema</b>
1.- El caso de uso inicia cuando el actor accede a la opción Gestionar rol.	2.- El sistema muestra una interfaz con todos los roles que se encuentran en el sistema.
3.- El actor selecciona el rol y la opción Modificar.	4.- El sistema muestra un formulario con todos los datos de ese rol.
5.- El actor modifica los datos que desea y oprime el botón Modificar.	6.- El sistema valida y guarda los datos mostrando un mensaje de éxito. (Ver flujo alternativo 3).
<b>Flujo alternativo 3</b>	
<b>Actor</b>	<b>Sistema</b>
1.- El actor no introduce todos los datos del formulario al modificar un rol.	2.- El sistema muestra un mensaje de error " Quedan campos vacíos ".

Sección 4: " Listar rol".	
Actor	Sistema
1.- El caso de uso inicia cuando el actor accede a la opción Gestionar rol.	2.- El sistema muestra una interfaz con todos los roles que se encuentran en el sistema.

Tabla 3: Descripción del caso de uso: Gestionar rol.

### **Diseño del sistema.**

La importancia del diseño del software se puede describir con una sola palabra, calidad. El diseño es el lugar en donde se fomentará la calidad en la ingeniería del software. Es la única forma de convertir exactamente los requisitos de un cliente en un producto o sistema de software finalizado. Este sirve como fundamento para todos los pasos siguientes del soporte del software y de la ingeniería del software. Sin un diseño, se corre el riesgo de construir un sistema inestable. (33)

A continuación se realizará la descripción de las clases que componen los diagramas de clases del diseño, se utilizan para una mejor comprensión de los modelos los siguientes prototipos:

**CP\_<Nombre de la página>:** Son las páginas que funcionan como interfaz a los usuarios. Se construirán dinámicamente para ser visualizadas en el explorador de los usuarios.

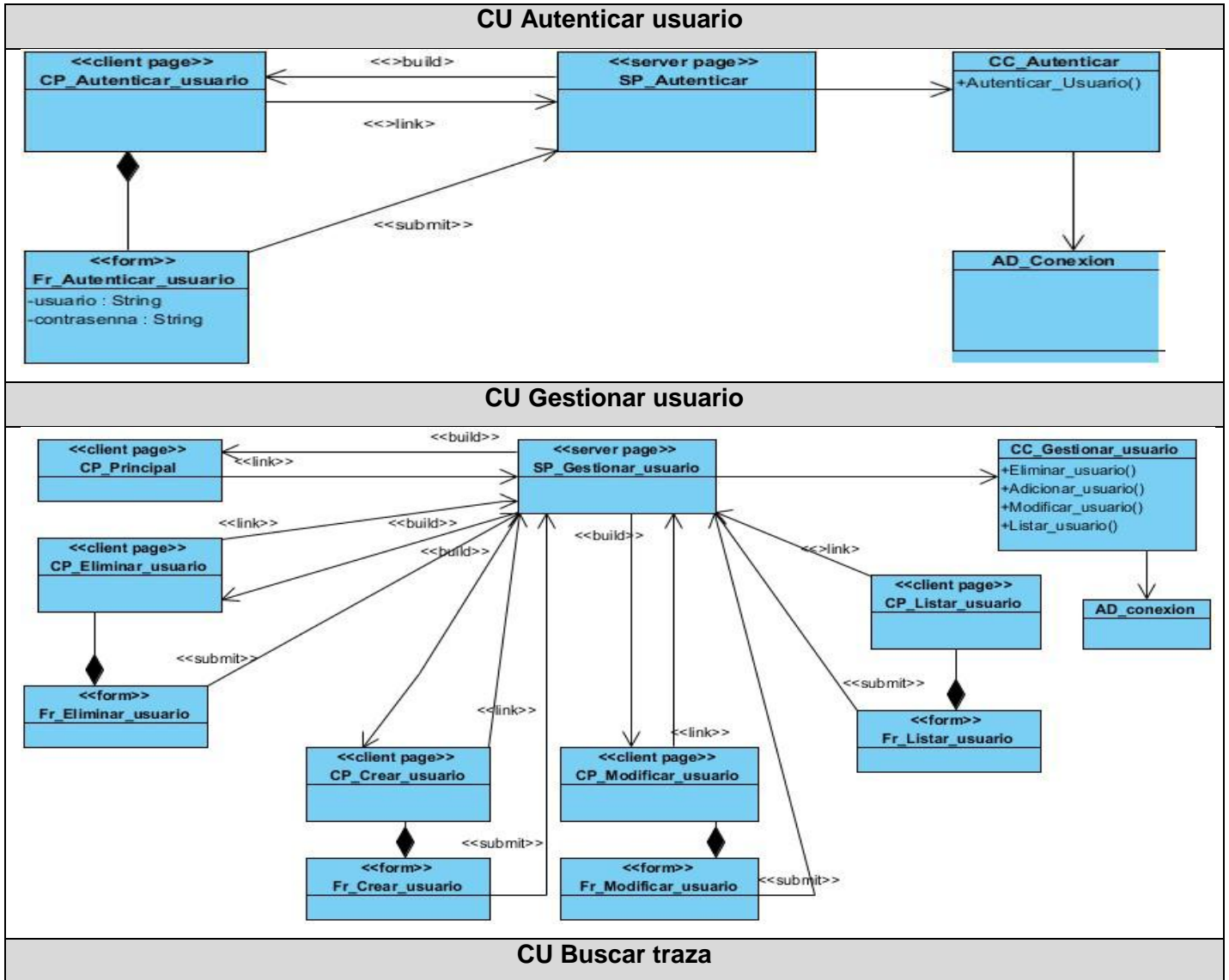
**Form\_<Nombre del formulario>:** Son los formularios que se utilizan para obtener los datos introducidos por el usuario en cada una de las actividades que se realiza durante el procesamiento de un documento.

**SP\_<Nombre de la página>:** Son las páginas servidoras que construyen a las páginas clientes y tienen toda la lógica de presentación.

### **Diagramas de clases del diseño.**

Un diagrama de clases representa las clases que serán utilizadas dentro del sistema y las relaciones que existen entre ellas. Visualiza las relaciones entre las clases involucradas en el sistema, las cuales pueden ser asociativas y de herencia, entre otras.

