



**Universidad de las Ciencias Informáticas
Facultad 2**

*Trabajo de diploma para optar el título de Ingeniero en Ciencias
Informáticas*

*“Módulo de Gestión de Nomencladores para el Gestor de Documentos
Administrativos XABAL eXcriba 3.1”*

Autor:

Alejandro Jesús Pírez Prendes

Tutores:

MSc. Jenísley Verde Acosta

Ing. Lisandra Olivares Labarcena

Ing. Yaniel Blanco Fernández

La Habana, junio de 2018



Se puede adquirir conocimientos y conciencia a lo largo de toda la vida, pero jamás en ninguna otra época de su existencia una persona volverá a tener la pureza y el desinterés con que, siendo joven, se enfrenta a la vida.

Fidel Castro Ruz

Declaración de autoría

Se declara que Alejandro Jesús Pirez Prendes es el único autor de este trabajo, que tiene por título: Módulo de gestión de nomencladores para el Gestor de Documentos Administrativos XABAL eXcriba 3.1 y se autoriza a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ___ días del mes de _____ del año 2018.

Firma del autor.

Alejandro Jesús Pirez Prendes

Firma del tutor

MSc. Jenisley Verde Acosta

Firma del tutor

Ing. Yaniel Blanco Fernández

Firma del tutor

Ing. Lisandra Olivares Labarcena

Dedicatoria

- A mis abuelos en especial a mi abuelita del alma Eloisa la cual su sueño fue verme convertido en universitario, y la vida no le permitió verlo con sus propios ojos, a ti va esta tesis todo cuanto hay de bueno en mi te lo debo a ti.
- A mi hermanita Yousi, por enseñarme a luchar, incluso en los momentos más difíciles, espero que desde el cielo me sigas cuidando, tu eres mi inspiración y mi ejemplo a seguir en esta vida.
- A mis padres por ser el motor impulsor durante estos difíciles cinco años.
- A mis tios por darme su apoyo y sus buenos consejos.
- A la Universidad de las Ciencias Informáticas por haber cumplido mi sueño de ser informático.

Agradecimientos

Agradecimientos

A mis tutores

- A Yaniel por ser más que un tutor un amigo y apoyarme incluso cuando más complicado estaba.
- A Lisandra porque es la tutora más exigente y más buena que un ser humano pueda tener, y porque cada vez que lo necesité, me extendió sus brazos y me enseñó a dar pasos firmes.
- A Jenisley por brindarme su apoyo, darme buenos consejos y apoyarme desde que me asumió como tesista como un hijo más.

A mis padres

- A mi madre por sus enseñanzas y su apoyo incondicional.
- A mi padre por apoyarme en todo y por estar siempre en los momentos difíciles.

A los profesores que me han ayudado en mi formación como profesional en el transcurso de la carrera.

A mis amigos

- En especial a Sergio por ser mi hermanito.
- A Yoanis, Yanlee, Osniel, Osciél, Henry, Yaidel, Felipe, Alianhs, Lachi y a todos los que de una forma u otra estuvieron conmigo durante estos años.
- A mi amigo Conrado por sus consejos y por ser un padre más para mí.

Resumen

El Gestor de Documentos Administrativos Xabal eXcriba en su versión 3.1, es un software cubano diseñado para la gestión documental, que permite el trámite de los documentos que se generan o reciben. En la actualidad las empresas y organizaciones cubanas se encuentran en un estado de perfeccionamiento, lo que trae consigo que sus procesos se encuentren en un constante cambio hacia el desarrollo, por lo que la información declarada como estática o poco variable en el tiempo se esté modificando. Esto obstaculiza el trabajo pues la aplicación no está diseñada para tratar esta problemática, lo que trae como consecuencia que tengan que reimplementarse funcionalidades que ya se habían desarrollado, para agregar informaciones relevantes no contempladas al principio. El presente trabajo describe el desarrollo de un módulo de Gestión de Nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1, que permita el manejo de la información poco variable en el tiempo de las diferentes empresas y organizaciones. En el documento se describen los conceptos fundamentales utilizados, además de realizarse un estudio minucioso sobre sistemas similares en el ámbito nacional e internacional. Durante el desarrollo del módulo se utilizaron como lenguajes de desarrollo JavaScript 1.6, XML 1.0, UML y HTML5 y como herramientas de desarrollo el Visual Paradigm para UML 8.0, Apache Tomcat 7.0 y el editor de texto visual code. La implementación del módulo permite la gestión de la información poco variable en el tiempo, dándole funcionalidades al eXcriba que contribuyen con su despliegue y uso en empresas e instituciones.

Palabras clave: GDA XABAL eXcriba, Gestión de nomencladores, información poco variable en el tiempo, nomenclador.

Abstract

The Manager of Administrative Documents Xabal eXcriba in version 3.1, is a Cuban software designed for document management, which allows the processing of documents that are generated or received. Currently, Cuban companies and organizations are in a state of improvement, which means that their processes are in constant change towards development, so that information declared as static or little variable over time is being modified. This hinders the work because the application is not designed to address this problem, which results in having to reimplement functionalities that had already been developed, to add relevant information not contemplated at the beginning. The present work describes the development of a Module of Management of Nomenclators in the Manager of Administrative Documents Xabal eXcriba 3.1, that allows the handling of the information little variable in the time of the different companies and organizations. The document describes the fundamental concepts used, in addition to a thorough study of similar systems at national and international level. During the development of the module were used as development languages JavaScript 1.6, XML 1.0, UML and HTML5 and as development tools the Visual Paradigm for UML 8.0, Apache Tomcat 7.0 and the text editor visual code. The implementation of the module allows the information management little variable over time, giving functionality to the eXcriba that contribute to its deployment and use in companies and institutions.

Keywords: GDA XABAL eXcriba, information little variable in time, Management of nomenclators, nomenclator.

Índice

Introducción	1
Capítulo I. Fundamentación teórica	6
1.1. Conceptos asociados a la investigación.....	6
1.2. Sistemas similares.....	6
1.2.1. Características de algunos sistemas internacionales.	7
1.2.2. Características de algunos sistemas nacionales.	8
1.2.3. Resultados del estudio	9
1.3. Metodología de desarrollo de software	10
1.3.1. Proceso Unificado Ágil para el Desarrollo de la actividad productiva en la Universidad de las Ciencias Informáticas	10
1.3.2. Escenarios para la disciplina requisitos que presenta AUP-UCI.....	11
1.4. Plataforma de desarrollo.....	12
1.4.1. ECM Alfresco Community 4.2.f.....	12
1.4.2. Gestor de Documentos Administrativos (GDA) XABAL eXcriba 3.1	13
1.4.3. Lenguajes de desarrollo	13
1.4.4. Tecnologías de desarrollo	14
1.5. Sistema gestor de base de datos.....	16
1.5.1. PostgreSQL v9.4.2	16
1.6. Lenguaje de modelado UML.....	16
1.6.1. Herramienta CASE: Visual Paradigm para UML 8.0.....	17
1.7. Axure RP PRO v7.0: Diseño de prototipos.....	17
Conclusiones parciales.....	17
Capítulo II. Propuesta de solución	18
2.1. Modelo conceptual.....	18
2.1.1. Diccionario de datos.....	19
2.2. Requisitos funcionales	22
2.3. Requisitos no funcionales	23
2.4. Descripción de la propuesta de solución.....	24
2.5. Técnicas de validación de requisitos.....	24
2.6. Definición de los casos de usos del sistema	24
2.6.1. Definición de los actores	24
2.6.2. Diagrama de caso de uso del sistema.....	25

2.6.3.	Descripción de casos de uso del sistema.....	25
2.7.	Técnica de trazabilidad de requisitos	30
2.8.	Descripción de la arquitectura.....	31
2.8.1.	Arquitectura en capas	32
2.9.	Modelo de despliegue.....	33
2.9.1.	Diagrama de despliegue.....	34
2.10.	Patrones de diseño	35
2.10.1.	Patrones GRASP.....	35
2.10.2.	Patrones GOF	36
	Conclusiones parciales.....	37
Capítulo III.	Implementación y prueba.....	38
3.1.	Modelo de contenido.....	38
3.2.	Webscript.....	38
3.3.	Despliegue de la solución	41
3.4.	Interfaz gráfica del módulo.....	42
3.5.	Pruebas de software	43
3.5.1.	Pruebas internas	43
3.5.2.	Pruebas de aceptación.....	50
	Conclusiones parciales.....	51
	Conclusiones generales.....	52
	Recomendaciones	53
	Referencias Bibliográficas.....	54
	Bibliografía.....	58
	Anexos.....	62
	Anexo 1. Entrevista con el cliente.....	62
	Anexo 2. Acta de aceptación de requisitos	63
	Anexo 3. Casos de uso del sistema.....	63
	Anexo 4. Matrices de trazabilidad.....	65
	Anexo 5. Casos de prueba	67
	Anexo 6. Acta de evaluación interna de producto de software.....	68
	Anexo 7. Acta de aceptación del módulo.....	69

Índice de figuras

<i>Fig 1: Escenario 2 Metodología AUP-UCI (Rodríguez Sánchez, 2015)</i>	11
<i>Fig 2: Modelo conceptual</i>	18
<i>Fig 3: Diagrama de casos de uso</i>	25
<i>Fig 4: Descripción de la arquitectura</i>	33
<i>Fig 5: Diagrama de despliegue</i>	34
<i>Fig 6: Listado de webscript de acceso al repositorio</i>	39
<i>Fig 7: list-nodes.get.desc.xml</i>	40
<i>Fig 8: list-nodes.get.js</i>	40
<i>Fig 9: list-nodes.get.json.ftl</i>	41
<i>Fig 10: list-nodes.get.json.ftl</i>	42
<i>Fig 11: Interfaz gráfica del módulo</i>	43
<i>Fig 12: Método getAssocs(node) del webscript view-node.get.js</i>	45
<i>Fig 13: Técnica del camino básico. Elaboración propia</i>	45
<i>Fig 14: Entrevista con el cliente</i>	62
<i>Fig 15: Acta de aceptación de requisitos de software</i>	63
<i>Fig 16: Acta de evaluación interna de producto de software</i>	68
<i>Fig 17: Acta de aceptación del módulo</i>	69

Índice de tablas

<i>Tabla 1: Descripción del concepto “Nomenclador”</i>	20
<i>Tabla 2: Descripción del concepto “Valor”</i>	21
<i>Tabla 3: Definición de los actores.</i>	24
<i>Tabla 4: Descripción del caso de uso: Gestionar nomenclador.</i>	25
<i>Tabla 5: Matriz de trazabilidad de RF-CU.</i>	30
<i>Tabla 6: Diseño de caso de prueba para el camino 1.</i>	46
<i>Tabla 7: Caso de prueba: Gestionar nomenclador SC1 Crear nomenclador.</i>	48
<i>Tabla 8: Resultados de las pruebas internas ejecutadas por el desarrollador.</i>	49
<i>Tabla 9: Resultados de las pruebas internas a nivel de centro CIGED.</i>	50
<i>Tabla 10: Descripción del caso de uso: Listar nomenclador.</i>	63
<i>Tabla 11: Descripción del caso de uso: Buscar nomenclador</i>	64
<i>Tabla 12: Matriz de trazabilidad de RF-Concepto.</i>	65
<i>Tabla 13: Matriz de trazabilidad de RF-RF.</i>	66
<i>Tabla 14: Matriz de trazabilidad de CU-CP.</i>	66
<i>Tabla 15: Caso de Prueba Gestionar nomenclador SC2 Editar nomenclador.</i>	67
<i>Tabla 16: Caso de Prueba Gestionar nomenclador SC3 Ver detalles del nomenclador.</i>	67

Introducción

La información es un fenómeno que proporciona significado o sentido a las cosas, es un conjunto organizado de datos procesados. Es un elemento clave y fundamental para el buen funcionamiento de cualquier tipo de organización (Hernández, 2015).

Una propiedad importante de la información es, ser almacenada. Su conservación a través del paso del tiempo es una de las mejores características que posee, más aún, cuando esta acción se realiza de forma ordenada y sistemática, de manera tal que pueda ser localizada rápidamente. Su organización ha sido una necesidad del hombre que ha trascendido con el paso de los años, llegando a ser el proceso que estructura y da vida a una biblioteca (Restrepo-Arango, 2016).

En la actualidad la información es un valor agregado en las empresas, pero se evidencia que muy pocas organizaciones implementan los mecanismos que garanticen que esta responda eficientemente a las demandas del mercado. Sin embargo, con el avance vertiginoso de las tecnologías de la información y las comunicaciones (TIC) ha ocurrido una migración en el funcionamiento interno de las empresas y organizaciones actuales hacia sistemas electrónicos y digitales. Estos sistemas son aplicaciones informáticas modernas que permiten compartir los recursos de la información, acelerar los procesos de comunicación y disponer de una adecuada organización de la misma (Mosquera, y otros, 2017).

La información organizada debe estar presente como un elemento básico en el diseño de las estrategias de la innovación tecnológica y en las acciones que de ella se deriven. La utilización de las tecnologías ha permitido compartir los recursos, quebrantar las fronteras geográficas y acelerar los procesos de comunicación, fomentando un nuevo paradigma (Delgado, 2006).

Cuba no queda aislada de la necesidad de utilizar las TIC para el proceso de organización de la información. Este a pesar de ser un país subdesarrollado, ha identificado desde muy temprano la necesidad de dominar e introducir en la práctica social las TIC, permitiéndole acercarse hacia el objetivo de un desarrollo justo, equitativo, sostenible y alcanzable. Por tal motivo, se ha extendido el uso de las TIC a los diferentes sectores de la sociedad.

La Universidad de las Ciencias Informáticas (UCI) ha sido uno de los sectores beneficiados con la aplicación de las TIC. En esta institución se encuentra el Centro de Informatización de la Gestión Documental (CIGED), dedicado al desarrollo de sistemas y servicios informáticos integrales de alta calidad y competitividad en la informatización de los procesos de gestión documental, cuenta con el producto Gestor de Documentos

Administrativos Xabal eXcriba en su versión 3.1 (GDA XABAL eXcriba 3.1), este es un software diseñado para la gestión documental, que permite el trámite de los documentos administrativos que se generan o reciben dentro de las organizaciones a partir de sus funciones, por lo tanto, involucra todas las áreas de una organización, permitiéndoles gestionar de forma correcta la documentación como prueba, testimonio y evidencia de las actividades organizacionales. La solución de software agiliza el trámite de los documentos, permitiendo controlar el estado de los mismos y evitando la pérdida de información.

El GDA XABAL eXcriba 3.1, permite trabajar en un entorno de colaboración entre los usuarios de la organización, compartiendo los documentos con otros usuarios y estableciendo niveles de acceso y permiso. Además, optimiza la organización de los documentos, posibilita la búsqueda de información y gestiona los documentos durante su ciclo de vida. Las interfaces de usuario de esta aplicación están orientadas a personas que no tienen que poseer un alto grado de conocimiento de informática para trabajar con el sistema. Es una aplicación que se personaliza¹ para cada empresa u organización teniendo en cuenta sus procesos y sistema de trabajo.

En la actualidad el desarrollo de dichas personalizaciones se dificulta, puesto que las empresas y organizaciones cubanas están en un estado de perfeccionamiento, lo que trae consigo, que sus procesos se encuentren en un constante cambio hacia el desarrollo, por lo que la información declarada como estática o poco variable en el tiempo se esté modificando constantemente. Esto obstaculiza el trabajo, pues el GDA eXcriba no está diseñado para tratar esta problemática, lo que trae como consecuencia:

- Reimplementación de funcionalidades que ya se habían desarrollado, para agregar informaciones relevantes que no se habían contemplado al principio.
- Empleo de personal calificado del proyecto para realizar esta tarea, ya que es necesario acceder al código para efectuar las modificaciones.
- Empleo de personal del proyecto para la realización del soporte a la personalización del GDA desarrollada para una empresa determinada.

Teniendo presente la problemática planteada anteriormente, se propone como **problema a resolver** Cómo contribuir a la gestión de la información poco variable en el tiempo de las diferentes empresas y organizaciones con el GDA XABAL eXcriba 3.1.

Como **objeto de estudio** de esta investigación se tiene: El proceso de gestión de la información poco variable en el tiempo para sistemas de gestión documental.

¹ Individualizar, singularizar o dar carácter personal a algo (Diccionario, 2017).

Para dar solución al problema planteado se propone como **objetivo general**: Desarrollar un módulo de Gestión de Nomencladores en el GDA XABAL eXcriba 3.1, que permita el manejo de la información poco variable en el tiempo de las diferentes empresas y organizaciones.

Enmarcando esta investigación en el **campo de acción**: El proceso de gestión de la información poco variable en el tiempo para el GDA Xabal eXcriba 3.1.