A continuación se muestran algunos de los diagramas relacionados con el módulo, los restantes pueden consultarse en el expediente de proyecto:



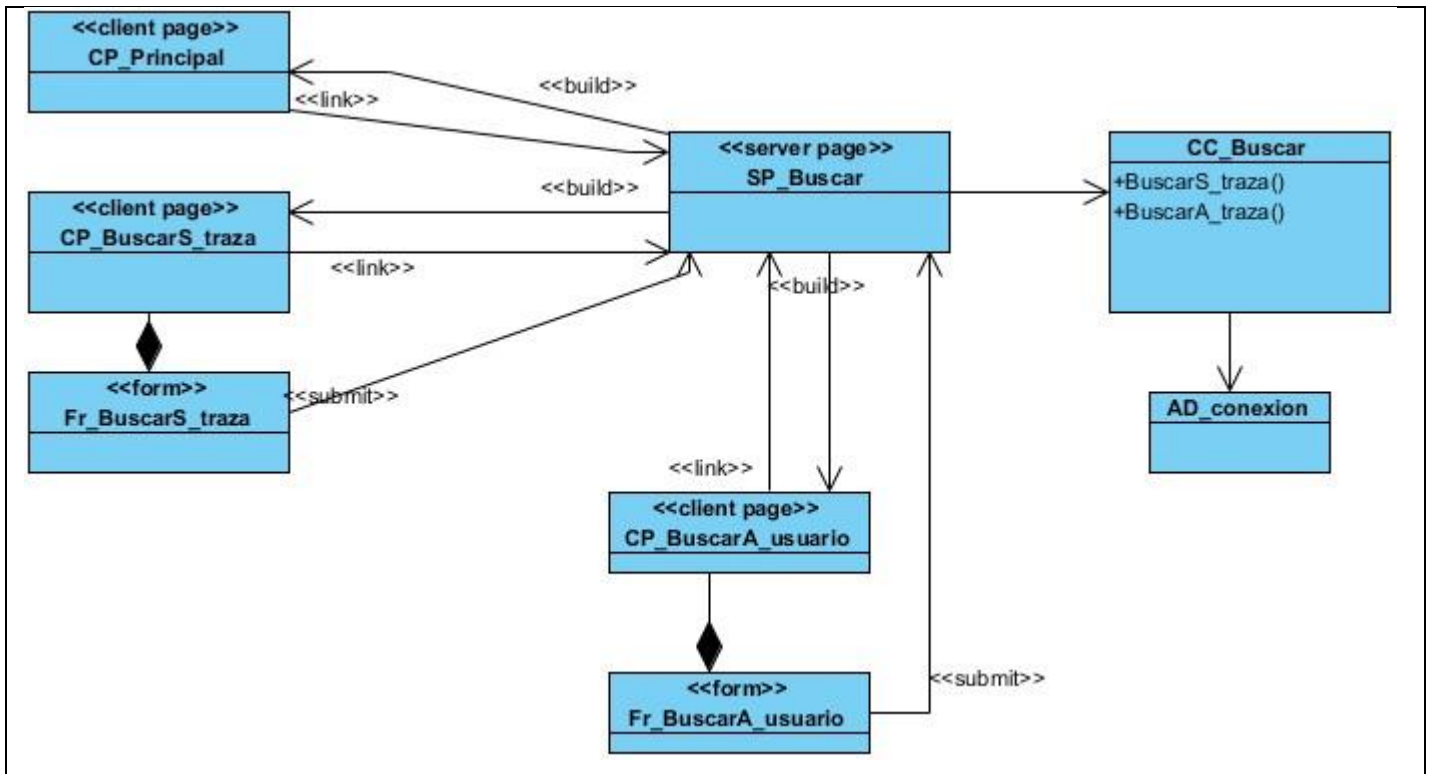


Tabla 4: Diagramas de clases del diseño.

### Modelo de datos.

Un modelo de datos no es más que la representación de un fenómeno de la realidad objetiva a través de los objetos, sus propiedades y las relaciones que se establecen entre ellos. Permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí. (34)

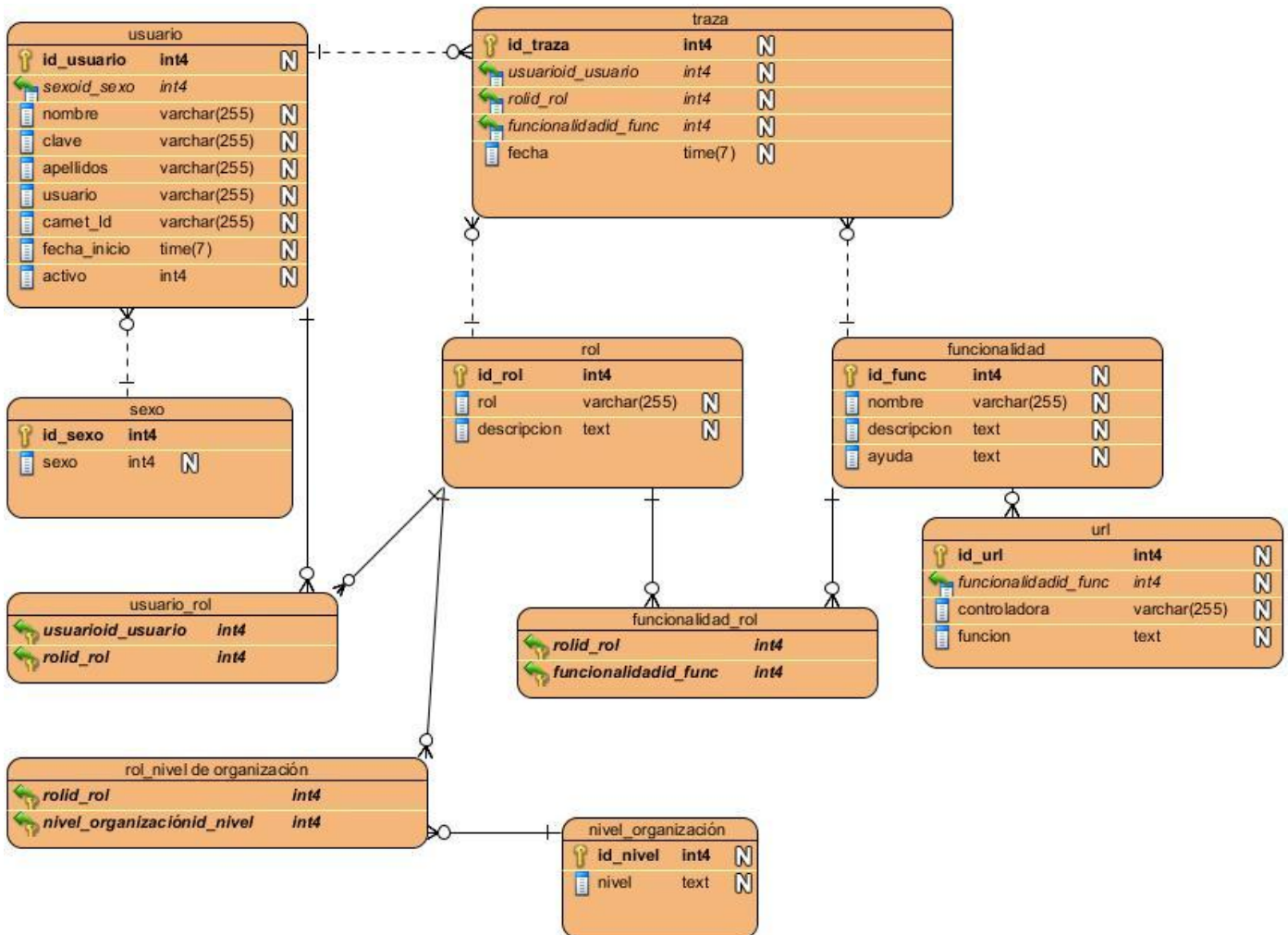


Figura 3 Modelo de datos.

### Descripción de las tablas del modelo de datos

Tabla	usuario	
Descripción	En esta tabla se almacenan todos los datos del usuario	
Atributos	Tipo	Descripción
id_usuario	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor.

sexoid_sexo	Int(4)	Identifica a la tabla que pertenece (llave foránea)
nombre	Varchar(255)	Nombre del usuario
clave	Varchar(255)	Clave del usuario
apellidos	Varchar(255)	Apellidos del usuario
usuario	Varchar(255)	Usuario del sistema
carnet_id	Varchar(255)	Carnet de identidad del usuario
fecha_inicio	Time(7)	Fecha de inicio de registro del usuario en el sistema
activo	Int(4)	1_activo y 0_no activo

Tabla 5: Modelo de datos entidad: usuario

<b>Tabla</b>	traza	
<b>Descripción</b>	En esta tabla se almacenan todas las trazas de los usuarios en el sistema.	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_traza	int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor
usuarioid_usuario	Int(4)	Identifica a la tabla que pertenece(llave foránea)
rolid_rol	Int(4)	Identifica a la tabla que pertenece(llave foránea)
funcionalidadid_func	Int(4)	Identifica a la tabla que pertenece(llave foránea)
fecha	Time(7)	Fecha de la traza,

		incluye la hora.
--	--	------------------

Tabla 6: Modelo de datos entidad: traza

<b>Tabla</b>	funcionalidad	
<b>Descripción</b>	Se almacenan todas las funcionalidades del sistema	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_func	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor
nombre	Varchar(255)	Nombre de la funcionalidad
descripción	Text	Características de la funcionalidad
ayuda	Text	Ofrece una ayuda de la funcionalidad

Tabla 7: Modelo de datos entidad: funcionalidad

<b>Tabla</b>	rol	
<b>Descripción</b>	Se almacenan todos los roles del sistema	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_rol	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor
rol	Varchar(255)	Nombre del rol
descripción	Text	Características del rol

Tabla 8: Modelo de datos entidad: rol

<b>Tabla</b>	url	
<b>Descripción</b>	Almacena los datos de las url de las funcionalidades.	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_url	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor.
funcionalidadid_func	Int(4)	Identifica a la tabla que pertenece(llave foránea)
controladora	Varchar(255)	Clase controladora
Función	Text	Funcionalidad del sistema.

Tabla 9: Modelo de datos entidad: url

<b>Tabla</b>	nivel_organización	
<b>Descripción</b>	Almacena los niveles de organización de la información.	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_nivel	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor.
nivel	Text	Niveles definidos por la norma ISAD (G).

Tabla 10: Modelo de datos entidad: nivel\_organización.

## **Patrones.**

Un patrón de arquitectura es una plantilla para una arquitectura de aplicaciones. Los patrones especifican las propiedades generales de la estructura del sistema y repercuten en la arquitectura de sus subsistemas. La selección de un patrón de arquitectura es por lo tanto una decisión



fundamental al desarrollar un sistema de software. El objetivo de los patrones es crear un lenguaje común a una comunidad de desarrolladores, para comunicar experiencias sobre los problemas y sus soluciones. (35)

Para la implementación del módulo se utilizaron los patrones de diseño GRASP (patrones generales de software para asignar responsabilidades), por sus siglas en inglés (*General Responsibility Assignment Software Patterns*).

A continuación se describe el uso de estos en el módulo.

- ✓ **Creador:** se utiliza en las clases controladoras al instanciar, a través del objeto load de la clase loader, las vistas y los modelos.
- ✓ **Controlador:** consiste en asignar la responsabilidad de controlar el flujo de eventos del sistema. Este patrón se utiliza en las clases controladoras que son las que se encargan de obtener datos y enviarlos a las vistas, en este caso particular se evidencia en la clase controladora control\_acceso.
- ✓ **Bajo acoplamiento y Alta Cohesión:** la propia implementación del *framework* utilizado, contiene estos patrones nivelados, puesto que permite el uso de los componentes de forma individual, evidenciando el bajo acoplamiento así como la dependencia entre ellos o alta cohesión.

## ***Conclusiones.***

Al culminar esta fase puede decirse que, con la elaboración del modelo de dominio ya existe una idea más clara y definida del negocio. Se ha logrado un gran por ciento del desarrollo del módulo, con la identificación de los requisitos y la elaboración del diseño ya todo está listo para comenzar la implementación.

## Capítulo III: Implementación y Prueba del módulo

### ***Introducción.***

En el presente capítulo quedan reflejados los principales elementos referentes a la implementación del módulo, los diagramas de componentes y el diagrama de despliegue. Se incluye, además, todo lo referente a las pruebas que se realizaron al sistema.

### ***Diagrama de componentes.***

Un diagrama de componentes muestra cómo se organizan los elementos del sistema y las dependencias existentes entre ellos. A continuación se muestra el diagrama de componentes de forma global:

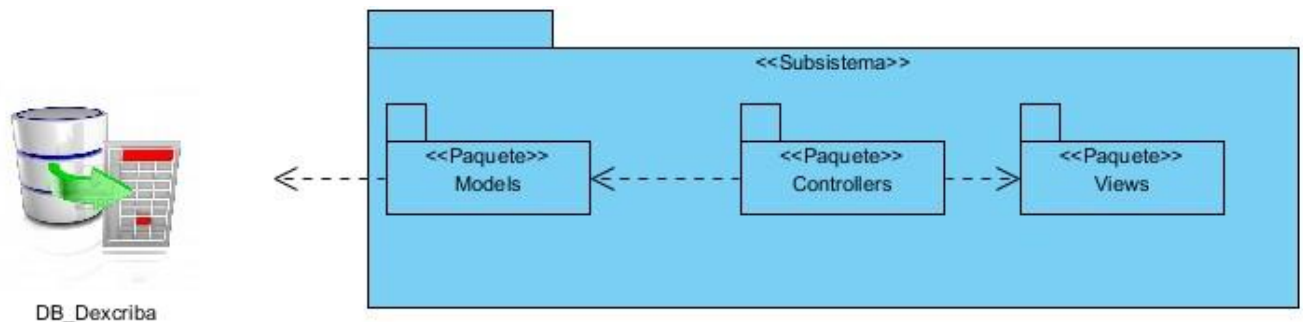


Figura 4: Diagrama de componentes.

### **Diagrama de componentes del paquete Modelo.**

El paquete Modelo es el responsable de interactuar con la capa de almacenamiento de datos, de esta forma gestiona la información con la que trabaja el sistema. A continuación se presentan de forma detallada los componentes de dicho paquete.

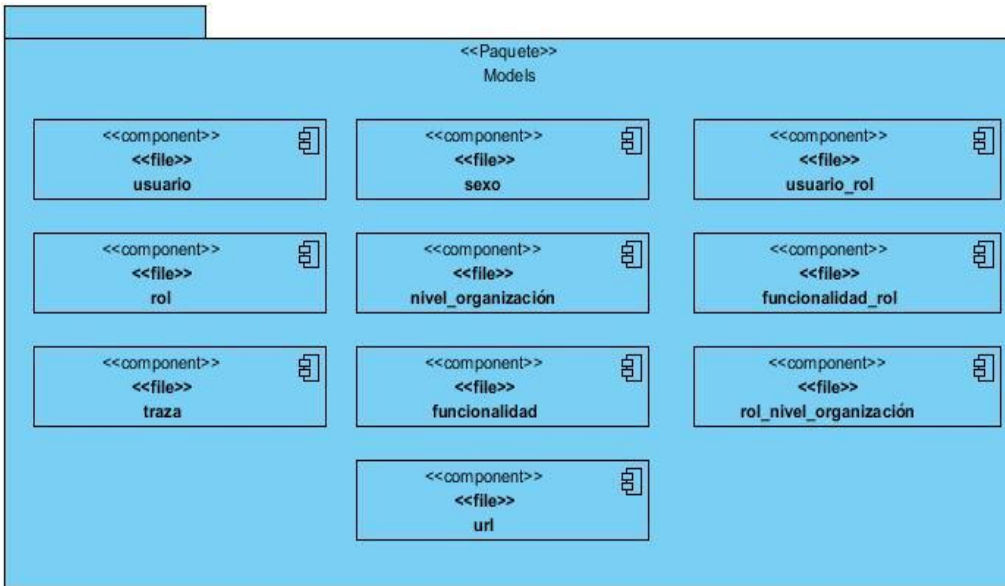


Figura 5: Diagrama de componentes-Paquete Modelo.