Las **preguntas científicas** que guían y orientan el desarrollo del proceso investigativo son las siguientes:

- ¿Cómo desarrollar un módulo de Gestión de Nomencladores en el GDA XABAL eXcriba 3.1, que permita el manejo de las diferentes empresas y organizaciones?
- ¿Cuáles son los referentes teóricos a tener en cuenta para abordar la solución del problema planteado relacionado con la gestión de la información poco variable en el tiempo en el GDA XABAL eXcriba 3.1?
- ¿Qué propuesta de solución se define para implantar en el desarrollado para el GDA XABAL eXcriba 3.1?
- ¿Cómo se valida el correcto funcionamiento del módulo de Gestión de Nomencladores en el GDA XABAL eXcriba 3.1?

Para dar solución al problema planteado y dar respuesta a las preguntas científicas formuladas se proponen las siguientes **tareas de investigación**:

- Análisis de los principales conceptos asociados a la Gestión de Nomencladores para el GDA eXcriba 3.1 para obtener los fundamentos teóricos necesarios para el desarrollo de la solución.
- Análisis de los sistemas homólogos para identificar tendencias actuales relacionadas con las funcionalidades y mecanismos utilizados.
- Análisis de la metodología de software, herramientas informáticas y tecnologías para su posterior utilización en el desarrollo del módulo.
- Definición de los requisitos funcionales y no funcionales para la implementación del módulo gestión de nomencladores del GDA eXcriba 3.1.
- Análisis de los métodos y técnicas para la realización de pruebas de software.
- Validación del módulo de gestión de nomencladores en el GDA eXcriba 3.1 para comprobar su correcto funcionamiento.

Para el desarrollo de esta investigación se utilizan diferentes métodos científicos, estos son:

Métodos de la investigación:

Teóricos:

- **Analítico-Sintético:** Para analizar y extraer de la documentación especializada los aspectos referentes a la Gestión de los nomencladores.
- **Histórico-Lógico:** Para realizar un estudio cronológico y explícito de las tendencias actuales sobre los sistemas informáticos que intervienen en la Gestión de los nomencladores.
- **Modelado:** Se utiliza en la modelación de los diagramas correspondientes a la etapa de análisis, diseño e implementación del módulo a desarrollar en la presente investigación.

Métodos empíricos:

- **Observación:** Para comprender la forma en que se realiza, de manera manual, el proceso de la Gestión de los nomencladores.
- **Entrevista:** Se empleó esta técnica para el levantamiento de los requisitos con el cliente, ver Anexo 1 Entrevista con el cliente.

El trabajo de diploma presenta como **aporte práctico** que permite la gestión de la información, poco variable en el tiempo dándole funcionalidades al GDA eXcriba 3.1 que contribuyen con su uso en empresas e instituciones.

El presente documento de tesis está compuesto por: Resumen, Introducción, tres Capítulos, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía, Anexos y Glosario de términos, donde se abarca todo lo relacionado con la investigación realizada.

Capítulo 1: Fundamentación Teórica, se incluye el estado del arte del tema tratado tanto a nivel internacional como nacional. Se describen los conceptos fundamentales asociados al dominio del problema y se hace un análisis entre las soluciones existentes y la propuesta que se realiza. Además, se analizan las herramientas, tecnologías y metodologías a usar en la solución del problema planteado.

Capítulo 2: Propuesta de solución, se realiza una breve descripción del Modelo conceptual, así como de sus elementos más importantes. Se especifican los requerimientos funcionales y no funcionales, a partir de los cuales se representan los casos de uso del sistema y la descripción de los mismos. Además, se expone la arquitectura, se establecen los patrones de diseño y la distribución física del sistema mediante el diagrama de despliegue.

Capítulo 3. Implementación y prueba, en este capítulo se presenta el modelo de contenido desarrollado, se detallan los webscript y se plantea la interfaz gráfica del módulo. Además se trata todo lo relacionado

Introducción

con las pruebas realizadas al módulo desarrollado, en pos de verificar la calidad y efectividad de acuerdo con las necesidades del cliente.

Capítulo I. *Fundamentación teórica*

Introducción

En el presente capítulo se abordan primeramente conceptos necesarios para el entendimiento y desarrollo de esta investigación, se realiza un estudio detallado de sistemas existentes para la gestión de nomencladores a nivel nacional e internacional. También se exponen las principales características de la metodología, herramientas, lenguajes y tecnologías informáticas que se emplearán en el desarrollo de la solución.

1.1. **Conceptos asociados a la investigación**

Gestión: término utilizado para referirse al conjunto de acciones, o diligencias que permiten la realización de cualquier actividad o deseo. Dicho de otra manera, una gestión se refiere a todos aquellos trámites que se realizan con la finalidad de resolver una situación o materializar un proyecto (Porto, et al., 2008).

Importante es subrayar que la gestión, que tiene como objetivo primordial el conseguir aumentar los resultados óptimos de una empresa, depende fundamentalmente de cuatro pilares básicos, estos son (Gestión, 2017):

- Estrategia: conjunto de líneas y de trazados de los pasos que se deben llevar a cabo.
- Grupo de acciones para promover los valores de la empresa en cuestión: para fortalecer la misma, para recompensar los logros alcanzados y para poder realizar las decisiones adecuadas.
- Estructura: para diseñar las formas para compartir el conocimiento.
- Ejecución: para tomar las decisiones adecuadas y oportunas.

Nomenclador, según la Real Academia Española, es un Catálogo que tiene la nomenclatura de una ciencia (Diccionario, 2017). El nomenclador o clasificador constituye un conjunto de conceptos generales definidos que permiten distinguir y formar agrupaciones de conceptos de orden en particular, buscando establecer una clasificación flexible y ordenada (2008).

Otro término importante a tener en cuenta para la investigación es la **Gestión de Nomencladores**, el que después del estudio realizado, se puede definir como el manejo de los datos poco variables en el tiempo, con la necesidad de tener la información disponible y asociada en todo momento, permitiendo crearla, modificarla, eliminarla y visualizarla en cualquier momento que se desee.

1.2. **Sistemas similares**

Después de una búsqueda de sistemas informáticos que gestionen información poco variable en el tiempo, tanto en el ámbito internacional como nacional se encontraron aplicaciones que aun cuando no se pueda

generalizar su uso, sirven de referencia para comprender los procesos de gestión de la información manejada.

1.2.1. Características de algunos sistemas internacionales.

Nomenclador Nacional

El Nomenclador Nacional fue creado en el año 1977 es, básicamente, un listado codificado de servicios médicos quirúrgicos, de laboratorios y hospitalarios, valorizados en unidades convencionales ajustados periódicamente (Salud).

Está orientado a un:

- Listado que conforma un catálogo, con códigos indicadores (prestaciones-grupo de prestaciones diagnósticos-equipos médicos) para orientar las actividades administrativas y de control.
- Cuerpo normativo y reglamentario que dará las guías de trabajo para la calidad y la instrumentación del sistema arancelado.
- Orden en tablas para la valorización de los códigos por medio de unidades referidas a honorarios profesionales y gastos de la práctica, que dan el arancel de la prestación (Moreno, y otros, 2011).

Nomenclador de procedimientos y servicios

El nomenclador de procedimientos y servicios es un documento técnico que consiste en el catálogo estandarizado de los procedimientos y servicios de salud disponibles en el mercado privado de salud, constituyéndose, por lo tanto, en un valioso instrumento dentro del Sistema Integrado de Información en Salud (Moreno, y otros, 2011).

El nomenclador ha sido elaborado de manera conjunta por la Asociación de Clínicas Particulares (ACP), la Asociación Peruana de Empresas de Seguros (APESEG) y la Asociación Peruana de Entidades Prestadoras de Salud (APEPS). La Superintendencia de Entidades Prestadoras de Salud (SEPS) participó en su elaboración como facilitador y coordinador, en su calidad de principal usuario del Nomenclador como estándar de procedimientos y servicios médicos (Moreno, y otros, 2011).

El instrumento desarrollado se enmarca dentro de la propuesta de Transmisión Electrónica de Datos Estandarizados del Expediente de Facturación (TEDEF) que la Superintendencia de Entidades Prestadoras de Salud (SEPS) viene impulsando y tiene el propósito de contar con una base de datos estandarizada de información administrativa y económica, permanentemente actualizada, de manera que se constituye en un insumo importante para la toma de decisiones en todos los niveles de las organizaciones de salud (Moreno, y otros, 2011).

El nomenclador detalla todos los procedimientos y servicios disponibles en el mercado privado de salud. No todos ellos son cubiertos por los financiadores. Los procedimientos y servicios que no tienen unidades están sujetos a condiciones pactadas entre financiadores y proveedores (Moreno, y otros, 2011).

Nomenclador cartográfico para personas con deficiencia visual.

Dos estudiantes de la Facultad de Matemática, Astronomía y Física de Córdoba en Argentina, desarrollaron un software de distribución libre y gratuita, para facilitar el desplazamiento de los discapacitados visuales por la ciudad de Córdoba. El nomenclador brinda información para trasladarse de un punto a otro, para personas con discapacidad visual.

El programa se distribuye bajo Licencia Pública General (GPL) para que pueda ser modificado, mejorado y ampliado por otros desarrolladores. Fue realizado en Java, un lenguaje que funciona en diversos sistemas operativos como Windows y Linux, entre otros. Tiene una arquitectura cliente-servidor. El servidor aloja la base cartográfica y procesa las solicitudes que recibe de los clientes. Así, se otorga mayor seguridad en la protección de los datos. Los autores consideran que en el futuro se deberán incorporar datos sobre el recorrido del transporte público y sumar tecnología de geo-posicionamiento satelital (Malbrán, y otros, 2008). El nomenclador cartográfico desarrollado durante los años 2007 y 2008 contiene las siguientes funcionalidades:

- Consulta sobre existencia de una dirección particular en la ciudad de Córdoba.
- Consulta del camino más corto a pie entre dos puntos de la ciudad.

El resultado de una consulta se compone de:

- Mapa del camino a realizar, contiene colores en alto contraste para facilitar la creación de un mapa mental por parte del usuario con deficiencia visual parcial.
- Instrucciones en texto que guían al usuario sobre cómo llegar a destino.
- Lectura audible de las instrucciones, permite guardar un archivo de sonido que podrá ser alojado en un celular o mp4 para su reproducción durante el trayecto.

1.2.2. Características de algunos sistemas nacionales.

Registro de la Clasificación Internacional de Enfermedades (RCIE)

El módulo tiene como objetivo principal la adopción de un lenguaje unificado y estandarizado, propuesto por la Organización Mundial de la Salud (OMS) en la Décima Revisión de la Clasificación Estadística Internacional de Enfermedades y Problemas Relacionados con la Salud (CIE-10). Todos los módulos que trabajen con diagnósticos tienen relación con este módulo, dado que es el que gestiona la codificación de todas las enfermedades y problemas relacionados con la salud que se puedan presentar en la atención médica diaria y causas de lesiones y muerte para su posterior cómputo y generación de estadísticas de morbilidad, mortalidad y servicios, entre otros (Marín, 2009).

El Editor a nivel nacional de este módulo es el único que puede agregar, modificar o eliminar los capítulos, grupos, categorías, subcategorías y problemas de salud que constan en la CIE, así como sus respectivas descripciones. Por otro lado, este módulo tiene como valor agregado que sirve de una especie de software educativo para el conocimiento de la CIE, en el cual el usuario a modo visualizador puede hacer búsquedas o leer todo lo relacionado con las diferentes enfermedades (Marín, 2009).

Registro de Problemas de Salud de la Atención Primaria (RPSAP)

El módulo tiene como objetivo principal gestionar una clasificación especial para los problemas de salud que se presentan en la Atención Primaria de Salud (APS) tratando que resulte una clasificación más breve y de manejo más simple que la CIE-10 y además compatible con esta. Permite realizar búsquedas dinámicas de acuerdo a criterios seleccionados por el usuario. Ofrece un lenguaje unificado y estándar para la información de la morbilidad en la APS.

El editor a nivel nacional de este módulo es el único que puede gestionar (agregar, modificar y/o eliminar) la información (nombre, grupo y descripción) de los riesgos, enfermedades y discapacidades registradas, así como gestionar la información (nombre y tipo) de los grupos y sub-grupos de discapacidades. En el caso del usuario visualizador puede obtener listado general de grupos de riesgos y riesgos incluidos en cada grupo, grupos de enfermedades y enfermedades incluidas en cada grupo, grupos de discapacidades y discapacidades incluidas en cada grupo y realizar búsquedas y conseguir en base a los criterios de la búsqueda, listado de riesgos específicos por grupos, listado de enfermedades específicas por grupos y detalles de códigos de CIE-10 que agrupan, y listado de discapacidades específicas por grupos y detalles de códigos de CIE-10 que agrupan (Moreno, y otros, 2011).

Registro de Enfermedades de Declaración Obligatoria (REDO)

Este módulo tiene como objetivo principal apoyar el proceso de vigilancia en salud que se realiza en las diferentes instancias del sistema sanitario, a través de una rápida detección y notificación de enfermedades transmisibles y no transmisibles, así como otras de interés para el Sistema Nacional de Salud (SNS) (Malbrán, y otros, 2008).

Registro de Indicadores y Conductas para la Atención Primaria (RICAPS)

Este módulo tiene como objetivo gestionar la información de las conductas asistenciales y la información de los indicadores de la salud de la atención primaria y es el único sistema automatizado para la clasificación o codificación de los indicadores y las conductas asistenciales definidos por el SNS para el nivel de APS (Malbrán, y otros, 2008).

1.2.3. Resultados del estudio

Después del análisis realizado a diferentes sistemas de gestión de nomencladores, tanto nacionales, como internacionales, se pudo conocer el funcionamiento de los sistemas de gestión de nomencladores y tenerlos

en cuenta en el desarrollo de esta investigación. Además, se toman las siguientes características relevantes a tener presente en el módulo de gestión de nomencladores a desarrollar:

- Disponibilidad inmediata de los nomencladores.
- Control sobre la gestión de los nomencladores.

1.3. Metodología de desarrollo de software

El proceso de creación de un software constituye un reto en la actualidad para cualquier equipo de desarrollo, pues este consta de la realización de varias tareas y actividades, las cuales pueden tornarse difíciles y engorrosas. El empleo de una metodología de desarrollo de software provee la técnica para la construcción de un software, abarcando una amplia selección de tareas, actividades de modelado y otras técnicas descriptivas (Pressman, 2010).

Para el desarrollo de esta investigación se determinó emplear la metodología ágil, Proceso Unificado Ágil para el Desarrollo de la actividad productiva en la Universidad de las Ciencias Informáticas (AUP-UCI). Dicha metodología fue estandarizada en la UCI, para guiar el proceso de desarrollo de software en sus distintos centros productivos.

1.3.1. Proceso Unificado Ágil para el Desarrollo de la actividad productiva en la Universidad de las Ciencias Informáticas

AUP es una versión simplificada del Proceso Unificado de Software (RUP). Describe de una forma fácil de entender una aproximación al desarrollo de software empresarial mediante técnicas ágiles manteniéndose fiel a las técnicas y conceptos definidos por RUP. Las técnicas ágiles empleadas por AUP incluyen desarrollo dirigido por pruebas, modelado ágil, gestión de cambios ágil y refactorización de base de datos para mejorar la productividad.

La metodología AUP-UCI responde a una variación que se realiza a la metodología ágil Proceso Unificado Ágil (AUP), de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI.

➤ Descripción de las fases que presenta AUP-UCI

AUP-UCI divide el ciclo de vida de la producción del software en tres fases:

Inicio: Fase donde se realiza un estudio inicial de la organización cliente que permite obtener una información fundamental acerca del alcance del proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto.

Ejecución: Fase donde se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

Cierre: Fase donde se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto (Rodríguez Sánchez, 2015).

Dentro de la fase de Ejecución, dicha metodología propone 7 disciplinas para el ciclo de vida de los proyectos de la UCI, donde sus flujos de trabajos son: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación

1.3.2. Escenarios para la disciplina requisitos que presenta AUP-UCI

Partiendo que la disciplina Modelado de negocio propone tres variantes a utilizar en los proyectos (casos de uso del negocio, descripción de proceso de negocio y modelo conceptual) y existen tres formas de encapsular los requisitos (casos de uso del sistema, historias de usuario y descripción de requisitos por proceso), surgen cuatro escenarios para modelar el sistema en los proyectos estos son:

Escenario No 1: Proyectos que modelen el negocio con casos de uso del negocio solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 2: Proyectos que modelen el negocio con modelo conceptual solo pueden modelar el sistema con casos de uso del sistema.

Escenario No 3: Proyectos que modelen el negocio con descripción de procesos de negocio solo pueden modelar el sistema con descripción de requisitos por procesos.

Escenario No 4: Proyectos que no modelen negocio solo pueden modelar el sistema con historias de usuario (Rodríguez Sánchez, 2015).

Para llevar a cabo el desarrollo del módulo se decide utilizar el escenario 2, ya que una vez que fue evaluado el negocio a informatizar se obtuvo que no fue necesario incluir las responsabilidades de las personas que ejecutan las actividades. Además, se comprobó que el negocio no está bien definido y el cliente no acompaña al equipo de desarrollo, permitiendo así, modelar exclusivamente los conceptos fundamentales del negocio. Este escenario se recomienda para proyectos donde el objetivo primario es la gestión y presentación de la información.



Fig 1: Escenario 2 Metodología AUP-UCI (Rodríguez Sánchez, 2015).

1.4. Plataforma de desarrollo

Una plataforma de desarrollo se puede definir a grandes rasgos como el entorno de software en el cual se desenvuelve la programación de una o varias aplicaciones (Beust, y otros, 2002).

1.4.1. *ECM Alfresco Community 4.2.f*

Alfresco es un Gestor de Contenidos Empresariales (ECM, por sus siglas en inglés) de código abierto compatible con sistemas operativos, tales como: Microsoft Windows y Linux. Esta plataforma de gestión de contenidos utiliza estándares abiertos para proveer gestión documental y gestión de contenidos web. Además, contiene un repositorio de contenidos que permite su integración con sistemas de archivos virtuales y aplicaciones web.

También posee una arquitectura escalable lo que le permite soportar las mejoras continuas sin perder la calidad en los servicios que presta. Además, brinda la posibilidad de prestar un conjunto de permisos, los cuales pueden ser aplicados a usuarios o grupos de usuarios en cualquier parte de un proceso, generando mayor seguridad y control de acceso a los documentos. También presenta una interfaz que ofrece servicios de contenido y colaboración para la personalización y el desarrollo de aplicaciones. Incluye además un conjunto de Interfaz de Aplicación de Programación (API) para la gestión de sus contenidos como son: OpenSearch para la búsqueda, REST para servicios web, Java API para la vinculación con código JAVA y otras. (Shariff, y otros, 2013).

Modelo de contenidos en ECM Alfresco Community 4.2.f

Un modelo de contenidos es un conjunto de tipos de contenido y aspectos. Además, define la estructura de información que acompaña a los documentos digitales permitiendo dotar de semántica al documento para que posteriormente se puedan implementar búsquedas y procesos más eficientes e inteligentes. Los modelos de contenidos definidos son identificados de forma unívoca mediante espacios de nombre (namespaces) y por defecto se conocen en Alfresco como Diccionario de Datos o Data Dictionary. Además, estos soportan ser extendidos para que el repositorio pueda manejar nuevos modelos de contenido. Un modelo de contenidos puede definir N tipos de contenido nuevos.

Creación de un nuevo modelo de contenido:

- Paso 1: Declarar un nuevo modelo y definir los tipos de contenidos personalizados.
- Paso 2: Adicionar al fichero bootstrap-context.xml, la ubicación del modelo de contenido definido anteriormente.
- Paso 3: Reiniciar Alfresco para que los cambios sean efectivos (Salinas Revelles).

1.4.2. Gestor de Documentos Administrativos (GDA) XABAL eXcriba 3.1

El Gestor de Documentos Administrativos XABAL eXcriba 3.1 es un sistema que automatiza los procesos documentales de las organizaciones. Esta emplea como núcleo de su sistema informático el Gestor de Contenido Empresarial (ECM) Alfresco Community en su versión 4.2f. Además, incluye las 3 primeras fases propuestas por la ISO 15489 para el ciclo de vida de un documento: creación, revisión/aprobación y publicación, manteniendo la seguridad de los documentos en formato electrónico. Entre sus principales funcionalidades se destacan el control de versiones, accesos y permisos, automatización de los flujos documentales, gestión de documentos y carpetas, realización de auditorías sobre los documentos electrónicos (UCI).

1.4.3. Lenguajes de desarrollo

Para el desarrollo de esta investigación se emplea como lenguaje de programación JavaScript 1.6, para el desarrollo en el lado del servidor y del cliente, en este último también se hizo uso del lenguaje HTML 5 (HyperText Markup Language) y XML 1.0 (Extensible Markup Language) para realizar los modelos de contenido y los servicios de datos.

JavaScript 1.6

JavaScript es un lenguaje interpretado con múltiples propósitos, pero se considera hasta ahora como un complemento. No requiere ningún software de servidor por lo tanto es un lenguaje de script del lado del cliente (Gauchat, y otros, 2012). Alfresco posee una Interfaz de Programación de Aplicación (API por sus siglas en inglés) en su versión 4.0 para JavaScript que permite la modificación y creación de nodos, aspectos y propiedades del repositorio de este. A través de esta API los desarrolladores de scripts podrán encontrar nodos, recorrer jerarquías de nodos, realizar búsquedas, examinar y modificar el valor de propiedades y aspectos. También, con los scripts se pueden crear nuevos ficheros, espacios o nodos de cualquier tipo (Andalucía).

HTML 5

Es el Lenguaje de Marcado de Hipertexto (Hyper-Text Markup Language, HTML por sus siglas en inglés) con el cual se construyen las páginas web que interpretan los navegadores web, las cuales no son más que documentos de hipertexto. De modo básico, constituye un conjunto de etiquetas de marcado que sirven para definir el texto y otros elementos que compondrán una página. Una combinación de estas, agrupadas y anidadas, constituye la estructura del documento. Dichas etiquetas le dicen al navegador web como mostrar todos los contenidos de la página y describe la estructura y la información en forma de texto, además de que complementa esta información con otros elementos como son las imágenes. Este fue el lenguaje utilizado para la construcción de la página web que formará posteriormente la interfaz de usuario (Bradenbaugh, 2000).

XML 1.0

XML es un lenguaje de marcado extensible para los documentos que contienen información estructurada. La información estructurada contiene tanto el contenido (palabras, imágenes, etc.) y alguna indicación de lo que desempeña el papel de contenido (por ejemplo, contenido en un encabezado de sección tiene un significado diferente del contenido de una nota al pie, que significa algo diferente de lo contenido en una figura leyenda o el contenido de una tabla de base de datos, etc.). Casi todos los documentos tienen cierta estructura. Un lenguaje de marcado es un mecanismo para identificar las estructuras de un documento. La especificación XML define una forma estándar de añadir marcadores para documentos (O'Reilly Media, 2014).

Para la implementación de este módulo se hace necesaria la utilización de este lenguaje en la creación de los modelos de contenido y los servicios de datos.

1.4.4. Tecnologías de desarrollo

Para el desarrollo del sistema se proponen las tecnologías definidas por el Centro de Informatización de la Gestión Documental (CIGED) de la Facultad 2, que pertenece a la Universidad de las Ciencias Informáticas. En el presente epígrafe se expone la fundamentación de todos los elementos seleccionados para el proceso de construcción del módulo de Gestión de nomencladores del (GDA) eXcriba 3.1.

JQuery

jQuery es una librería de JavaScript de código abierto que sirve como base para la programación de aplicaciones para la web. Brinda un conjunto de funciones y objetos que permiten abstraerse de uno de los principales problemas a los que se enfrenta el desarrollo de una aplicación web: la incompatibilidad entre navegadores. De manera general, también soluciona o da soporte a problemas frecuentes en el desarrollo de aplicaciones web actuales, entre ellos (jQuery, 2012).

- Acceso a una parte específica de una página.
- Modificación de la apariencia de una página web, su contenido o parte de este.
- Respuesta a la interacción de los usuarios con las páginas.

Teniendo en cuenta lo anteriormente planteado, se decidió emplear JQuery en conjunto con diferentes funciones nativas de JavaScript para el desarrollo de la capa de presentación del módulo.

AJAX

El término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML". Es una técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes. AJAX no es una tecnología en sí mismo.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.

- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

AJAX permite mejorar la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano. Las aplicaciones construidas con AJAX eliminan esa problemática, mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor (Eguíluz Pérez).

WebScript

WebScript es el framework que provee Alfresco para el desarrollo de servicios web basados en el estilo arquitectónico REST². Proporciona una forma de interactuar con el repositorio de contenido e integrar Alfresco con otros sistemas haciendo uso del lenguaje JavaScript (Alfresco, 2013). Este framework ha sido utilizado para el desarrollo de aplicaciones que utilizan Alfresco como repositorio, entre ellas Alfresco Share, Web Studio incluso el propio GDA eXcriba. Se decidió utilizar este framework porque ha sido utilizado para desarrollar todos los servicios que forman parte de la versión actual del GDA eXcriba, además se tuvo en cuenta la facilidad de implementación de los servicios haciendo uso del lenguaje JavaScript.

FreeMarker

FreeMarker es un motor de plantillas, así como una herramienta genérica para generar la salida de texto basado en plantillas. No es una aplicación para los usuarios finales en sí mismo, sino un paquete de Java que los programadores pueden utilizar en sus productos (Freemarker, 2013). Este motor es usado por el framework WebScript de Alfresco para proveer un mecanismo de generación de respuestas preformateadas, una vez finalizada la ejecución de un webScript, o sea, una vez finalizada la ejecución de la lógica de negocio se emite una respuesta cuya salida queda determinada por la estructura de la plantilla de presentación correspondiente al servicio en ejecución.

Una de las principales ventajas que usa el framework WebScript de este motor es que permite generar diferentes vistas y representaciones como resultado de la ejecución de un mismo servicio. De esta manera se logra la separación definitiva entre la lógica de negocio que se ejecuta durante una petición y el formato de salida de la respuesta generada.

JSON

² **REST** es una arquitectura de desarrollo web que puede ser utilizada en cualquier cliente HTTP (José, 2014).

JSON (Java Script Object Notation – Notación de objetos de JavaScript) es un formato de texto utilizado para el intercambio de datos. Está basado en un subconjunto del lenguaje de programación Java Script. JSON es completamente independiente del lenguaje que se emplee, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos (Json, 2013). Este es el formato utilizado para el intercambio de datos entre cliente y servidor, dicho intercambio puede ser entre el navegador *web* y el servidor Apache cuando el cliente envía una petición o entre el cliente web de eXcriba y Alfresco en el momento que se hacen las peticiones a los servicios.

Apache Tomcat v7.0

Apache Tomcat un contenedor web con soporte de Servlets³ y JSP, desarrollado en código abierto en lenguaje Java por lo que funciona en cualquier sistema operativo que disponga de una máquina virtual Java. Es desarrollado en un entorno abierto y participativo. Publicado bajo la licencia Apache versión 2, siendo una marca registrada de la Fundación de Software Apache. Teniendo además una aplicación de administración intuitiva basada en web (Tomcat, 2014).

1.5. Sistema gestor de base de datos

Un sistema gestor de bases de datos es el software que se encarga de manejar la creación y todos los accesos a la base de datos. Se compone de un lenguaje de definición, manipulación y consulta de datos (Faudón, y otros, 2001).

1.5.1. PostgreSQL v9.4.2

PostgreSQL es un sistema de administración de bases de datos de propósito general y objeto-relacional, fue diseñado para ejecutarse en plataformas tipo UNIX. Sin embargo, también se diseñó para ser portátil, de modo que pudiera ejecutarse en varias plataformas como Mac OS X, Solaris y Windows. Este es un software gratuito y de código abierto. Su código fuente está disponible bajo la licencia PostgreSQL, una licencia liberal de código abierto. PostgreSQL requiere muy pocos esfuerzos mantenidos debido a su estabilidad. Por lo tanto, si se desarrollan aplicaciones utilizando dicho gestor de base de datos, el costo total de propiedad es bajo en comparación con otros sistemas similares (PostgreSQL, 2018).

1.6. Lenguaje de modelado UML

UML se define como un "lenguaje que permite especificar, visualizar y construir los artefactos de los sistemas de software" (Larman, 2003). Este permite mediante diagramas definir la estructura y el comportamiento de un sistema, subsistema o módulo.

³ Es una tecnología que permite crear aplicaciones web interactivas (dinámicas), es decir, le permite al usuario interactuar con la aplicación (hacer consultas, insertar y eliminar datos) (2018).

1.6.1. Herramienta CASE: Visual Paradigm para UML 8.0

Visual Paradigm utiliza UML como lenguaje de modelado. Es una herramienta para el desarrollo de aplicaciones utilizando modelado UML diseñada para ingenieros de software, analistas de sistemas y arquitectos de sistemas. Estos deben estar interesados en la construcción de sistemas a gran escala y necesitan confiabilidad y estabilidad en el desarrollo orientado a objetos. Captura requisitos mediante el modelado de los casos de uso. Permite además exportar los diagramas a imágenes y páginas HTML (Paradigm, 2018).

1.7. Axure RP PRO v7.0: Diseño de prototipos

Axure es uno de los estándares de la industria de software orientada a generar prototipos básicos o avanzados de forma fácil. Esta herramienta está dirigida tanto a la creación de aplicaciones web como de escritorio. El programa tiene por defecto una amplia gama de widgets⁴ con los cuales se pueden crear interfaces de una forma rápida y sencilla. Se puede crear controles personalizados mediante la combinación de los widgets existentes y la asignación de acciones en respuesta a eventos como OnClick, Onmouseover y onMouseOut o gestos como deslizar. Se podrá crear widgets personalizados (que se podrán reutilizar) que pueden simular funcionalidades o módulos que se pueden necesitar para simular un proceso, interacción, etc. Este es uno de sus puntos fuertes, la interactividad que simula el comportamiento de las interfaces hace que sea una herramienta muy útil para el de prototipado (Axure, 2014).

Conclusiones parciales

Una vez realizado el estudio del marco teórico de la investigación se arribó a las siguientes conclusiones:

- El estudio de los principales conceptos referentes a los nomencladores permitió una mejor comprensión de la presente investigación.
- El estudio de sistemas homólogos, permitió identificar las características principales de los sistemas de gestión de nomencladores.
- El estudio realizado a la metodología, tecnologías y herramientas, proporcionó los conocimientos necesarios para su utilización en el desarrollo de la solución propuesta.

⁴ Pequeñas aplicaciones que tienen como principal cometido mostrar y dar fácil acceso a algunas de las principales funciones del terminal.

Capítulo II. Propuesta de solución

Introducción

El presente capítulo tiene como objetivo describir la propuesta de solución y precisar las características con que contará el módulo que se desarrollará. Se evidencian las disciplinas Modelado de negocio, Requisitos y Análisis y diseño que propone la metodología AUP-UCI para el ciclo de vida del proyecto. Para ello se realiza una representación del modelo conceptual, se definen los requisitos funcionales y no funcionales del módulo. Posteriormente se presenta el diagrama de casos de uso; así como su descripción para una mayor comprensión. Además, se explica la técnica de trazabilidad de requisitos: matrices de trazabilidad. Se muestra la arquitectura a utilizar, los patrones de diseño y la distribución física del sistema mediante el diagrama de despliegue.

2.1. Modelo conceptual

Un modelo conceptual también conocido como modelo de dominio tiene como propósito fundamental organizar y representar, de manera semiformal y unívoca, el conocimiento de un área o campo específico asociado a un sistema de gestión o de información. Está orientado a describir semántica y aseveraciones sobre la información del dominio particular que representa (Planeación, 2013).

Para una mayor comprensión del módulo de Gestión de nomencladores, se propone la realización de un modelo conceptual; pues permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del módulo a desarrollar. Además, brinda a los usuarios, clientes, desarrolladores e interesados, un vocabulario común para poder entender el negocio o un dominio de problema. En la figura que se muestra a continuación se procederá a explicar la relación entre los conceptos que se identifican en el siguiente modelo.

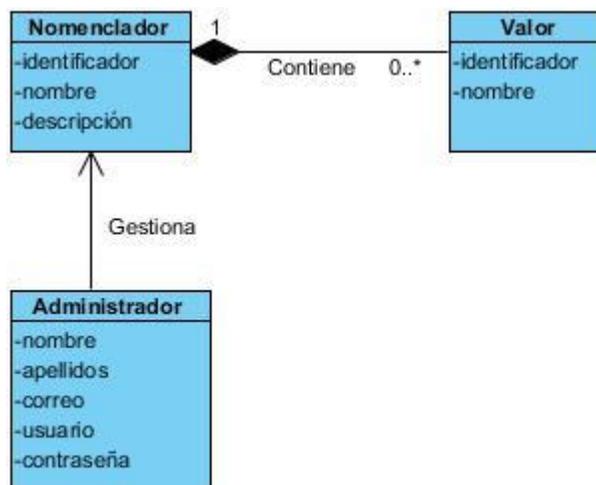


Fig 2: Modelo conceptual.

Capítulo II

Nomenclador: Representa la información poco variable en el tiempo.

Valor: Representa el valor asociado al nomenclador.

Administrador: Usuario con privilegios administrativos encargado de gestionar los nomencladores en el sistema.