### Diagrama de componentes del paquete Control.

El paquete Control es el rector de las actividades de la aplicación, este contiene los ficheros de código fuente que interactúan con los demás paquetes, coordinando las acciones del software. A continuación se representa de forma detallada el componente de dicho paquete.



Figura 6: Diagrama de componentes-Paquete Control.

## Diagrama de componentes del paquete Vistas.

En el paquete Vistas se agrupan los componentes necesarios para la interacción del usuario con el sistema, los cuales son manejados por el paquete de Control. A continuación se representan de forma detallada los componentes de dicho paquete.

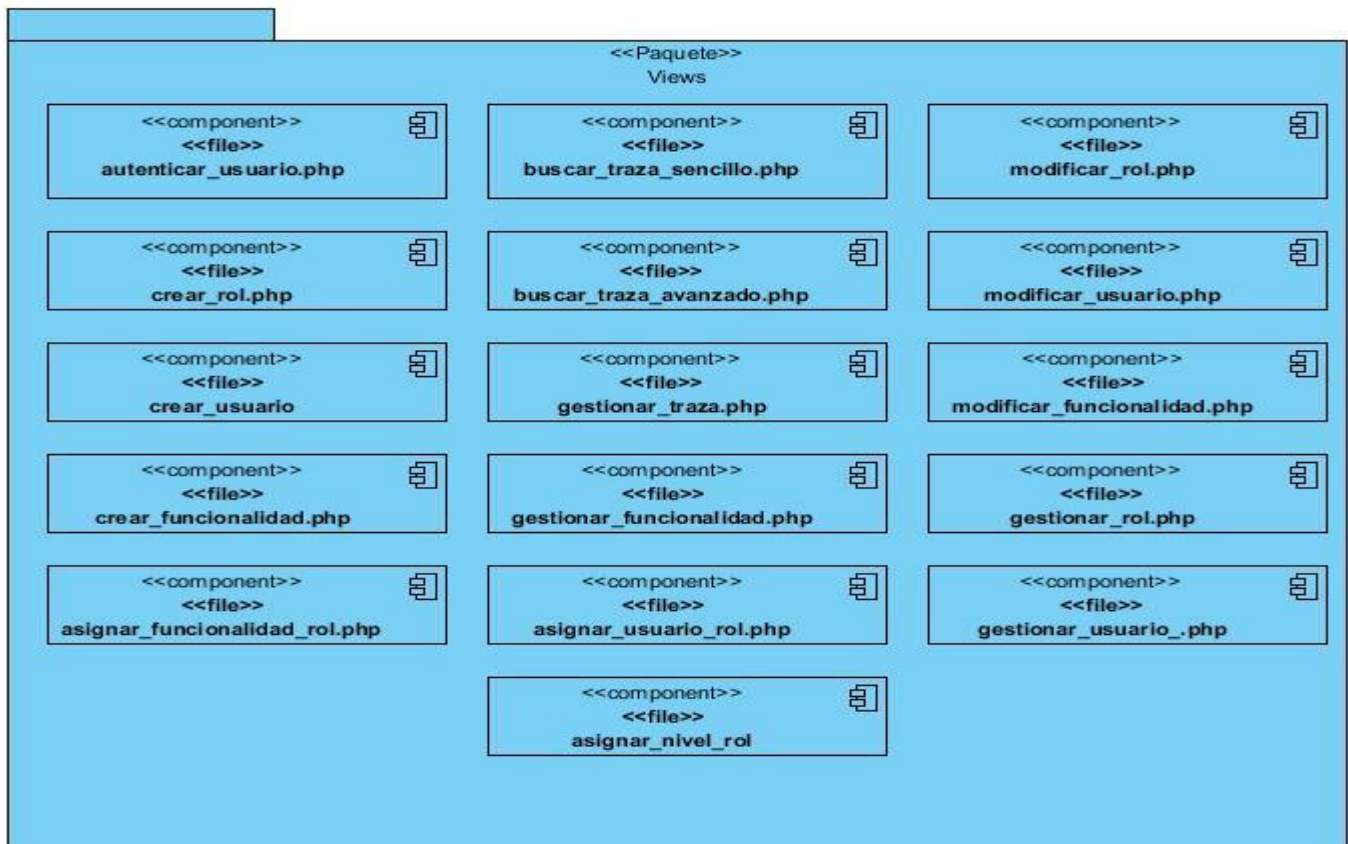


Figura 7: Diagrama de componentes-Paquete Vistas.

## Diagrama de despliegue.

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados mediante enlaces de comunicación. (36)

A continuación se describen los elementos que componen el diagrama de despliegue:

- ✓ **Nodo PC Cliente:** se refiere a las computadoras que utilizarán los usuarios para interactuar con la aplicación.
- ✓ **Nodo Servidor de aplicaciones:** representa el servidor Apache donde se encuentra instalado el sistema.
- ✓ **Nodo Servidor de Base de Datos:** representa el servidor PostgreSQL donde se almacena la base de datos del sistema.



Figura 8: Modelo de despliegue

## **Pruebas.**

Las pruebas del software constituyen un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación. Para hablar del aseguramiento de la calidad se debe también hablar de control de calidad, siendo estos los procedimientos que se establecen para poder verificar y medir la aptitud de un producto, servicio o proceso. Una vez generado el código fuente el software debe ser probado para descubrir y corregir el máximo de errores posible antes de su entrega al cliente. (37)

### **Pruebas de caja blanca**

*“La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control de diseño procedimental para obtener los casos de prueba”.* (38)

Existen varios tipos de pruebas de caja blanca entre los cuales se encuentran los siguientes:

- ✓ Prueba del camino básico: permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la

definición de un conjunto básico de caminos de ejecución. Cualquier representación del diseño procedimental se puede traducir a un grafo de flujo.

- ✓ Prueba de condiciones: consiste en el diseño de los casos de prueba donde se ejercitan las condiciones lógicas en el módulo de un programa. Se basa en el criterio de que si un conjunto de pruebas de un programa P es efectivo para detectar errores en las condiciones que se encuentran en P, es probable que el conjunto de pruebas sea también efectivo para detectar otros errores en el programa P.

En el presente trabajo de diploma se decidió emplear la prueba del camino básico, ya que permite comprobar la complejidad ciclomática de cada uno de los métodos del módulo.