2.1.1. Diccionario de datos

Un diccionario de datos es una colección de nombres, definiciones y atributos sobre los elementos de datos que se utilizan o capturan en una base de datos, sistema de información o parte de un proyecto de investigación. Describe los significados y los propósitos de los elementos de datos dentro del contexto de un proyecto, y proporciona orientación sobre la interpretación, los significados aceptados y la representación (Rouse, 2018).

Capítulo II

Tabla 1: Descripción del concepto "Nomenclador".

Nomenclador

Descripción	Esta entidad representa las propiedades de un nomenclador en el módulo gestión de nomencladores..					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Identificador	Valor único que identifica a cada nomenclador	Número entero	No	Si	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	Puede tomar como valor máximo $2 \cdot 10^9$.
Nombre	Nombre propio del nomenclador en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	Si	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio. Admite hasta 50 caracteres.	
Descripción	Descripción del nomenclador en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	No	Puede tomar cualquier combinación de letras [a-z], [A-Z], espacio y números.	

Capítulo II

Tabla 2: Descripción del concepto "Valor".

Valor

Descripción	Esta entidad representa los valores que se asocian al nomenclador en el módulo gestión de nomencladores.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Identificador	Valor único que identifica a cada nomenclador	Número entero	No	Sí	Toma valores que se obtengan de la combinación de los dígitos del [0-9].	Puede tomar como valor máximo $2 \cdot 10^9$.
Nombre	Nombre propio del valor que se le asocia al nomenclador en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.	Cadena de caracteres.	No	Sí	Admite cualquier tipo de caracteres. Admite hasta 100 caracteres.	

2.2. Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, especifican las acciones que el sistema debe ser capaz de realizar, sin tener ningún tipo de restricción física y cómo debe reaccionar ante situaciones particulares. Estos deben ser de fácil entendimiento para el usuario y para los desarrolladores (Dobson, y otros, 2009).

En el análisis para el desarrollo de la solución propuesta, fueron obtenidos 16 requisitos funcionales descritos a continuación:

RF1. Crear nomenclador: El sistema debe permitir al usuario con privilegios de administración crear un nomenclador. Se registrarán los siguientes campos: Nombre, Descripción y Valores.

RF2. Editar nomenclador: El sistema debe permitir al usuario con privilegios de administración Editar un nomenclador. Se podrán editar los siguientes campos: Nombre, Descripción y Valores.

RF3. Ver detalles del nomenclador: El sistema debe permitir al usuario con privilegios de administración visualizar los siguientes campos: Nombre, Descripción y Valores.

RF4. Eliminar nomenclador: El sistema debe permitir al usuario con privilegios de administración eliminar un nomenclador.

RF5. Listar nomenclador: El sistema debe permitir al usuario con privilegios de administración visualizar en forma de tabla la información asociada a un nomenclador. De cada nomenclador se mostrará: Nombre, Descripción y Valores.

RF6. Buscar nomenclador: El sistema debe permitir al usuario con privilegios de administración buscar nomenclador(es) a partir de un criterio insertado. Solo se visualizarán los nomencladores en los que coincidan parcial o totalmente el criterio insertado.

RF7. Importar nomenclador: El sistema debe permitir al usuario con privilegios de administración importar el o los nomenclador(es) en formato json.

RF8. Exportar nomenclador: El sistema debe permitir al usuario con privilegios de administración exportar un archivo json con los nomencladores.

RF9. Eliminar valor: El sistema debe permitir al usuario con privilegios de administración eliminar un valor.

RF10. Ordenar nomencladores por la columna nombre: El sistema debe permitir al usuario con privilegios de administración ordenar ascendente y descendientemente los nomencladores por la columna nombre.

RF11. Ordenar nomencladores por la columna descripción: El sistema debe permitir al usuario con privilegios de administración ordenar ascendente y descendientemente los nomencladores por la columna descripción.

RF12. Ordenar nomencladores por la columna valores: El sistema debe permitir al usuario con privilegios de administración ordenar ascendente y descendentemente los nomencladores por la columna valores.

RF13. Configurar cantidad de elementos a listar: El sistema debe permitir al usuario con privilegios de administración configurar la cantidad de elementos a listar.

RF14. Filtrar nomenclador por nombre: El sistema debe permitir al usuario con privilegios de administración filtrar los nomencladores por el campo nombre.

RF15. Filtrar nomenclador por descripción: El sistema debe permitir al usuario con privilegios de administración filtrar los nomencladores por el campo descripción.

RF16. Filtrar nomenclador por valores: El sistema debe permitir al usuario con privilegios de administración filtrar los nomencladores por el campo valores.

2.3. Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (Sommerville, 2005).

A continuación, se muestran los requisitos no funcionales definidos para el módulo de gestión de nomencladores:

Usabilidad:

El módulo debe proporcionar mensajes de error que sean informativos y orientados al usuario final. Además, debe contar con una interfaz gráfica bien definida.

Confiabledad:

El módulo contará con campos obligatorios, debidamente identificados para garantizar la integridad de la información que se introduce.

Fiabilidad

La precisión y exactitud de las respuestas del módulo se corresponden con la calidad y exactitud de la información insertadas por el usuario con privilegios de administración y la información contenida en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.

Legales:

Las herramientas seleccionadas para el desarrollo del módulo están respaldadas por licencias libres, bajo las condiciones de software libre.

Software:

El módulo podrá ser utilizado mediante el navegador Mozilla Firefox.

2.4. Descripción de la propuesta de solución

Para dar solución a la situación problemática planteada en esta investigación se propone como solución: la creación del módulo para la gestión de nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1. Este permitirá a un usuario con privilegios de administración gestionar los nomencladores, permitiéndole crear, editar, ver detalles y eliminar un nomenclador, entre otras funcionalidades especificadas en el epígrafe 2.2. Además, como valor agregado, el módulo proveerá a los desarrolladores un componente visual para formularios que permitirá ser integrado en el desarrollo de personalizaciones del GDA XABAL eXcriba 3.1.

2.5. Técnicas de validación de requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de estos detalla realmente el sistema que el usuario necesita o el cliente desea (Alvarez, 2013). En esta investigación se utilizaron como técnicas de validación de requisitos: Reviews o Walk-throughs y Prototipos.

Reviews o Walk-throughs: Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante (del Valle Rojo, 2013). En dicha revisión con el cliente, el mismo una vez revisados los requisitos que le fueron propuestos, firmó un acta de aceptación de requisitos, la cual puede observarse en el Anexo 2 Acta de aceptación de requisitos.

Prototipos: Algunas propuestas se basan en obtener de la definición de requisitos: prototipos, que sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema (Escalona, 2002).

2.6. Definición de los casos de usos del sistema

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado notorio de valor para un actor concreto (Jacobson, y otros, 2004).

2.6.1. Definición de los actores

Un actor es un conjunto coherente de roles que los usuarios de casos de uso desempeñan cuando interaccionan con estos casos de uso (Jacobson, y otros, 2004).

Tabla 3: Definición de los actores.

Actores	Justificación
Administrador	Usuario con privilegios de administración con la responsabilidad de gestionar el módulo de gestión

	de nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.
--	--

2.6.2. Diagrama de caso de uso del sistema

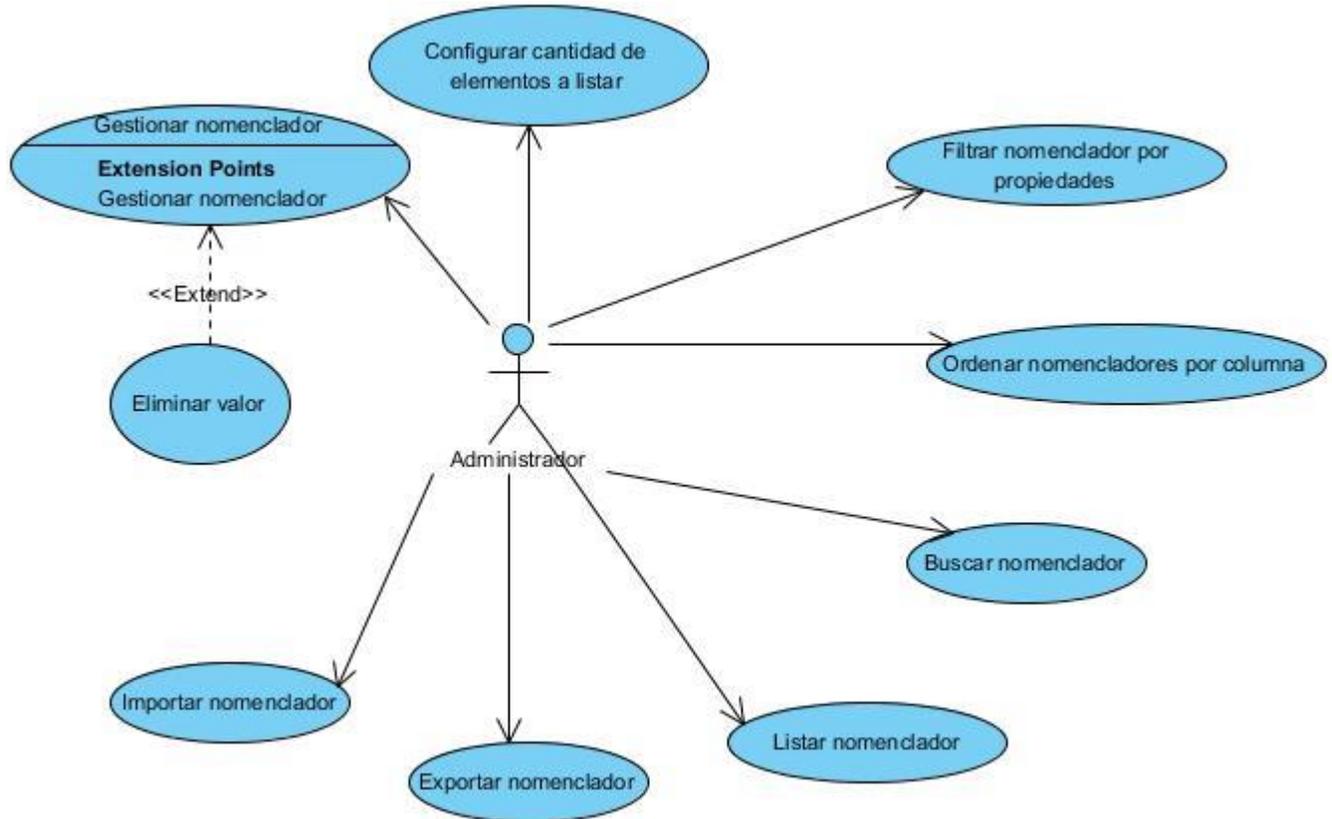


Fig 3: Diagrama de casos de uso.

2.6.3. Descripción de casos de uso del sistema

A continuación, se expone la descripción del caso de uso Gestionar nomenclador. El resto de los casos de uso se pueden consultar en el Anexo 3 Casos de uso del sistema.

Tabla 4: Descripción del caso de uso: Gestionar nomenclador.

Caso de Uso	Gestionar nomenclador
Objetivo	Crear, modificar, listar y elimina nomenclador.
Actores	Administrador
Resumen	El caso de uso comienza cuando el administrador desea realizar alguna de estas acciones: Crear, Modificar, Ver detalles o Eliminar los nomencladores. El caso de uso termina cuando realiza alguna de estas acciones.
Complejidad	Media
Prioridad	Media
Precondiciones	El administrador debe estar autenticado en el sistema. Para las secciones Editar y Eliminar debe existir al menos un nomenclador registrado.

Capítulo II

Post-condiciones	Se almacenan los nomencladores creados o modificados, permite visualizar los mismos y en cuanto al nomenclador eliminado desaparece de manera permanente del sistema.	
Flujo de eventos		
Flujo básico: Gestionar nomenclador.		
	Actor	Sistema
1.	Crea, modifica, lista o elimina un nomenclador.	
2.		Permite realizar varias acciones con el nomenclador: <ul style="list-style-type: none"> - Crear un nomenclador. Ver Sección 1: Crear Nomenclador. - Editar un nomenclador. Ver Sección 2: Editar Nomenclador. - Listar un nomenclador. Ver Sección 3: Ver detalles del nomenclador. - Elimina un nomenclador. Ver Sección 4: Eliminar nomenclador.
3.		Termina el caso de uso.
Sección 1: “Crear Nomenclador.”		
Flujo básico		
	Actor	Sistema
1.	Accede a la opción “Crear nomenclador” dentro del menú “Administración de nomencladores”.	
2.		El sistema muestra un formulario con los siguientes datos: <ul style="list-style-type: none"> -Nombre* -Descripción* -Valores
3.	Inserta los valores valores.	
4.	Presiona el botón “Aceptar”.	
5.		El sistema valida la información insertada.
6.		El sistema crea el tipo de contenido nomenclador con los datos insertados.
7.		El sistema crea dentro de Nomenclador/ los nomencladores
8.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón “Cancelar”.		
	Actor	Sistema
1.	Presiona el botón “Cancelar”.	

Capítulo II

2.		El sistema cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1		El sistema muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2		Continuar en el paso 3 del flujo básico.
6.a Existen campos incorrectos.		
	Actor	Sistema
1.		El sistema muestra el mensaje: " El nombre es incorrecto."
2.		Continuar en el paso 3 del flujo básico.
Sección 2: "Editar Nomenclador."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Editar" del nomenclador escogido.	
2.		El sistema muestra un formulario con los siguientes datos: Datos -Nombre* -Descripción* -Valores
3.	Inserta los valores.	
4.	Presiona el botón "Aceptar".	
5.		El sistema valida la información insertada.
6.		El sistema edita el tipo de contenido nomenclador con los datos insertados.
7.		Termina el caso de uso.
Flujos alternos		
4.a Presionar el botón "Cancelar".		
	Actor	Sistema
1.	Presiona el botón "Cancelar".	
2.		El sistema cancela la acción y muestra la interfaz anterior.
3.		Termina el caso de uso.
5.a Existen campos obligatorios vacíos.		
	Actor	Sistema
1.		El sistema muestra encima del campo vacío el mensaje: "El valor no puede estar vacío."
2.		Continuar en el paso 3 del flujo básico.
6.a Existen campos incorrectos.		
	Actor	Sistema

Capítulo II

1.		El sistema muestra el mensaje: " El nombre es incorrecto."
2.		Continuar en el paso 3 del flujo básico.
Sección 3: "Ver detalles del nomenclador ."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Ver detalles" del nomenclador escogido.	
2.		El sistema muestra una interfaz con todos los detalles del nomenclador.
3.	Selecciona el botón "Aceptar".	
4.		Termina el caso de uso
Sección 4: "Eliminar Nomenclador ."		
Flujo básico		
	Actor	Sistema
1.	Presiona el ícono "Eliminar" del nomenclador escogido.	
2.		El sistema muestra la interfaz de confirmación de eliminación.
3.	Selecciona el botón "Eliminar".	
4.		El sistema elimina el nomenclador escogido.
5.		El sistema actualiza la tabla.
6.		Termina caso de uso.
Flujos alternos		
3.a Presionar el botón "Cancelar".		
	Actor	Sistema
1	Presiona el botón "Cancelar".	
2		El sistema permite cerrar la interfaz de confirmación de eliminación.
Relaciones	CU incluidos	N/A
	CU extendidos	Sección 1: CU 6. Eliminar valor Sección 2: CU 6. Eliminar valor
Requisitos no funcionales	N/A	
Asuntos pendientes	N/A	
Prototipo elemental de interfaz gráfica de usuario		
<i>Sección 1: Crear nomenclador</i>		

Capítulo II

Crear nomenclador

Nombre: *

Descripción: *

Valores:

Aceptar Cancelar

Sección 2: Editar nomenclador

Editar nomenclador

Nombre: *

Descripción: *

Valores:

Aceptar Cancelar

Sección 3: Ver detalles del nomenclador

Detalles del nomenclador

Nombre: Tipo de documento
Descripción: Representa las diferentes tipologías identificadas en la empresa en cuestión.
Valores: Carta, Instrucción, Procedimiento, DHL, Indicación

Aceptar Cancelar

Sección 4: Eliminar nomenclador

Eliminar nomenclador

¿Está seguro que desea eliminar este elemento?

Eliminar Cancelar

2.7. Técnica de trazabilidad de requisitos

Cuando se trabaja con requisitos en el desarrollo de un proyecto, un aspecto fundamental es asegurar la trazabilidad de requisitos. Una de las técnicas utilizadas para realizar la representación de la trazabilidad son las llamadas matrices de trazabilidad. Dicha técnica recoge información bi-direccional de trazas: trazabilidad entre requisitos, trazabilidad de requisitos con su origen y trazabilidad de requisitos con su implementación (Sevilla, 2016).

En esta investigación se realizaron 4 tipos de matrices de trazabilidad: Matriz de trazabilidad RF-Concepto, Matriz de trazabilidad RF-RF, Matriz de trazabilidad RF-CU y Matriz de trazabilidad CU-CP.