Esta prueba resulta ser la más conveniente porque posibilita probar caminos importantes que son de gran relevancia para la operación del software, resulta ser flexible y centrada en las funciones significativas del módulo que se prueba. (38)

Para conocer la complejidad del algoritmo es necesario calcular la complejidad ciclomática del mismo, para realizar el cálculo es necesario primero tener el código o el diseño del algoritmo, luego enmarcar cada instrucción del código con un número, que representa cada lugar del camino que puede seguir la secuencia del algoritmo, a continuación se representa el código con sus instrucciones enmarcadas.

```
public function gestionar_usuario() {  
    1 $permisos = $this->session->userdata('permisos');  
    2 if ($this->contiene("gestionar_usuario", $permisos)) {  
    3 $usuario= $this->session->userdata('id_usuario');  
        $this->insertar_traza($usuario, 1,39);  
        $this->template->add_js('modulos/control_acceso/gestionar_usuario');  
        $data = array();  
        //Cargar listado de usuarios  
        $filas = $this->usuario_md1->obtenerPorPagina(null, null, "", 'ASC');  
        $data['filas'] = $filas;  
        $this->template->set_content('gestionar_usuario', $data);  
        $this->template->build();}  
    4 else{  
        redirect('control_acceso/autenticar_usuario');  
    }  
    5 }
```

Después de este paso, es necesario representar el grafo de flujo asociado, en el cuál se representan distintos componentes como es el caso de:

**Nodo:** son los círculos representados en el grafo de flujo, el cual representa una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

**Aristas:** son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aún cuando el nodo no representa la sentencia de un procedimiento.

**Regiones:** son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalentes a la cantidad de caminos independientes del conjunto básico de un procedimiento.

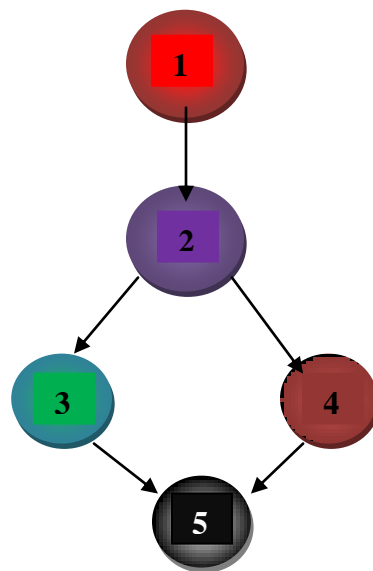


Figura 9: Grafo de flujo asociado a la función `gestionar_usuario()`

Seguidamente a la construcción del grafo de flujo se procede a efectuar el cálculo de la complejidad ciclomática del código, el cálculo es necesario efectuarlo mediante tres vías o fórmulas que, para concluir que el cálculo fue correcto es necesario que por las tres vías el resultado sea el mismo. A continuación se muestran las fórmulas aplicadas para calcular la complejidad en cada uno de los métodos.

Fórmula empleada para calcular la complejidad ciclomática del método:

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

$$V(G) = (5 - 5) + 2$$

$$\mathbf{V(G) = 2}$$

De modo que A es la cantidad total de aristas, siendo N la cantidad total de nodos.

También podemos calcular la complejidad ciclomática de otra forma que a continuación se expone:

$$V(G) = P + 1$$

$$V(G) = 1 + 1$$

$$\mathbf{V(G) = 2}$$

Donde P es la cantidad total de nodos predicados, es decir los nodos de los cuáles parten dos o más aristas.

Se puede utilizar además la siguiente fórmula:

$$V(G) = R$$

$$\mathbf{V(G) = 2}$$

R es la cantidad total de regiones incluyendo la del exterior, para cada fórmula V(G) representa el valor del cálculo.

El valor de las fórmulas anteriormente expuestas representa los posibles caminos por donde se transitará, así como el valor mínimo de casos de pruebas que deben realizarse teniendo en cuenta el procedimiento escogido. Se llega a la conclusión que el algoritmo presentado tiene una complejidad ciclomática de 2 dando visión de que existen a lo sumo dos caminos lógicos por donde recorrer el algoritmo.

En la siguiente tabla se representan los caminos básicos identificados:

Nombre del camino	Secuencia
Camino 1	1-2-3-5
Camino 2	1-2-4-5

Tabla 11: Caminos básicos



A continuación se da paso a la realización de los casos de prueba, teniendo en cuenta que se deben efectuar al menos un caso de prueba por cada camino básico identificado. Para realizar los casos de pruebas es necesario cumplir con las siguientes exigencias:

**Descripción:** Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.

**Condición de ejecución:** Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.

**Entrada:** Se muestran los parámetros que entran al procedimiento.

**Resultados Esperados:** Se expone el resultado que se espera que devuelva el procedimiento.

<b>Caso de prueba para el camino básico # 1</b>	
<b>Nombre de la persona que realiza la prueba:</b> Leidy González Reyes	<b>Estado de evaluación:</b> Satisfactorio
<b>Descripción</b>	En este caso no se hace entrada de datos.
<b>Entradas</b>	contiene ("gestionar_usuario", \$permisos)
<b>Condición de ejecución</b>	El usuario debe tener permiso de acceso a esta funcionalidad.
<b>Resultados esperados</b>	Se muestra un listado con todos los usuarios existentes en el sistema y las opciones para gestionarlos.
<b>Caso de prueba para el camino básico # 2</b>	
<b>Nombre de la persona que realiza la prueba:</b> Leidy González Reyes	<b>Estado de evaluación:</b> Satisfactorio
<b>Descripción</b>	En este caso no se hace entrada de datos.
<b>Entradas</b>	contiene ("gestionar_usuario", \$permisos).
<b>Condición de ejecución</b>	El usuario no tiene permiso de acceso a esta funcionalidad.
<b>Resultados esperados</b>	Devuelve al usuario a la página principal puesto que no tiene permiso de acceso.

--	--

Tabla 12: Casos de prueba de caja blanca.

### Pruebas de caja negra.

Las pruebas de caja negra también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software. Permiten al ingeniero del software obtener un conjunto de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. (37)

En el presente trabajo de diploma se diseñaron varios casos de prueba basados en los casos de uso definidos, cada uno de ellos con sus respectivas secciones y escenarios. Por la extensión de los mismos no fue posible incluirlos todos en el presente documento, pero pueden ser consultados en el expediente de proyecto.