A continuación, se enumeran los casos de uso del sistema para facilitar el entendimiento de la matriz de trazabilidad RF-CU que se muestra posteriormente.

CU1. Gestionar nomenclador

CU2. Listar nomenclador

CU3. Buscar nomenclador

CU4. Importar nomenclador

CU5. Exportar nomenclador

CU6. Eliminar valor

CU7. Ordenar nomencladores por columna

CU8. Configurar cantidad de elementos a visualizar

CU9. Filtrar nomenclador por propiedades

Tabla 5: Matriz de trazabilidad de RF-CU.

(9) Use Case	RF1. Crear nomenclador	RF2. Editar nomenclador	RF3. Ver detalles del nomenclador	RF4. Eliminar nomenclador	RF5. Listar nomenclador	RF6. Buscar nomenclador	RF7. Importar nomenclador	RF8. Exportar nomenclador	RF9. Eliminar valor	RF10. Ordenar nomencladores por la columna nombre	RF11. Ordenar nomencladores por la columna descripción	RF12. Ordenar nomencladores por la columna valores	RF13. Configurar cantidad de elementos a visualizar	RF14. Filtrar nomenclador por nombre	RF15. Filtrar nomenclador por descripción	RF16. Filtrar nomenclador por valores
<input checked="" type="radio"/> Gestionar nomenclador	✓	✓	✓	✓												
<input type="radio"/> Listar nomenclador					✓											
<input type="radio"/> Buscar nomenclador						✓										
<input type="radio"/> Importar nomenclador							✓									
<input type="radio"/> Exportar nomenclador								✓								
<input type="radio"/> Eliminar valor									✓							
<input type="radio"/> Ordenar nomencladores p...										✓	✓	✓				
<input type="radio"/> Configurar cantidad de ele...													✓			
<input type="radio"/> Filtrar nomenclador por pro...														✓	✓	✓

Como se puede observar en la tabla anterior cada requisito identificado ha sido cubierto por al menos un caso de uso del sistema, lo que significa que la especificación de caso de uso propuesta satisface todas las necesidades del cliente. El resto de las matrices podrán ser consultadas en el Anexo 4 Matrices de trazabilidad.

2.8. Descripción de la arquitectura

La arquitectura de software de un sistema de programa o computación es la estructura del mismo, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos (Pressman, 2010).

Para el desarrollo del Gestor de Documentos Administrativos Xabal eXcriba 3.1 se utilizó la arquitectura en capas o n-capas, como también es conocida, por consiguiente, para el desarrollo del módulo gestión de nomencladores se utilizará dicha arquitectura.

2.8.1. Arquitectura en capas

La arquitectura basada en capas es un miembro de la familia de estilos de llamada y retorno, esta arquitectura se define como una organización jerárquica tal que cada capa proporciona servicios a la capa inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior (Garlan, y otros, 1994). Esta arquitectura del software va a contar con 3 capas, estas son: Capa de presentación, Capa de aplicación y Capa de datos.

Capa de presentación

En esta capa se brindan dos clientes, el cliente web eXcriba y el explorador de Alfresco:

- El cliente web de eXcriba es un Gestor de Documentos Administrativos cuyo objetivo es brindar servicios que cumplan con los requisitos de la gestión documental, apoyando a las actividades de los procesos documentales.
- El explorador de Alfresco es incluido en el núcleo como una herramienta de administración de apoyo, ya que expone todas las funcionalidades del ECM utilizado y es requerido para llevar a cabo acciones más avanzadas como el mantenimiento de las reglas de negocio y la gestión de grupos y usuarios (Torres Hernández, 2013).

Capa de aplicación

Esta capa posee los servicios de datos, estos brindan servicios avanzados para la gestión de contenidos y son incluidos dentro del repositorio de Alfresco.

Capa de datos

Esta capa agrupa al sistema de ficheros y la base de datos. El sistema de ficheros está compuesto a la vez por los ficheros del contenido y los índices de búsqueda. En esta capa se guarda toda la información y el contenido que es gestionado por el núcleo cuando realiza sus funciones (Torres Hernández, 2013).

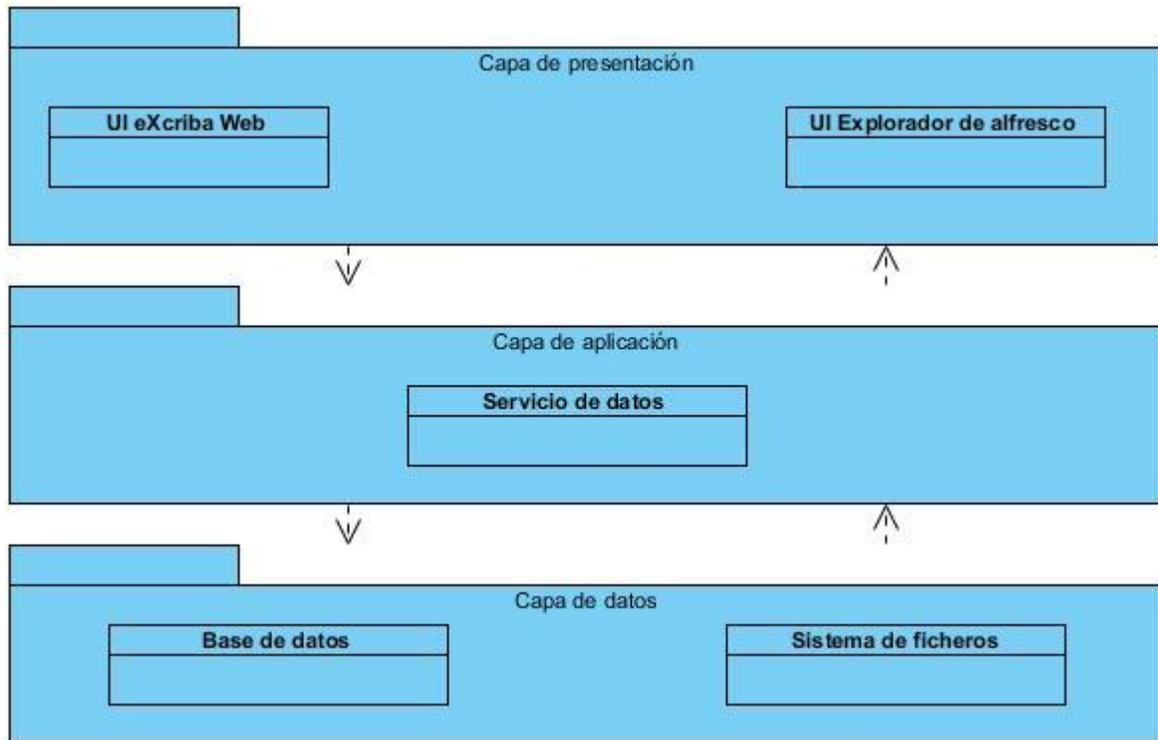


Fig 4: Descripción de la arquitectura.

2.9. Modelo de despliegue

Describe la distribución física del sistema en términos de cómo las funcionalidades se distribuyen entre los nodos de computación sobre los que se va a instalar el sistema. Cada nodo representa un recurso de computación. Los nodos tienen relaciones entre ellos que representan los medios de comunicación que estos poseen. La funcionalidad de un nodo viene representada por los componentes que se ejecutan en él (Alonso Amo, y otros, 2005). Para cada componente de un diagrama es necesario documentar las características técnicas requeridas para la ejecución del mismo. El modelo de despliegue está compuesto por:

Nodos: Elementos de proceso en el sistema. Los nodos pueden tener las siguientes propiedades:

- Nombre
- Una descripción que proporciona información sobre el procesador, capacidad de almacenamiento, capacidad de memoria o cualquier otra información sobre las posibilidades del dispositivo.
- Una lista de los procesos y hebras que se ejecutan en el procesador. Esta lista también puede enumerar los componentes de software que se ejecutan dentro de cada proceso.
- Una lista de las unidades de despliegue que se instalarán en el nodo (IBM Corp, 2006).

Dispositivos: Dispositivos físicos, que no tienen ninguna capacidad de proceso (al nivel modelado de abstracción), que dan soporte a los nodos de procesador. Los dispositivos pueden tener las siguientes propiedades:

- Nombre
- Una descripción, que proporciona información sobre las posibilidades del dispositivo (IBM Corp, 2006).

Conectores: Conexiones entre nodos, y entre nodos y dispositivos. Los conectores pueden tener información asociada en relación con la capacidad o ancho de banda del conector (IBM Corp, 2006).

2.9.1. Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas entre los distintos nodos que componen un sistema y la distribución 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 por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria. Los estereotipos permiten precisar la naturaleza del equipo: dispositivos, procesadores y memoria (2006).

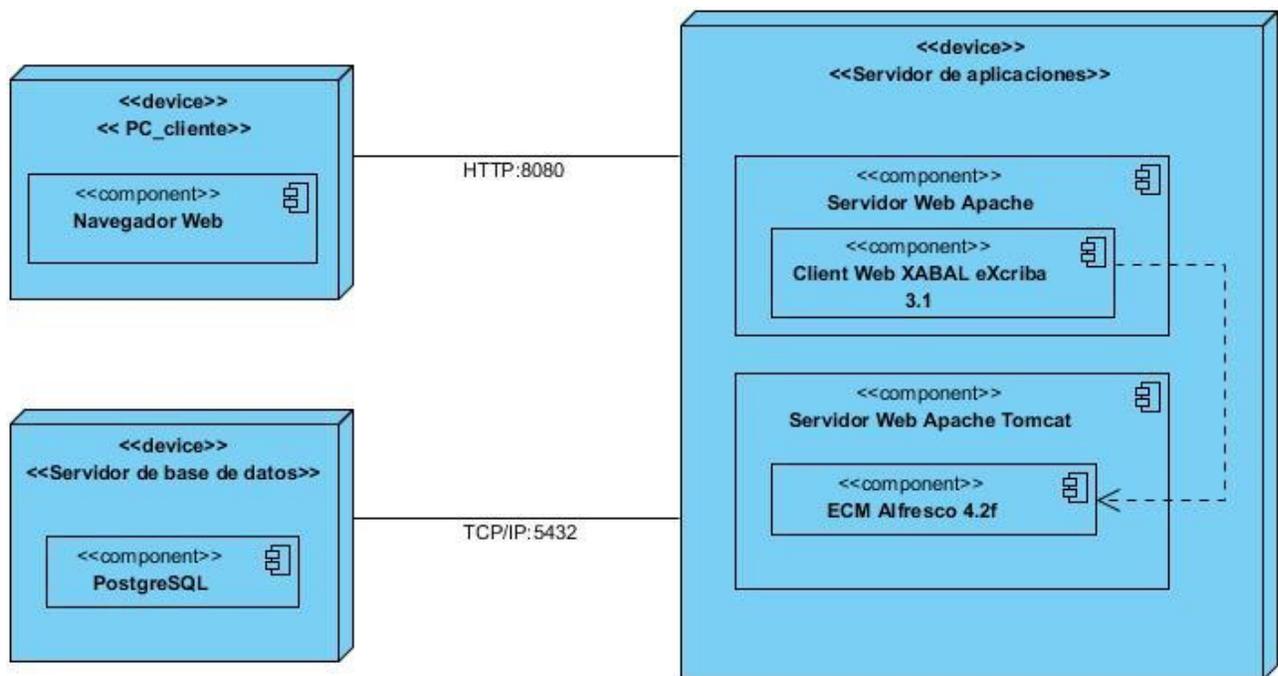


Fig 5: Diagrama de despliegue.

A continuación, se describen los nodos:

PC cliente: En este nodo físico estarán alojados los navegadores web (firefox, etc.) para acceder a las páginas clientes que interactúan con las páginas servidoras. Se procesan todas las interfaces de usuario las cuales son las encargadas de establecer la comunicación entre el usuario y el GDA eXcriba.

Servidor de aplicaciones: En la cual estará alojado el sistema en cuestión, o sea, la interfaz web (Front End) del sistema GDA, hosteado en el servidor web Apache, y su núcleo el ECM Alfresco además de la capa de servicios incluida en el propio sistema y el servidor de aplicaciones Apache-Tomcat.

- **Servidor Web Apache:** El servidor web apache provee un mecanismo para ofrecer la aplicación web a los usuarios. En el caso del presente módulo, en este nodo se encuentran todos los componentes referentes a la capa lógica del negocio.
- **Servidor Web Apache Tomcat:** Tomcat es un servidor web cuyo uso sólo tiene el propósito de contener al ECM Alfresco. En el caso de este módulo, aquí se encuentran desplegados los servicios que serán usados para brindar funcionalidades al módulo.

Servidor de base de datos: En este servidor es donde se encuentra el sistema gestor de bases de datos PostgreSQL. Aquí se encuentra toda la información estructurada, en el caso del presente módulo es donde se encuentra toda la información referente a los nomencladores.

Descripción de las relaciones entre los nodos

Primeramente, el administrador, a través del ordenador cliente interactúa con el servidor web apache mediante el protocolo de comunicación HTTP, luego este servidor se comunica a través del protocolo HTTP con el servidor web tomcat y este a su vez con el servidor de base de datos por TCP/IP para acceder a la información almacenada.

2.10. Patrones de diseño

Un patrón es una descripción de un problema y la solución, a la que se da un nombre, y que se puede aplicar a nuevos contextos; idealmente, proporciona consejos sobre el modo de aplicarlo en varias circunstancias, y considera los puntos fuertes y compromisos (Larman, 2003).

2.10.1. Patrones GRASP

Los Patrones GRASP (General Responsibility Assignment Software Patterns) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (Larman, 2003).

Patrón Experto: Es la clase que tiene la información necesaria para realizar la responsabilidad. Este se utiliza a la hora de modelar las entidades persistentes de la solución donde las funcionalidades propias de cada información reimplementan en la clase que contiene toda la información, o sea, la clase experta en la información (Larman, 2003).

Patrón Creador: guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental de este patrón es encontrar

un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento (Larman, 2003).

Patrón Controlador: propone el diseño de clases con la responsabilidad de controlar el flujo de eventos del sistema a clases específicas. Es más que asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase (Larman, 2003).

Patrón Bajo Acoplamiento: es un principio que se debe recordar durante las decisiones de diseño: es la meta principal que es preciso tener presente siempre. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras, con las que conoce y con que recurre a ellas. El patrón propone un diseño de clases más independiente, lo que reduce el impacto del cambio y facilita la reutilización en otros sistemas (Larman, 2003).

Patrón Alta Cohesión: es un patrón evaluativo que el desarrollador aplica al valorar sus decisiones de diseño. Grady Booch señala que se da una alta cohesión funcional cuando los elementos de un componente (clase, por ejemplo) "colaboran para producir algún comportamiento bien definido" (Larman, 2003).

2.10.2. Patrones GOF

Los Patrones Gang of Four (GOF) describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Gamma, 1994).

- **Creacionales:** describen las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (Gamma, 1994).
- **Estructurales:** describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionales pueden ser incluso objetos simples u objetos compuestos (Gamma, 1994).
- **Comportamiento:** definen la comunicación e iteración entre los objetos de un sistema. El propósito de este patrón es reducir el acoplamiento entre los objetos (Gamma, 1994).

Dentro de la amplia gama de patrones GOF, se escogió para la confección del módulo de nomencladores el siguiente:

Singleton: Garantiza que solamente se cree una instancia de la clase y provee un punto de acceso global a él. Todos los objetos que utilizan una instancia de esa clase, la utilizan a ella misma (Guía de construcción de software en java con patrones de diseño).

Conclusiones parciales

Una vez realizada la propuesta de solución de la investigación se arribó a las siguientes conclusiones:

- La definición del modelo conceptual con su respectivo diccionario de datos, facilitó el entendimiento del negocio a informatizar.
- La definición de los requisitos funcionales permitió identificar las funcionalidades a desarrollar en el módulo de gestión de nomencladores.
- La descripción de casos de uso del sistema proporcionó una visión inicial de cómo pudiera quedar el módulo en términos de interfaz.
- La arquitectura de software empleada, el diagrama de despliegue y los patrones de diseño sentaron las bases para la siguiente etapa de construcción del software.

Capítulo III. Implementación y prueba.

Introducción

El presente capítulo tiene como objetivo la implementación y prueba del módulo, evidenciándose las disciplinas: Implementación, Pruebas Internas y Pruebas de Aceptación que propone la metodología AUP-UCI para el ciclo de vida del proyecto. En dicho capítulo se definen el modelo de contenido, los webscripts y la interfaz gráfica del módulo. Además, se diseñan los casos de pruebas para facilitar la disciplina Pruebas internas.

3.1. Modelo de contenido

Un modelo de contenido proporciona una base para la estructuración y el trabajo con contenidos específicos. Para la creación del modelo de contenido referente a los nomencladores se definieron dos tipos de contenido: “nomenc:Nomenclador” referente al nomenclador y “nomenc:Values” referente al valor que se le asignará luego al nomenclador. Ambos heredan de “cm:content” que es el tipo de contenido que posee las propiedades básicas de todo tipo de documento. Se estableció la asociación del tipo Child-association la cual funciona como un cascade delete, es decir, el hijo sólo existe si existe el padre. Esta asociación se estableció de tipo “nomenc:assoc” entre el tipo de contenido “nomenc:Nomenclador” con “nomenc:Values”, para establecer las relaciones entre los diferentes tipos de contenidos. Para insertar el modelo de contenido personalizado en Alfresco se crea el archivo nomenclator.xml. Luego se ubica en la carpeta extensión del servidor de Alfresco (alfresco/tomcat/shared/classes/alfresco/extension/).

3.2. Webscript

Alfresco para la implementación de los módulos web propone en su arquitectura dos tipos de webscripts, webscript de presentación y Webscript de acceso al repositorio.

Webscripts de presentación: Dichos webscripts modifican o extienden la capa de presentación de Alfresco. Pueden definir nuevos componentes en su interfaz de usuario, definir portlets⁵, crear un website dentro de Alfresco, etc.

Webscripts de acceso al repositorio: Son aquellos que consultan, modifican y retornan datos de Alfresco (documentos), ofreciendo una interfaz para consumir dicho servicio por terceros.

A continuación, se muestra el listado de los webscript de acceso al repositorio, desarrollados en la realización del módulo de gestión de nomencladores.

⁵ Los portlets son módulos Web reutilizables que se ejecutan en un servidor del portal y proporcionan acceso al contenido, a las aplicaciones y a otros recursos basados en la web. (IBM, 2018).

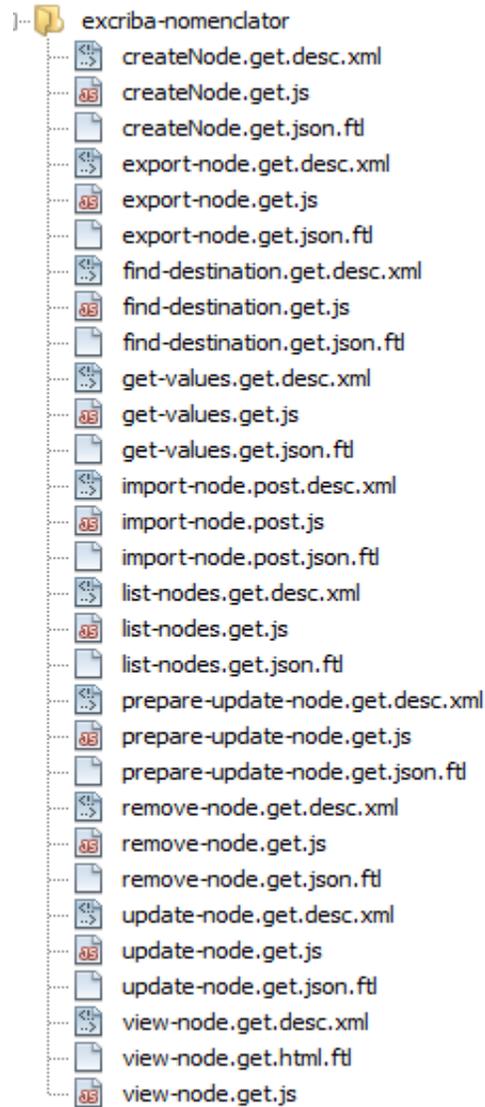


Fig 6: Listado de webscript de acceso al repositorio.

A continuación, se explica el webscript list-nodes, el cual está compuesto de tres archivos al igual que todos los webscript de acceso al repositorio.

list-nodes.get.desc.xml: Archivo de descripción del webscript. El archivo se encuentra en la siguiente dirección dentro del eXcriba /cu/uci/excriba-nomenclator/list-nodes/ list-nodes.get.desc.xml.

```

<webscript>
  <shortname>Lista de Nomencladores </shortname>
  <description>Lista de Nomencladores</description>
  <url>/cu/uci/excriba-nomenclator/list-nodes</url>
  <format default="json">extension</format>
  <authentication>user</authentication>
</webscript>

```

Fig 7: list-nodes.get.desc.xml.

list-nodes.get.js: Archivo donde se desarrolla la lógica del servicio web. El archivo se encuentra en la siguiente dirección dentro del eXcriba /cu/uci/excriba-nomenclator/list-nodes/ list-nodes.get.js

```

function getNodesNomenc() {
  var oferta = search.luceneSearch('PATH:"/app:company_home/Nomenclador/*" +TYPE:"nomenc:Nomenclador");
  return oferta;
}
function getNodesTypesNomenc(){
  var assoc= new Array();
  var nomenclator = search.luceneSearch('PATH:"/app:company_home/Nomenclador/*" +TYPE:"nomenc:Nomenclador");

  for(var i = 0; i < nomenclator.length; i++){
    var values = "";
    if(nomenclator[i].assocs["nomenc:assoc"] != null){
      for (var j = 0; j < nomenclator[i].assocs["nomenc:assoc"].length; j++) {
        if(nomenclator[i].assocs["nomenc:assoc"][j] != null){
          values += "<span style='border-radius:2px; background-color:#ccc; padding:3px 5px;margin-right:7px;display: inline-
        }
      }
      assoc[nomenclator[i].properties['sys:node-uuid']] = values;
    }
  }
  return assoc;
}

function main(){
  model.nodes = getNodesNomenc();
  model.associacions = getNodesTypesNomenc();
}
main();

```

Fig 8: list-nodes.get.js.

list-nodes.get.json.ftl: Archivo donde se define la estructura, de la respuesta del servidor, este puede ser en formato html o json como el fichero actual. Se vincula junto al motor de plantillas FreeMarker. El archivo

se encuentra en la siguiente dirección dentro del eXcriba /cu/uci/excriba-nomenclator/list-nodes/ list-nodes.get.json.ftl

```
{
  "data": [
    <#list nodes as node>
    <#assign actions = "">
    <#assign actions = actions + "<div><a href='#' title='Detalles' class='onShowAction actions'><i class='fa fa-file-text'></i></a></div>">
    <#assign actions = actions + "<div><a href='#' title='Editar' class='onEditAction actions'><i class='fa fa-pencil'></i></a></div>" >
    <#assign actions = actions + "<div><a href='#' title='Eliminar' class='onDeleteAction actions'><i class='fa fa-trash'></i></a></div>" >
    [
      "${node.properties['sys:node-uuid']}?string",
      <#if node.properties['nomenc:name']??>
        "${node.properties['nomenc:name']}?string",
      <#else>
        "-",
      </#if>
      "${node.properties['nomenc:Descripcion']}?string",
      <#if asociacions[node.properties['sys:node-uuid']]??>
        "${asociacions[node.properties['sys:node-uuid']]?string",
      <#else>
        "-",
      </#if>
      "${actions}"
    ]
    <#if node_has_next>,</#if>
  </#list>
  ]}
  <#if ps??>
    ${ps}
  </#if>
}
```

Fig 9: list-nodes.get.json.ftl.

3.3. Despliegue de la solución

A continuación, se muestra el despliegue de la solución del módulo de gestión de nomencladores, donde se expone la estructura de carpetas y archivos con que cuenta el desarrollador del módulo.

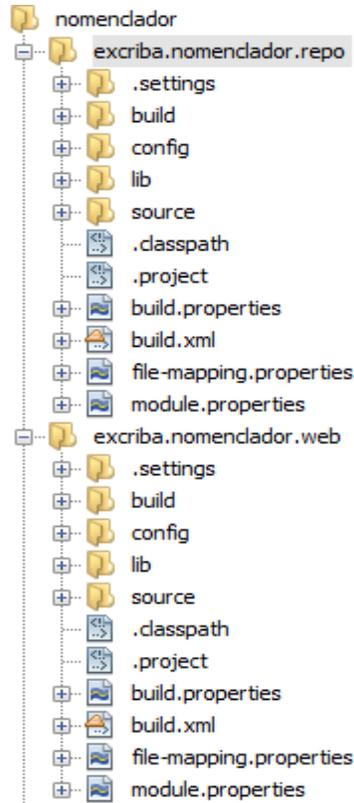


Fig 10: list-nodes.get.json.ftl.

3.4. Interfaz gráfica del módulo

A continuación, se muestra una imagen con la interfaz final del módulo de gestión de nomencladores del GDA Xabal eXcriba 3.1.



Fig 11: Interfaz gráfica del módulo.

3.5. Pruebas de software

Las pruebas de software son un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación (Pressman, 2010). Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de un sistema. Básicamente es una fase en el desarrollo de software que consiste en probar la aplicación construida. Se integran dentro de las diferentes fases del ciclo de la Ingeniería de Software ejecutando un programa que mediante técnicas experimentales trata de descubrir los errores que tiene el sistema.

La metodología de software empleada en esta investigación define tres disciplinas de pruebas: Pruebas internas, Pruebas de liberación y Pruebas de aceptación. Para dicha investigación se desarrollarán las disciplinas: Pruebas internas y Pruebas de aceptación.

3.5.1. Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas (Rodríguez Sánchez, 2015).

Para validar el correcto funcionamiento del módulo desarrollado se establecieron dos etapas de pruebas: Pruebas internas ejecutadas por el desarrollador y Pruebas internas a nivel de centro CIGED.

➤ **Pruebas internas ejecutadas por el desarrollador**

En esta etapa el desarrollador realizó pruebas al módulo mediante las técnicas: Pruebas de caja blanca, haciendo uso del método: Prueba del camino básico y Prueba de caja negra utilizando la técnica de partición de equivalencia. A continuación, se explican dichas pruebas.

Pruebas de caja blanca

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

➤ **Prueba del camino básico**

La prueba de caja blanca realizada a la propuesta de solución fue la prueba del camino básico, a partir del cálculo de la complejidad ciclomática del algoritmo a ser analizado.

Para realizarla se deben enumerar las sentencias de código y a partir de ahí elaborar el grafo de flujo de esta funcionalidad. Se definieron una serie de pasos a seguir:

1. **Notación del grafo de flujo:** usando el código como base se realiza la representación del grafo de flujo, mediante una sencilla notación. Cada construcción estructurada tiene su correspondiente símbolo (Pressman, 2010).
 - **Nodo:** a cada círculo denominado nodo, representa una o más sentencias procedimentales.
 - **Arista:** las flechas del grafo de flujo, denominadas aristas, representan el flujo de control y son análogas a las flechas del diagrama de flujo.
 - **Región:** las áreas delimitadas por aristas y nodos se denominan regiones.
2. **Complejidad ciclomática:** es una métrica que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado define el número de caminos independientes del conjunto básico de un programa. Esto indica el límite superior para el número de pruebas que se deben realizar, para asegurar que se ejecuta cada sentencia al menos una vez. La complejidad ciclomática tiene fundamentos en la teoría de gráficos y proporciona una medición de software extremadamente útil. La complejidad se calcula en una de tres formas:
 - El número de regiones del gráfico de flujo corresponde a la complejidad ciclomática.

Capítulo III

- La complejidad ciclomática $V(G)$ para un gráfico de flujo G se define como $V(G) = A - N + 2$ donde A es el número de aristas del gráfico de flujo y N el número de nodos del gráfico de flujo.
 - La complejidad ciclomática $V(G)$ para un gráfico de flujo G también se define como $V(G) = P + 1$ donde P es el número de nodos predicado contenidos en el gráfico de flujo G (Pressman, 2010).
3. Determinar un conjunto básico de caminos linealmente independientes: el valor de $V(G)$ es el número de caminos linealmente independientes de la estructura de control del programa (Pressman, 2010).
 4. Obtención de casos de prueba: se realizan los casos de pruebas que forzarán la ejecución de cada camino del conjunto básico (Pressman, 2010).

```

function getAssocs(node){
1  var values = [];
2  if(node.assocs["nomenc:assoc"] != null){
3    for (var j = 0; j < node.assocs["nomenc:assoc"].length; j++) {
4      if(node.assocs["nomenc:assoc"][j] != null){
5        values.push(node.assocs["nomenc:assoc"][j].properties['nomenc:name']);
        }
      }
    }
6  return values;
}
  
```

Fig 12: Método `getAssocs(node)` del webscript `view-node.get.js`.

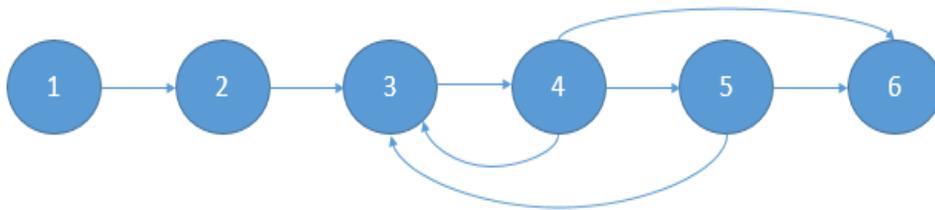


Fig 13: Técnica del camino básico. Elaboración propia.

Capítulo III

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

$$= 8 - 6 + 2$$

$$= 4$$

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

$$= 3 + 1$$

$$V(G) = \text{Regiones}$$

$$V(G) = 4$$

$$V(G) = \text{Complejidad ciclomática}$$

A = Cantidad de aristas del grafo

N = Cantidad de nodos del grafo

P = Nodos predicados. Nodos que tienen más de una arista de salida

Regiones: Áreas delimitadas por nodos

y aristas del grafo

Resultados de la prueba del camino básico

El cálculo efectuado mediante las fórmulas ha dado el mismo valor, dando como resultado 4, lo que indica que existen 4 posibles caminos por donde el flujo puede circular, y determina el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. El cálculo arrojó que $V(G) = 4$, por lo que los posibles caminos básicos son

Camino 1: 1, 2, 3, 4, 5, 6

Camino 2: 1, 2, 3, 4, 3, 4, 5, 6

Camino 3: 1, 2, 3, 4, 3, 4, 5, 3, 4, 5, 6

Camino 4: 1, 2, 3, 4, 3, 4, 6

Caso de prueba para el camino 1

Tabla 6: Diseño de caso de prueba para el camino 1.

Descripción	Permite guardar en la variable values, todas las asociaciones del nodo que se obtiene como parámetro.
Condición de ejecución	El node tiene que tener asociados elementos de tipo "nomenc:assoc" y cada una de las asociaciones son distintas de null.
Entrada	Node
Resultado	Devuelve un arreglo con todas las asociaciones del node entrado como parámetro.

Pruebas de caja negra

Las pruebas de caja negra también denominadas pruebas de comportamiento, se centran en los requisitos funcionales del software. O sea, permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa (Pressman, 2010).

La prueba de caja negra intenta encontrar las siguientes categorías de errores (Pressman, 2010).

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

➤ **Técnica de partición equivalente**

Para desarrollar las pruebas de caja negra es posible aplicar varias técnicas, entre ellas (Jacobson, et al., 2004):

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

La técnica de partición de equivalencia es considerada una de las técnicas más efectivas, pues permite examinar los valores válidos e inválidos de las entradas existentes en el software. Esta técnica se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar. Una clase de equivalencia representa un conjunto de estados válidos o no válidos para condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica (Jacobson, y otros, 2004).

➤ **Casos de prueba**

Para la realización de las pruebas al módulo de gestión de nomencladores se generó el artefacto “Diseño de casos de pruebas basado en casos de uso”.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución, resultados esperados desarrollados para cumplir un objetivo en particular o una función esperada. La entidad más simple que siempre es ejecutada como una unidad, desde el comienzo hasta el final (Ferrer Rodríguez, 2013).

Los casos de pruebas deben verificar (Ferrer Rodríguez, 2013).

- Si el producto satisface los requisitos del usuario, tal y como se describe en las especificaciones.

Capítulo III

- Si el producto se comporta como se desea, tal y como se describe en las especificaciones funcionales del diseño.

Un caso de prueba (CP) se diseña según las funcionalidades descritas en los escenarios. Este es elaborado previo a la realización de las pruebas funcionales de la aplicación. Cada planilla de caso de prueba recoge la especificación de un escenario, dividido en secciones, para hacer más fructífera la ejecución de las pruebas. En esta investigación fueron diseñados 14 casos de pruebas. A continuación, se muestra la SC1 Crear nomenclador del CP Gestionar nomenclador y en el Anexo 5 Casos de prueba, se describen el resto de los casos de pruebas por escenarios.

Tabla 7: Caso de prueba: Gestionar nomenclador SC1 Crear nomenclador.