<b>DCP: Caso de prueba: Gestionar usuario</b>		
<b>Gestionar usuario (Adicionar usuario)</b>		
<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El actor introduce todos los datos válidos.	El sistema notifica al actor "Se ha adicionado correctamente".	El actor debe estar autenticado en el sistema.
El actor introduce el usuario incorrecto y todos los demás datos válidos.	El sistema notifica al actor "El usuario no es válido".	
El actor introduce el usuario vacío y todos los demás datos válidos.	El sistema notifica al actor "El usuario no puede estar vacío" (JavaScript) y con el mensaje "Los campos obligatorios no pueden estar vacíos".	
El actor introduce un usuario que	El sistema notifica al actor "Este	

ya existe y todos los demás datos válidos.	usuario ya está registrado en la Base de Datos”.	
El actor introduce el nombre incorrecto y todos los demás datos válidos.	El sistema notifica al actor “El nombre no es válido”.	
El actor introduce el nombre vacío y todos los demás datos válidos.	El sistema notifica al actor “El nombre no puede estar vacío.” (JavaScript) y con el mensaje “Los campos obligatorios no pueden estar vacíos”.	
El actor introduce el carnet de identidad incorrecto y todos los demás datos válidos.	El sistema notifica al actor “El carnet de identidad no es válido”.	
El actor introduce el carnet de identidad vacío y todos los demás datos válidos.	El sistema notifica al actor “El carnet de identidad no puede estar vacío” (JavaScript) y con el mensaje “Los campos obligatorios no pueden estar vacíos”.	
El actor introduce un carnet de identidad que ya existe y todos los demás datos válidos.	El sistema notifica al actor “Este carnet de identidad ya está registrado en la Base de Datos.”	
El actor introduce la clave vacía y todos los demás datos válidos.	El sistema notifica al actor “La clave no puede estar vacía”	

	(JavaScript) y con el mensaje “Los campos obligatorios no pueden estar vacíos”.	
<b>DCP: Caso de prueba: Gestionar usuario</b>		
<b>Gestionar usuario (Modificar usuario)</b>		
<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El actor introduce, no modifica el usuario.	El sistema notifica al actor “Este usuario ya está registrado en la Base de Datos.”	El actor debe estar autenticado en el sistema.
El actor en el usuario que se desea modificar introduce caracteres no válidos.	El sistema notifica al actor “El usuario especificado no puede contener caracteres extraños.”	
El actor modifica el usuario y todos los demás datos válidos.	El sistema notifica al actor “Se ha modificado correctamente. ”	
El actor no introduce el nombre que desea modificar.	El sistema notifica al actor “Debe especificar un nombre”.	
El actor en el nombre introduce caracteres no válidos.	El sistema notifica al actor “El nombre especificado no puede contener caracteres extraños.”	
El actor modifica el nombre y todos los demás datos válidos.	El sistema notifica al actor “Se ha modificado correctamente. ”	
El actor no introduce los apellidos que desea modificar.	El sistema notifica al actor “Debe especificar los apellidos”.	
El actor introduce, no modifica los apellidos.	El sistema notifica al actor “Este usuario ya está registrado en la Base de Datos.”	
El actor en los apellidos introduce	El sistema notifica al actor “Los	

caracteres no válidos.	apellidos especificados no pueden contener caracteres extraños.”	
El actor modifica los apellidos y todos los demás datos válidos.	El sistema notifica al actor “Se ha modificado correctamente. ”	
El actor no introduce el carnet de identidad que desea modificar.	El sistema notifica al actor “Debe especificar el carnet de identidad.”	
El actor introduce, no modifica el carnet de identidad.	El sistema notifica al actor “Este usuario ya está registrado en la Base de Datos.”	
El actor en el carnet de identidad introduce caracteres no válidos.	El sistema notifica al actor “El carnet de identidad especificado no puede contener caracteres extraños.”	
El actor modifica el carnet de identidad y todos los demás datos válidos.	El sistema notifica al actor “Se ha modificado correctamente. ”	
El actor modifica la clave y todos los demás datos válidos.	El sistema notifica al actor “Se ha modificado correctamente. ”	
El actor no introduce la clave que desea modificar.	El sistema notifica al actor “Debe especificar la clave”.	
<b>DCP: Caso de prueba: Gestionar usuario</b>		
<b>Gestionar usuario (Eliminar usuario)</b>		
<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>

El actor selecciona la opción eliminar.	El sistema elimina al usuario, actualiza el listado de usuarios restantes sin la presencia del usuario eliminado y notifica al sistema “Se ha eliminado correctamente”.	El actor debe estar autenticado en el sistema
El actor no selecciona ningún usuario y oprime el botón eliminar todos.	El sistema notifica al actor “Debe seleccionar al menos un usuario.”	
<b>DCP: Caso de prueba: Asignar usuario a rol</b>		
<b>Entrada</b>	<b>Resultado</b>	<b>Condiciones</b>
El actor selecciona la opción asignar usuario a rol.	El sistema muestra al actor la vista asignar usuario a rol.	El actor debe estar autenticado en el sistema.
El actor selecciona el usuario al que desea asignarle dicho rol.	El sistema notifica al actor “Se ha asignado correctamente.”	
El actor no selecciona el usuario que desea asignarle un nuevo rol.	El sistema notifica al actor “Debe seleccionar al menos un usuario.”	
El actor selecciona el usuario que desea eliminar.	El sistema notifica al actor “Se ha eliminado correctamente.”	
El actor no selecciona el usuario que desea eliminar.	El sistema notifica al actor “Debe seleccionar al menos un usuario.”	

Tabla 13: Casos de pruebas caja negra.

## Resultado de las pruebas realizadas

A cada una de las funcionalidades del módulo expuestas con anterioridad se le realizaron pruebas de caja negra, comprobando cada uno de sus campos, y verificando las respuestas que brindaba el sistema para detectar los posibles errores.

Se obtuvieron algunas no conformidades, estas fueron eliminadas mediante el transcurso de cada una de las iteraciones definidas.

A continuación se representan las no conformidades que fueron detectadas y a su vez la solución a cada una de ellas en forma de iteración:

No de la iteración	No conformidad detectada	Defectos encontrados	No conformidad resuelta
1	4	Errores de interfaz y validación.	4
2	2	Errores ortográficos y mensajes erróneos.	2
3	0	0	0

Tabla 14: Resultados de las pruebas.

En la tabla 23 pueden observarse las descripciones de los casos de prueba pertenecientes a 2 de los casos de uso documentados en el trabajo, estos fueron probados para comprobar las funcionalidades del módulo; las pruebas realizadas se llevaron a cabo en tres iteraciones, en la primera iteración se detectaron 4 no conformidades, todas fueron resueltas; en la segunda iteración se detectaron 2 no conformidades que fueron resueltas inmediatamente también y en la última iteración ya no se encontraron no conformidades, por lo tanto se puede concluir que las pruebas de funcionalidad en el módulo fueron realizadas satisfactoriamente quedando el sistema listo para su funcionamiento.

## ***Conclusiones***

Luego del proceso de implementación y prueba del módulo se concluyó que el diagrama de componentes permite garantizar las vistas, los modelos y la clase controladora que conforma el módulo, y con la realización de las pruebas de caja blanca y caja negra se cumple con las especificaciones que se trazaron ya que cada uno de los requisitos definidos en el transcurso de la investigación fueron desarrollados.



## Conclusiones generales

Durante el transcurso de esta investigación se desarrolló un módulo para gestionar el control de acceso del Sistema Dexcriba, por lo que se puede arribar a las siguientes conclusiones:

- ✓ Ninguno de los sistemas de gestión de documentos de archivos históricos estudiados da respuesta al problema planteado, puesto que en algunos casos la tecnología que se utiliza es de punta y en otros simplemente no existe el control de acceso con las características que debe cumplir el que se desea implementar. Por esto se decide desarrollar una nueva solución utilizando el modelo de control de acceso basado en roles, siendo más fácil gestionar el acceso al sistema asignando roles a los usuarios, y utilizar además el proyecto OWASP como material de referencia en el diseño, desarrollo, despliegue y verificación de la seguridad.
- ✓ El uso de la metodología RUP posibilita el desarrollo de manera eficiente de un sistema de software pues permite que los desarrolladores puedan construir herramientas que soporten la automatización del proceso entero, de cada flujo de trabajo individualmente y de la integración del trabajo a lo largo del ciclo de vida y a través de todos los modelos.
- ✓ El uso del lenguaje de modelado UML facilitó la modelación de los artefactos de los flujos de trabajo de la metodología utilizada.
- ✓ Con la implementación del módulo, se establece la seguridad de los documentos existentes en el Sistema de Gestión de Documentos Históricos Dexcriba.
- ✓ Las pruebas realizadas posibilitaron el correcto funcionamiento de la aplicación permitiendo un fácil manejo por parte de los usuarios, ya que cada uno de los requisitos definidos en el transcurso de la investigación fueron desarrollados.