Escenario	Descripción	Nombre	Descripción	Valores	Respuesta del sistema	Flujo central
EC 1.1 Crear nomenclador	El escenario de prueba permite al usuario autenticado crear un nomenclador.	V	V	V	El sistema crea el nuevo elemento y muestra el mensaje: "Nomenclador creado"	1- Presionar el menú Administración, 2- Seleccionar Administración de nomencladores.3- Seleccionar la funcionalidad "Crear nomenclador".4- Inserta los datos. 5- Presiona el botón "Aceptar"
		Carro	Moderno	Audi, Mercedes		
EC 1.2 Validar campos vacíos	El escenario de prueba permite al usuario validar los campos vacíos.	I	V	V	El sistema señala el campo obligatorio vacío y muestra el mensaje: "El valor no puede estar vacío."	
			Moderno	Audi, Mercedes		
		V	I	V		
	Carro		Audi, Mercedes			
EC 1.3 Campos incorrectos	El escenario de prueba permite al usuario autenticado validar los campos incorrectos.	I	V	V	El sistema muestra el mensaje: "El valor contiene caracteres no admitidos."	
		Carro21%\$	Moderno	Audi, Mercedes		
		V	I	V		
	Carro	#\$es un carro	Audi, Mercedes			
EC 1.4 Cancelar acción	El escenario de prueba permite al usuario autenticado cancelar la acción.	NA	NA	NA	El sistema cancela la acción.	1- Presionar el menú Administración, 2- Seleccionar Administración de nomencladores.3- Seleccionar la funcionalidad "Crear nomenclador".4- Presiona el botón Cancelar

Resultado de las pruebas internas ejecutadas por el desarrollador

Culminadas las pruebas internas ejecutadas por el desarrollador, fue posible comprobar que el módulo cumple con los requisitos funcionales definidos para la investigación. A continuación, se muestra la relación de no conformidades detectadas, resueltas y pendientes por iteración.

Tabla 8: Resultados de las pruebas internas ejecutadas por el desarrollador.

Iteración	NC Detectadas	Asociadas a	NC Resueltas	NC Pendientes
I	20	Interfaz Ortografía Funcionalidad	15	5
II	10 + 5	Interfaz Funcionalidad	13	2
III	4 + 2	Interfaz Ortografía Funcionalidad	6	0

Como se puede apreciar en la tabla anterior se realizaron 3 iteraciones de pruebas ejecutadas por el desarrollador. A lo largo de la primera iteración se detectaron 20 no conformidades: 17 asociadas a errores de interfaz y de funcionalidad y 3 a errores ortográficos. De estas no conformidades fueron resueltas 15 en dicha iteración, quedando 5 pendientes para la segunda iteración. En la segunda iteración fueron detectadas 10 no conformidades, las cuales, junto a las 5 pendientes de la iteración anterior, sumaron un total de 15 no conformidades. Dichas no conformidades estaban asociadas a errores de interfaz y de funcionalidad, siendo satisfactoriamente solucionadas 13, quedando pendiente 2 no conformidades para una tercera iteración. En una tercera iteración se detectaron 4 no conformidades, las cuales junto a las 2 pendientes de la iteración anterior sumaron 6 no conformidades a resolver, siendo satisfactoriamente resueltas en dicha iteración.

- **Pruebas internas a nivel de centro CIGED**

En esta etapa, especialistas del centro CIGED realizaron pruebas funcionales al módulo desarrollado, haciendo uso de los artefactos de pruebas diseños de Casos de pruebas.

Resultado de las pruebas internas a nivel de centro CIGED

Con la culminación de la disciplina Pruebas internas, fue posible comprobar que el módulo cumple con los requisitos funcionales definidos para la investigación. A continuación, se muestra la relación de no conformidades detectadas, resueltas y pendientes por iteración, para las pruebas desarrolladas al módulo por el centro CIGED.

Tabla 9: Resultados de las pruebas internas a nivel de centro CIGED.

Iteración	NC Detectadas	Asociadas a	NC Resueltas	NC Pendientes
I	25	Interfaz Ortografía Funcionalidad	19	6
II	13 + 6	Interfaz Funcionalidad	17	2
III	7 + 2	Interfaz Ortografía Funcionalidad	8	1
IV	1 + 1	Validación Funcionalidad	1	1

Como se puede apreciar en la tabla anterior se realizaron 4 iteraciones de pruebas. A lo largo de la primera iteración se detectaron 25 no conformidades: 23 asociadas a errores de interfaz y de funcionalidad y 2 a errores ortográficos. De estas no conformidades fueron resueltas 19 en dicha iteración, quedando 6 pendientes para la segunda iteración. En la segunda iteración fueron detectadas 13 no conformidades, las cuales, junto a las 6 pendientes de la iteración anterior, sumaron un total de 19 no conformidades. Dichas no conformidades estaban asociadas a errores de interfaz y de funcionalidad, siendo satisfactoriamente solucionadas 17, quedando pendiente 2 no conformidades para una tercera iteración. En una tercera iteración se detectaron 7 no conformidades, las cuales junto a las 2 pendientes de la iteración anterior sumaron 9 no conformidades a resolver, siendo resueltas 8 y quedando 1 pendiente para otra iteración. En una cuarta y última iteración se detectó 1 no conformidad, que junto a la pendiente de la iteración anterior sumaban 2 no conformidades. De estas no conformidades se resolvió la asociada a funcionalidad y quedó pendiente, la asociada a validación.

Luego de concluida dicha disciplina, se obtuvo por el centro CIGED, un Acta de evaluación interna de producto de software, ver Anexo 6 Acta de evaluación interna de producto de software.

3.5.2. Pruebas de aceptación

Para dar cumplimiento a esta disciplina, se desplegó un servidor piloto de prueba en el centro CIGED para realizar una prueba final antes del despliegue del sistema. El objetivo del servidor piloto era verificar que el software estaba listo y que podía ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido. Como resultado de dicha prueba, se obtuvo un acta de aceptación por el cliente (centro CIGED) donde quedó plasmada su conformidad con el módulo, ver Anexo 7 Acta de aceptación del módulo.

Conclusiones parciales

- La definición del modelo de contenido y el desarrollo de los webscripts contribuyeron en la construcción del módulo de gestión de nomencladores para el GDA XABAL eXcriba 3.1.
- Las pruebas internas realizadas permitieron corregir errores detectados durante la fase de construcción del software, logrando la correcta implementación de los requisitos funcionales.

Conclusiones generales

Durante el desarrollo de la investigación se planteó la necesidad de desarrollar un módulo para la gestión de nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1, dándole así cumplimiento al objetivo propuesto de la siguiente forma:

- Durante el desarrollo de esta investigación se evidenció la necesidad de un módulo de gestión de nomencladores que permita la gestión de la información, poco variable en el tiempo dándole funcionalidades al GDA eXcriba 3.1 que contribuyan con su uso en empresas e instituciones.
- El estudio de los referentes teóricos permitió identificar las características principales del módulo, propiciando un mejor entendimiento del diseño de la propuesta de solución.
- La implementación del módulo de nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1 contribuye a la gestión y modificación de la información poco variable en el tiempo de las diferentes empresas y organizaciones.
- Las pruebas aplicadas a la propuesta de solución propiciaron una correcta validación de las funcionalidades y el diseño del módulo de nomencladores en el Gestor de Documentos Administrativos Xabal eXcriba 3.1.

Recomendaciones

Para darle continuidad a la presente investigación el autor recomienda:

- Reimplementar la funcionalidad importar nomenclador, de forma tal que permita cargar un documento con los nomencladores.
- Añadir la posibilidad de establecer asociaciones entre los nomencladores.

Referencias Bibliográficas

- 2008.** [En línea] 11 de 12 de 2008. [Citado el: 11 de 1 de 2018.] https://www.google.com/cu/?gws_rd=ssl#q=nomencladores+definicion&revid=784233885..
- Freemarker. 2013.** FreeMarker: java template . *FreeMarker: java template* . [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://freemarker.org/>.
- Torres Hernández, Oscar Daniel. 2013.** *Propuesta arquitectónica para el núcleo del Gestor de Documentos Administrativos eXcriba 3.0*. La Habana : s.n., 2013.
- Alfresco. 2013.** Alfresco. *Alfresco*. [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://wiki.alfresco.com/wiki/>.
- . **2018.** Alfresco. [En línea] 2018. [Citado el: 4 de 6 de 2018.] <https://www.alfresco.com/>.
- . **2018.** Alfresco Documetation. [En línea] 2018. [Citado el: 4 de 6 de 2018.] <https://docs.alfresco.com/>.
- Alonso Amo, Fernando, Martínez Normand, Loïc y Segovia Pérez, Francisco Javier Javier. 2005.** *Introducción a la ingeniería del software*. Madrid : s.n., 2005.
- Alvarez, Miguel Angel. 2013.** Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. [En línea] 2013. [Citado el: 5 de 4 de 2018.] <http://www.desarrolloweb.com/articulos/1178.php..>
- Andalucía, Marco de Desarrollo de la Junta de.** *API de JavaScript.[PDF]*.
- autores, Colectivo de. 2015.** *Xabal eXcriba Gestor de Documentos Administrativos*. La Habana : s.n., 2015.
- Beust y Cedric. 2002.** *Profesional programación Java Server con J2EE*. 2002. Edición 1.3.
- Bradenbaugh, Jerry. 2000.** *Aplicaciones JavaScript*. Madrid : Anaya, 2000. ISBN:84-41 5-1 070-9.
- del Valle Rojo, Silvana. 2013.** *ELICITACION Y ESPECIFICACION DE REQUERIMIENTOS NO FUNCIONALES EN APLICACIONES WEB*. Universidad Nacional de La Plata. 2013. Tesis de Maestría.
- Delgado, Lic. Mercedes Moreira. 2006.** La organización de la información para la gestión del conocimiento de las empresas. [En línea] 2006. [Citado el: 2 de 3 de 2018.] <http://www.gestiopolis1.com/recursos7/Docs/ger/organizacion-de-lainformacion-para-la-gestion-del-conocimiento.htm..>
- 2006.** Diagrama de despliegue. 2006.
- Diccionario. 2017.** Diccionario. [En línea] 2017. [Citado el: 2 de 3 de 2018.] <http://dle.rae.es>.
- Dobson, Michael y Luttrell, Joe. 2009.** LESSONS FROM HISTORY. *LESSONS FROM HISTORY*. [En línea] 5 de 5 de 2009. [Citado el: 18 de 2 de 2018.] <http://www.lessons-from-history.com/node/83>.

- Eguíluz Pérez, Javier.** librosweb.es. *librosweb.es*. [En línea] [Citado el: 8 de 12 de 2017.] <http://www.librosweb.es/ajax/capitulo1.html>.
- Escalona, José María. 2002.** *Ingeniería de Requisitos en Aplicaciones para la Web*. Sevilla : s.n., 2002.
- Faudón, Ruiz y Luis María, Sergio. 2001.** *Introducción a los sistemas*. 2001.
- Ferrer Rodríguez, Orquidia. 2013.** *Sistema de Gestión de la información para las Ópticas de Cuba*. La Habana : s.n., 2013.
- Gamma, Erich. 1994.** *Design patterns: elements of reusable object-oriented software*. s.l. : Pearson Education, 1994.
- Garlan, David y Shaw, Mary. 1994.** *An Introduction to Software Architecture*. Carnegie Mellon University : s.n., 1994. 9810215940..
- Gauchat y Diego, Juan. 2012.** *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona, España : s.n., 2012.
- Gestión, Definición de. 2017.** Definición de Gestión. *Definición de Gestión*. [En línea] 2017. <http://conceptodefinicion.de/gestion/>.
- Martínez, Juan y Francisco, Javier.** *Guía de construcción de software en java con patrones de diseño*. Oviedo : s.n.
- Hernández, Karen. 2015.** *La información periodística*. Venezuela : s.n., 2015.
- 2014.** Herramientas de diseño. *Herramientas de diseño*. [En línea] 2014. [Citado el: 30 de 5 de 2018.] <http://multimedia.uoc.edu/blogs/labeines/es/prototipat/axure-disseny-de-wireframes-i-prototips/>.
- IBM Corp. 2006.** Artefacto: Modelo de despliegue. *Artefacto: Modelo de despliegue*. [En línea] 2006. [Citado el: 21 de 5 de 2018.] https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_deployment_model_el_57DF1DF5.html.
- IBM, Representante. 2018.** IBM Knowledge Center. [En línea] 2018. [Citado el: 21 de 5 de 2018.] https://www.ibm.com/support/knowledgecenter/es/SSYJ99_8.5.0/dev-portlet/wpsbpc.html.
- INNOCUO. 2006.** Patrones de diseño, la importancia de aprenderlos. [En línea] 2006. [Citado el: 4 de 6 de 2018.] [http://blog.innocuo.com/2006/09/patrones-de-diseno-la-importancia-de-aprenderlos/..](http://blog.innocuo.com/2006/09/patrones-de-diseno-la-importancia-de-aprenderlos/)
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004.
- José. 2014.** GaussWebApp. [En línea] 2014. [Citado el: 21 de 5 de 2018.] <https://gausswebapp.com/arquitectura-rest.html>.
- jQuery. 2012.** jQuery. *jQuery*. [En línea] 2012. [Citado el: 7 de 12 de 2017.] [http://jquery.com/..](http://jquery.com/)
- Json. 2013.** Introducing Json. *Introducing Json*. [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://www.json.org/json-es.html>.

- L. BASS, P. y CLEMENTS, R. K. 2003.** *Software Architecture in Practice*. New York : Addison Wesley, 2003.
- Larman, Craig. 2003.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Madrid : Pearson Educación, 2003. 2da Edición.
- 2018.** Los Teatinos. *Los Teatinos*. [En línea] 2018. [Citado el: 30 de 5 de 2018.] <http://www.losteatinos.es/servlets/servlet.html>.
- Lucene, Apache.** Apache Software Foundation. *Apache Software Foundation*. [En línea] [Citado el: 4 de 6 de 2018.] <http://apachefoundation.wikispaces.com/Apache+Lucene..>
- Malbrán, Francisco Gabriel y Trouillet, Germán Eduardo. 2008.** *Nomenclador Cartográfico Para Personas con Discapacidad Visual*. 2008.
- Marín, Dr. Miguel Eusebio. 2009.** *Informática en Salud*. [En línea] 2009. [Citado el: 14 de 5 de 2018.] http://informatica2009.sld.cu/Members/marin/nomencladores-medicos-nacionales-para-la-informatizacion-de-la-atencion-medica-en-el-sistema-nacional-de-salud/at_download/trabajo.
- Moreno, Renán Vázquez y Mojena Alpizar, José. 2011.** *Sistema para la gestión de nomencladores*. La Habana : s.n., 2011.
- Mosquera, Alexander , Rincón, Derlisiret y Gracia Romero, María. 2017.** *La organización basada en los sistemas de información*. Maracaibo : s.n., 2017.
- O'Reilly Media, Inc. 2014.** From the inside out. [En línea] 23 de 7 de 2014. [Citado el: 21 de 2 de 2018.] <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58..>
- Paradigm, Visual. 2018.** UML CASE tool for software development. *UML CASE tool for software development*. [En línea] 2018. [Citado el: 31 de 5 de 2018.] <https://www.visual-paradigm.com/>.
- Planeación, Departamento Nacional de. 2013.** *GUÍA DE ELABORACIÓN DE MODELOS CONCEPTUALES*. Bogotá D.C : s.n., 2013.
- Porto, Julián Pérez y Merino, María. 2008.** *Definicion.de*. [En línea] 2008. [Citado el: 7 de 11 de 2017.] <https://definicion.de/gestion/>.
- Pressman, Roger. 2010.** *Software Engineering. A practitioner's approach*. Nueva York : s.n., 2010. Vol. 6. 7ma Edición.
- Restrepo-Arango, Cristina. 2016.** Red de Egresados en Ciencia de la Información. [En línea] 2016. [Citado el: 5 de 4 de 2018.] <http://micros.nomono.co/redegresados/2016/08/07/por-que-es-importante-la-organizacion-de-la-informacion-en-las-bibliotecas/>.
- Rodríguez Sánchez, Tamara . 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
- Rouse, Margaret. 2018.** TechTarget. *TechTarget*. [En línea] 2018. [Citado el: 29 de 5 de 2018.] <https://searchmicroservices.techtarget.com/definicion/data-dictionary>.

Salinas Revelles, Jesús. *Alfresco ECM 4.2 Tipos de contenidos personalizados.*

Salud, New Lici. New Lici Salud. *New Lici Salud.* [En línea] [Citado el: 14 de 5 de 2018.]
[http://www.obrasociales.com.ar/Prod_productos.asp?producto=Nomencladores.](http://www.obrasociales.com.ar/Prod_productos.asp?producto=Nomencladores)

Sevilla, José. 2016. *Overti.* Madrid : s.n., 2016.