## Bibliografía

1. **Mena Mugica, Mayra.** Gestión Documental y organización de archivos. Ciudad de la Habana, Cuba : Félix Varela, 2005.
2. **Heredia Herrera, Antonia.** Archivística General. Teoría y práctica. SEVILLA: s.n., 1991. I.S.B.N. 84 - 7798 - 056 – X.
3. **Cruz Mundet, José Ramón.** Manual de Archivística. 2da edición. Madrid: s.n., 1996. ISBN 8.1-86168-94-5(F.G.S.R.), ISBN 83-368-0860-6 (Pirámide).
4. **Chavez González, M.** Ley de 9 de enero de 1984, Archivos de Andalucía. Andalucía: Boletín Oficial de la Junta de Andalucía.
5. **Diccionario de Terminología archivística.** [Consultado el: 5 de noviembre de 2012]. Disponible en: <<http://www.mcu.es/archivos/MC/DTA/Diccionario.html>>
6. **Consejo Internacional de Archivos.** ISAD (G) Norma Internacional General de descripción archivística, Segunda Edición, Madrid, 2000.
7. **Control de acceso en sistemas informáticos.** [Consultado el: 10 de enero 2013]. Disponible en:<<http://www.subinet.es/guias-y-tips/guias-tips-internet/%C2%BFque-es-el-control-de-acceso-en-sistemas-informaticos>>
8. **Auditoria.** [Consultado el: 10 de enero 2013]. Disponible en: <<http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/SO14.htm#Aud> >
9. **PCMAG Encyclopedia.** [Consultado el: 10 de enero 2013]. Disponible en:<[http://www.pcmag.com/encyclopedia\\_term/0,2542,t=Web+application&i=54272,00.asp](http://www.pcmag.com/encyclopedia_term/0,2542,t=Web+application&i=54272,00.asp) >
10. **OWASP.** Una guía para construir aplicaciones y servicios web seguros. [Consultado el: 11 de enero 2013]. Disponible en: <[https://www.owasp.org/images/b/b2/OWASP\\_Development\\_Guide\\_2.0.1\\_Spanish.pdf](https://www.owasp.org/images/b/b2/OWASP_Development_Guide_2.0.1_Spanish.pdf)>
11. **Navarro, María García González y Celia Chain.** [Consultado el: 11 de enero de 2013]. Disponible en:

- < [https://www.scielo.org.mx/scielo.php?pid=S0187-358X2010000200003&script=sci\\_arttext](https://www.scielo.org.mx/scielo.php?pid=S0187-358X2010000200003&script=sci_arttext)>
12. **ICA-Atom**. The ICA-Atom Project and Technology. [Consultado el: 12 de enero 2013].  
Disponible en:  
<[https://www.icaatom.org/download/VanGarderen\\_TheICAAtomProjectAndTechnology\\_AAB\\_RioDeJaniero\\_16-17March2009.pdf](https://www.icaatom.org/download/VanGarderen_TheICAAtomProjectAndTechnology_AAB_RioDeJaniero_16-17March2009.pdf)>
13. **Sistedes**. Sociedad de Ingeniería de Software y Tecnología de Desarrollo de Software.  
[Consultado el: 12 de enero 2013]. Disponible en: < <https://www.sistedes.es/TJISBD/Vol-1/No-1/.../SCHA-07-Sanchez-Acceso.pdf>>
14. **López Millán Gabriel**. Definición de una infraestructura de control de acceso basada en la arquitectura AAA y el uso de credenciales de autorización. Tesis Doctoral. Departamento de Ingeniería de la Información y las Comunicaciones. Universidad de Murcia. España. 2006. Disponible en:  
<<http://ants.dif.um.es/staff/gabilm/tesis/G.Lopez.PhD.pdf> >
15. **Lenguaje de programación**. [Consultado el: 12 de enero 2013]. Disponible en:<[http://es.wikipedia.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) >
16. **Manual de PHP. Qué es PHP**. [Consultado el: 12 de enero de 2013]. Disponible en:<[http://www.forosdelweb.com/wiki/Manual\\_de\\_PHP:\\_Qu%C3%A9\\_es\\_PHP](http://www.forosdelweb.com/wiki/Manual_de_PHP:_Qu%C3%A9_es_PHP)>
17. **Introducción a XHTML**. [Consultado el: 12 de enero de 2013]. Disponible en:  
< <http://www.librosweb.es/xhtml>>
18. **JavaScript**. [Consultado el: 12 de enero 2013]. Disponible en:  
<<http://es.wikipedia.org/wiki/JavaScript> >
19. **XML**. [Consultado el: 12 de enero 2013]. Disponible en:<<http://es.wikipedia.org/wiki/XML> >
20. **Fundación CTIC**. Hojas de estilos. [Consultado el: 12 de enero 2013]. Disponible en:  
<<http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo> >
21. **Introducción a CSS**. [Consultado el: 12 de enero 2013]. Disponible en:  
< <http://www.librosweb.es/css>>

22. **Gutiérrez, Javier J.** Qué es un framework web? [Consultado el: 12 de enero 2013].  
Disponible en: < [http://www.lsi.us.es/~javierj/investigacion\\_ficheros/Framework.pdf](http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf) >
23. **Murphey, Rebecca.** jQuery Fundamentals. [Consultado el: 12 de enero de 2013]  
Disponible en:< <http://www.etnassoft.com/biblioteca/jquery-fundamentals> >
24. **Primeros pasos a jquery UI.** [Consultado el: 12 de enero 2013]. Disponible en:  
<<http://www.desarrolloweb.com/articulos/primeros-paso-jquery-ui.html>>
25. **Álvarez, Miguel Ángel.** Manual de CodeIgniter. [Consultado el: 12 de enero 2013].  
Disponible en: <<http://sunshine.prod.uci.cu/gridfs/sunshine/books/manual-codeigniter.pdf>>
26. **PostgreSQL vs. MySQL.** [Consultado el: 15 de enero de 2013]. Disponible en:  
< [http://danielpecos.com/docs/mysql\\_postgres/x15.html](http://danielpecos.com/docs/mysql_postgres/x15.html) >
27. **DESARROLLO WEB.** [Consultado el: 15 de enero de 2013]. Disponible en:  
< <http://www.desarrolloweb.com/articulos/182.php> >
28. **INFORMÁTICA.** Historia de Netbeans IDE. [Consultado el: 20 de enero de 2013]. Disponible en: < <http://jany16.wordpress.com/2007/08/04/historia-de-netbeans-ide> >
29. **Bienvenido a Netbeans y a www.netbeans.org.** Portal del IDE Java Open Source.  
[Consultado el: 20 de enero de 2013]. Disponible en:  
< [http://netbeans.org/community/articles/welcome-template\\_es.html](http://netbeans.org/community/articles/welcome-template_es.html) >
30. **Booch, G., Rumbaugh, J. y Jacobson, I.** 2004. El Proceso Unificado de Desarrollo de Software. La Habana: Félix Varela, 2004.
31. **Menéndez- Barzanallana Asensio, R.** Informática Aplicada a la Gestión Pública. Facultad Derecho UMU. España: Universidad de Murcia, Última actualización: 2011/10/13. [Consultado el: 24 de enero de 2012]. Disponible en:  
< <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html> >
32. **Pressman, Roger S. 2002.** Ingeniería del software. Un enfoque práctico. Quinta Edición.s.l: Editorial McGraw-Hill, 2002. pág. 640. ISBN: 8448132149.
33. **Pressman, Roger S.** Ingeniería de Software. Un enfoque práctico. Madrid y Carachelejo (España): s.n., 2001.

34. **Mato García, R. M. Diseño de Bases de datos.** [Consultado el: 24 de enero de 2013].  
Disponible en:  
< [http://eva.uci.cu/mod/resource/view.php?id=11576&subdir=/Bibliografia\\_Basica](http://eva.uci.cu/mod/resource/view.php?id=11576&subdir=/Bibliografia_Basica) >
35. **Parra, José David. Guía de Patrones, Prácticas y Arquitectura .NET.** Disponible en:  
< <http://sunshine.prod.uci.cu/gridfs/sunshine/books/PPArquitecturaNET.pdf> >
36. **Quisbert Limachi, Nancy Susana y Marca Huallpara, Hugo Michael.** Diagrama de Despliegue. [ Consultado el: 1 de abril de 2013]. Disponible en:  
< <http://virtual.usalesiana.edu.bo/web/practica/archiv/despliegue.doc> >
37. **Pressman, Roger S. 2005.** Ingeniería del Software. Un enfoque práctico. La Habana : Félix Varela, 2005. ISBN: 970-105-473-3.
38. **Watson, A.H. and T.J. McCabe1996.** Structured testing: a testing methodology using the cyclomatic complexity metric. [Consultado el: 5 de mayo de 2013]. Disponible en:  
<<http://www.uv.mx/personal/jfernandez/files/2010/07/Cap3-Caminos.pdf> Technical Report NIST 500-225 >

## Anexos

### Anexo 1

#### CU Autenticar usuario

<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el actor procede a autenticarse en el sistema y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF1

Tabla 15: Descripción detallada de CU: Autenticar usuario.

### Anexo 2

#### CU Gestionar usuario

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a Gestionar usuario, puede adicionar, listar, modificar y eliminar usuarios y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF2,RF3,RF4,RF5,RF24

Tabla 16: Descripción detallada de CU: Gestionar usuario.

### Anexo 3

#### CU Buscar usuario

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a buscar los datos de un usuario determinado y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF6

Tabla 17: Descripción detallada de CU: Buscar usuario.

#### **Anexo 4**

### **CU Buscar rol**

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a buscar los datos de un rol determinado y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF11

*Tabla 18: Descripción detallada de CU: Buscar rol.*

#### **Anexo 5**

### **CU Asignar usuario a rol**

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a asignar un usuario a determinado rol y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF11

*Tabla 19: Descripción detallada de CU: Asignar rol a usuario.*

#### **Anexo 6**

### **CU Gestionar traza**

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a Gestionar traza, puede listar y eliminar trazas y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF13,RF14,RF15,RF24

*Tabla 20: Descripción detallada de CU: Gestionar traza.*

#### **Anexo 7**

### **CU Búsqueda traza sencilla**

<b>Actores</b>	Usuario
----------------	---------

<b>Resumen</b>	El caso de uso inicia cuando el actor se autentica en el sistema y procede a realizar una búsqueda general sobre las acciones que ha realizado en el sistema y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF16

Tabla 21: Descripción detallada de CU: Búsqueda traza sencilla.

## Anexo 8 CU Búsqueda traza avanzada

<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el actor se autentica en el sistema y procede a realizar una búsqueda avanzada sobre las acciones que ha realizado en el sistema teniendo en cuenta un rango de fecha determinado, y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF17

Tabla 22: Descripción detallada de CU: Búsqueda traza avanzada.

## Anexo 9 CU Gestionar funcionalidad

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a Gestionar funcionalidad, puede adicionar, modificar, listar, y eliminar funcionalidades y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF19,RF20,RF21,RF22,RF24

Tabla 23: Descripción detallada de CU: Gestionar funcionalidad.



## Anexo 10

### CU Buscar funcionalidad

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a buscar los datos de una funcionalidad determinada y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF23

Tabla 24: Descripción detallada de CU: Buscar funcionalidad.

## Anexo 11

### CU Registrar usuario

<b>Actores</b>	Usuario
<b>Resumen</b>	El caso de uso inicia cuando el usuario intenta autenticarse en el sistema pero sus datos no se encuentran en la base de datos y por ello se hace necesario que se registre una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF24

Tabla 25: Descripción detallada de CU: Registrar usuario.

## Anexo 12

### CU Asignar nivel de organización a rol

<b>Actores</b>	Administrador
<b>Resumen</b>	El caso de uso inicia cuando el administrador se autentica en el sistema, y procede a asignar un nivel de organización a determinado rol y una vez ejecutada la acción por parte del sistema, finaliza el caso de uso.
<b>Referencias</b>	RF25

Tabla 26: Asignar nivel de organización a rol.

### Anexo 13

<b>Tabla</b>	sexo	
<b>Descripción</b>	Almacena el sexo del usuario( Femenino(1) o Masculino(2))	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
id_sexo	Int(4)	Identifica las tuplas en la tabla, valor generado auto incrementalmente por el gestor
sexo	Int(4)	1 ó 2

Tabla 27: Modelos de datos entidad: sexo

### Anexo 14

<b>Tabla</b>	usuario_rol	
<b>Descripción</b>	Surge de la relación entre las tablas usuario y rol	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
usuarioid_usuario	Int(4)	Identifica a la tabla que pertenece( llave foránea)
rolid_rol	Int(4)	Identifica a la tabla que pertenece( llave foránea)

Tabla 28: Modelo de datos entidad: usuario\_rol

### Anexo 15

<b>Tabla</b>	funcionalidad_rol	
<b>Descripción</b>	Surge de la relación entre las tablas funcionalidad y rol	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
rolid_rol	Int(4)	Identifica a la tabla que pertenece( llave foránea)
funcionalidadid_func	Int(4)	Identifica a la tabla que pertenece( llave foránea)

Tabla 29: Modelo de datos entidad: funcionalidad\_rol

## Anexo 16

<b>Tabla</b>	rol_nivel_organización	
<b>Descripción</b>	Surge de la relación entre las tablas nivel_organización y rol	
<b>Atributos</b>	<b>Tipo</b>	<b>Descripción</b>
rolid_rol	Int(4)	Identifica a la tabla que pertenece (llave foránea)
nivel_organizaciónid_nivel	Int(4)	Identifica a la tabla que pertenece (llave foránea)

Tabla 30: Modelo de datos entidad: rol\_nivel\_organización.