Share, Alfresco. 2016. KeenSoft. *KeenSoft.* [En línea] 2016. [Citado el: 4 de 6 de 2018.]
[http://www.keensoft.es/alfresco-share-%c2%b7-secuencia-de-eventos-en-la-creacion-de-un-documento/.](http://www.keensoft.es/alfresco-share-%c2%b7-secuencia-de-eventos-en-la-creacion-de-un-documento/)

Shariff y Munwar. 2013. *Alfresco 4 Enterprise Content Management Implementation.* 2013.

Sommerville, Ian. 2005. *Ingeniería del software.* [ed.] Pearson Educación. Madrid : 7 ma edición, 2005.
9788478290741.

Tomcat, Apache. 2014. Apache Tomcat. *Apache Tomcat.* [En línea] 2014. [Citado el: 8 de 12 de 2017.]
[http://tomcat.apache.org/.](http://tomcat.apache.org/)

UCI, Portal. Portal UCI. *Portal UCI.* [En línea] [Citado el: 7 de 12 de 2017.] [http://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba-31.](http://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba-31)

WELLS, D. The Rules of Extreme Programming. Extreme Programing. *The Rules of Extreme Programming.*
Extreme Programing. [En línea] [Citado el: 4 de 6 de 2018.]
<http://www.extremeprogramming.org/rules.html..>

2018. What is PostgreSQL? *What is PostgreSQL?* [En línea] 2018. [Citado el: 30 de 5 de 2018.]
[http://www.postgresqltutorial.com/what-is-postgresql/.](http://www.postgresqltutorial.com/what-is-postgresql/)

Bibliografía

- 2008.** [En línea] 11 de 12 de 2008. [Citado el: 11 de 1 de 2018.] https://www.google.com/cu/?gws_rd=ssl#q=nomencladores+definicion&revid=784233885..
- Freemarker. 2013.** FreeMarker: java template . *FreeMarker: java template* . [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://freemarker.org/>.
- Torres Hernández, Oscar Daniel. 2013.** *Propuesta arquitectónica para el núcleo del Gestor de Documentos Administrativos eXcriba 3.0*. La Habana : s.n., 2013.
- Alfresco. 2013.** Alfresco. *Alfresco*. [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://wiki.alfresco.com/wiki/>.
- . **2018.** Alfresco. [En línea] 2018. [Citado el: 4 de 6 de 2018.] <https://www.alfresco.com/>.
- . **2018.** Alfresco Documetation. [En línea] 2018. [Citado el: 4 de 6 de 2018.] <https://docs.alfresco.com/>.
- Alonso Amo, Fernando, Martínez Normand, Loïc y Segovia Pérez, Francisco Javier Javier. 2005.** *Introducción a la ingeniería del software*. Madrid : s.n., 2005.
- Alvarez, Miguel Angel. 2013.** Editor web orientado a la programación de páginas PHP, con ayudas en la gestión de proyectos y depuración de código. [En línea] 2013. [Citado el: 5 de 4 de 2018.] <http://www.desarrolloweb.com/articulos/1178.php..>
- Andalucía, Marco de Desarrollo de la Junta de.** *API de JavaScript.[PDF]*.
- autores, Colectivo de. 2015.** *Xabal eXcriba Gestor de Documentos Administrativos*. La Habana : s.n., 2015.
- Beust y Cedric. 2002.** *Profesional programación Java Server con J2EE*. 2002. Edición 1.3.
- Bradenbaugh, Jerry. 2000.** *Aplicaciones JavaScript*. Madrid : Anaya, 2000. ISBN:84-41 5-1 070-9.
- del Valle Rojo, Silvana. 2013.** *ELICITACION Y ESPECIFICACION DE REQUERIMIENTOS NO FUNCIONALES EN APLICACIONES WEB*. Universidad Nacional de La Plata. 2013. Tesis de Maestría.
- Delgado, Lic. Mercedes Moreira. 2006.** La organización de la información para la gestión del conocimiento de las empresas. [En línea] 2006. [Citado el: 2 de 3 de 2018.] <http://www.gestiopolis1.com/recursos7/Docs/ger/organizacion-de-lainformacion-para-la-gestion-del-conocimiento.htm..>
- 2006.** Diagrama de despliegue. 2006.
- Diccionario. 2017.** Diccionario. [En línea] 2017. [Citado el: 2 de 3 de 2018.] <http://dle.rae.es>.
- Dobson, Michael y Luttrell, Joe. 2009.** LESSONS FROM HISTORY. *LESSONS FROM HISTORY*. [En línea] 5 de 5 de 2009. [Citado el: 18 de 2 de 2018.] <http://www.lessons-from-history.com/node/83>.

- Eguíluz Pérez, Javier.** librosweb.es. *librosweb.es*. [En línea] [Citado el: 8 de 12 de 2017.] <http://www.librosweb.es/ajax/capitulo1.html>.
- Escalona, José María. 2002.** *Ingeniería de Requisitos en Aplicaciones para la Web*. Sevilla : s.n., 2002.
- Faudón, Ruiz y Luis María, Sergio. 2001.** *Introducción a los sistemas*. 2001.
- Ferrer Rodríguez, Orquidia. 2013.** *Sistema de Gestión de la información para las Ópticas de Cuba*. La Habana : s.n., 2013.
- Gamma, Erich. 1994.** *Design patterns: elements of reusable object-oriented software*. s.l. : Pearson Education, 1994.
- Garlan, David y Shaw, Mary. 1994.** *An Introduction to Software Architecture*. Carnegie Mellon University : s.n., 1994. 9810215940..
- Gauchat y Diego, Juan. 2012.** *El gran libro de HTML5, CSS3 y JavaScript*. Barcelona, España : s.n., 2012.
- Gestión, Definición de. 2017.** Definición de Gestión. *Definición de Gestión*. [En línea] 2017. <http://conceptodefinicion.de/gestion/>.
- Martínez, Juan y Francisco, Javier.** *Guía de construcción de software en java con patrones de diseño*. Oviedo : s.n.
- Hernández, Karen. 2015.** *La información periodística*. Venezuela : s.n., 2015.
- 2014.** Herramientas de diseño. *Herramientas de diseño*. [En línea] 2014. [Citado el: 30 de 5 de 2018.] <http://multimedia.uoc.edu/blogs/labeines/es/prototipat/axure-disseny-de-wireframes-i-prototips/>.
- IBM Corp. 2006.** Artefacto: Modelo de despliegue. *Artefacto: Modelo de despliegue*. [En línea] 2006. [Citado el: 21 de 5 de 2018.] https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_deployment_model_el_57DF1DF5.html.
- IBM, Representante. 2018.** IBM Knowledge Center. [En línea] 2018. [Citado el: 21 de 5 de 2018.] https://www.ibm.com/support/knowledgecenter/es/SSYJ99_8.5.0/dev-portlet/wpsbpc.html.
- INNOCUO. 2006.** Patrones de diseño, la importancia de aprenderlos. [En línea] 2006. [Citado el: 4 de 6 de 2018.] [http://blog.innocuo.com/2006/09/patrones-de-diseno-la-importancia-de-aprenderlos/..](http://blog.innocuo.com/2006/09/patrones-de-diseno-la-importancia-de-aprenderlos/)
- Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2004.** *El Proceso Unificado de Desarrollo de Software*. La Habana : Félix Varela, 2004.
- José. 2014.** GaussWebApp. [En línea] 2014. [Citado el: 21 de 5 de 2018.] <https://gausswebapp.com/arquitectura-rest.html>.
- jQuery. 2012.** jQuery. *jQuery*. [En línea] 2012. [Citado el: 7 de 12 de 2017.] [http://jquery.com/..](http://jquery.com/)
- Json. 2013.** Introducing Json. *Introducing Json*. [En línea] 2013. [Citado el: 7 de 12 de 2017.] <http://www.json.org/json-es.html>.

- L. BASS, P. y CLEMENTS, R. K. 2003.** *Software Architecture in Practice*. New York : Addison Wesley, 2003.
- Larman, Craig. 2003.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Madrid : Pearson Educación, 2003. 2da Edición.
- 2018.** Los Teatinos. *Los Teatinos*. [En línea] 2018. [Citado el: 30 de 5 de 2018.] <http://www.losteatinos.es/servlets/servlet.html>.
- Lucene, Apache.** Apache Software Foundation. *Apache Software Foundation*. [En línea] [Citado el: 4 de 6 de 2018.] <http://apachefoundation.wikispaces.com/Apache+Lucene..>
- Malbrán, Francisco Gabriel y Trouillet, Germán Eduardo. 2008.** *Nomenclador Cartográfico Para Personas con Discapacidad Visual*. 2008.
- Marín, Dr. Miguel Eusebio. 2009.** *Informática en Salud*. [En línea] 2009. [Citado el: 14 de 5 de 2018.] http://informatica2009.sld.cu/Members/marin/nomencladores-medicos-nacionales-para-la-informatizacion-de-la-atencion-medica-en-el-sistema-nacional-de-salud/at_download/trabajo.
- Moreno, Renán Vázquez y Mojena Alpizar, José. 2011.** *Sistema para la gestión de nomencladores*. La Habana : s.n., 2011.
- Mosquera, Alexander , Rincón, Derlisiret y Gracia Romero, María. 2017.** *La organización basada en los sistemas de información*. Maracaibo : s.n., 2017.
- O'Reilly Media, Inc. 2014.** From the inside out. [En línea] 23 de 7 de 2014. [Citado el: 21 de 2 de 2018.] <http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58..>
- Paradigm, Visual. 2018.** UML CASE tool for software development. *UML CASE tool for software development*. [En línea] 2018. [Citado el: 31 de 5 de 2018.] <https://www.visual-paradigm.com/>.
- Planeación, Departamento Nacional de. 2013.** *GUÍA DE ELABORACIÓN DE MODELOS CONCEPTUALES*. Bogotá D.C : s.n., 2013.
- Porto, Julián Pérez y Merino, María. 2008.** *Definicion.de*. [En línea] 2008. [Citado el: 7 de 11 de 2017.] <https://definicion.de/gestion/>.
- Pressman, Roger. 2010.** *Software Engineering. A practitioner's approach*. Nueva York : s.n., 2010. Vol. 6. 7ma Edición.
- Restrepo-Arango, Cristina. 2016.** Red de Egresados en Ciencia de la Información. [En línea] 2016. [Citado el: 5 de 4 de 2018.] <http://micros.nomono.co/redegresados/2016/08/07/por-que-es-importante-la-organizacion-de-la-informacion-en-las-bibliotecas/>.
- Rodríguez Sánchez, Tamara . 2015.** *Metodología de desarrollo para la Actividad productiva de la UCI*. La Habana : s.n., 2015.
- Rouse, Margaret. 2018.** TechTarget. *TechTarget*. [En línea] 2018. [Citado el: 29 de 5 de 2018.] <https://searchmicroservices.techtarget.com/definicion/data-dictionary>.

- Salinas Revelles, Jesús.** *Alfresco ECM 4.2 Tipos de contenidos personalizados.*
- Salud, New Lici.** New Lici Salud. *New Lici Salud.* [En línea] [Citado el: 14 de 5 de 2018.]
[http://www.obrasociales.com.ar/Prod_productos.asp?producto=Nomencladores.](http://www.obrasociales.com.ar/Prod_productos.asp?producto=Nomencladores)
- Sanchez, Alvaro Fontela.** ¿Que es Bootstrap? [online]. [En línea] [Citado el: 7 de 12 de 2017.]
[http://openwebcms.es/2013/quees-bootstrap/.](http://openwebcms.es/2013/quees-bootstrap/)
- Sevilla, José.** 2016. *Overti.* Madrid : s.n., 2016.
- Share, Alfresco.** 2016. KeenSoft. *KeenSoft.* [En línea] 2016. [Citado el: 4 de 6 de 2018.]
[http://www.keensoft.es/alfresco-share-%c2%b7-secuencia-de-eventos-en-la-creacion-de-un-documento/.](http://www.keensoft.es/alfresco-share-%c2%b7-secuencia-de-eventos-en-la-creacion-de-un-documento/)
- Shariff y Munwar.** 2013. *Alfresco 4 Enterprise Content Management Implementation.* 2013.
- Sommerville, Ian.** 2005. *Ingeniería del software.* [ed.] Pearson Educación. Madrid : 7 ma edición, 2005.
9788478290741.
- Tomcat, Apache.** 2014. Apache Tomcat. *Apache Tomcat.* [En línea] 2014. [Citado el: 8 de 12 de 2017.]
[http://tomcat.apache.org/.](http://tomcat.apache.org/)
- UCI, Portal.** Portal UCI. *Portal UCI.* [En línea] [Citado el: 7 de 12 de 2017.] [http://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba-31.](http://www.uci.cu/investigacion-y-desarrollo/productos/xabal/excriba-31)
- WELLS, D.** The Rules of Extreme Programming. Extreme Programing. *The Rules of Extreme Programming. Extreme Programing.* [En línea] [Citado el: 4 de 6 de 2018.]
<http://www.extremeprogramming.org/rules.html..>
- 2018.** What is PostgreSQL? *What is PostgreSQL?* [En línea] 2018. [Citado el: 30 de 5 de 2018.]
[http://www.postgresqltutorial.com/what-is-postgresql/.](http://www.postgresqltutorial.com/what-is-postgresql/)

Anexos

Anexo 1. Entrevista con el cliente

Entrevista

Nombre del entrevistado: Solangel Rodríguez

Cargo en CIGED: Jefe de Departamento de Aplicaciones

Preguntas:

1. ¿Qué problemas debe solucionar el módulo de Gestión de nomencladores?
2. ¿Quiénes serán los usuarios que gestionarán el módulo?
3. ¿Cuáles son las expectativas de usabilidad del módulo?
4. ¿Cuáles serían las propiedades con que deberá contar el producto?
5. ¿Cómo valoraría que la solución ha sido un éxito?

Respuestas:

1. El módulo de gestión de nomencladores solucionaría varios problemas, tales como:
 - Reimplementación de requisitos que contiene información variable en el tiempo.
 - Empleo de desarrolladores para dar soporte a personalizaciones, en temas de gestión de la información variable en el tiempo.
2. El módulo debe ser gestionado por usuarios con privilegios de administración en el GDA eXcriba 3.1, que tengan conocimientos básicos en programación y gestión.
3. Se espera que el módulo cuente con una interfaz bien definida y entendible para el usuario que lo gestionara.
4. El módulo deberá contar con varias propiedades, tales como:
 - Acceso desde un navegador web.
 - Las herramientas seleccionadas para el desarrollo del módulo deben estar respaldadas por licencias libres.
 - El módulo debe contar con una interfaz gráfica bien definida.
5. Luego de concluida la disciplina de Pruebas internas, se debe implantar un piloto del módulo para ser probado por el centro y se emitiría un acta de aceptación, siempre que cumpla con la solución de los problemas planteados anteriormente.

Fig 14: Entrevista con el cliente.

Anexo 2. Acta de aceptación de requisitos



UCI Universidad
de las Ciencias
Informáticas

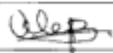
Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración con el centro CIGED** y en función de la ejecución del proyecto de tesis: **Módulo de Gestión de nomencladores para el GDA XABAL eXcriba 3.1**, se hace entrega del producto que se relaciona a continuación:

- Especificación de Requisitos de Software.

La Parte Cliente, luego de haber revisado el producto de trabajo determina su aceptación.

Entrega Tesista	Recibe Centro CIGED
Nombre y apellidos: Alejandro Pirez	Nombre y apellidos: Solangel Rodríguez
Cargo: Desarrollador 	Cargo: Jefe de Dpto. Aplicaciones 

Fecha: 15/12/2017



Fig 15: Acta de aceptación de requisitos de software.

Anexo 3. Casos de uso del sistema

Tabla 10: Descripción del caso de uso: Listar nomenclador.

Caso de Uso	Listar nomenclador	
Objetivo	Listar todos los nomencladores creados o modificados previamente en el sistema.	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el sistema lista todos los nomencladores.	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe existir al menos un nomenclador.	
Postcondiciones	Permite listar todos los nomencladores.	
Flujo de eventos		
Flujo básico Listar nomenclador		
	Actor	Sistema
1.	Se accede a la pestaña administración y dentro accede a "Administración de nomencladores".	

2.		El sistema realiza una búsqueda en su repositorio de todos los nodos de tipo nomenclador.																					
3.		El sistema lista en una tabla los siguientes metadatos: Nomenclador, Descripción, Valores y Acciones.																					
4.		Termina el caso de uso																					
Relaciones	CU incluidos	N/A																					
	CU extendidos	N/A																					
Requisitos no funcionales	N/A																						
Asuntos pendientes	N/A																						
Prototipo elemental de interfaz gráfica de usuario																							
	<table border="1"> <thead> <tr> <th>Nombre</th> <th>Descripción</th> <th>Valores</th> <th>Acciones</th> </tr> </thead> <tbody> <tr> <td>Tipo de documento</td> <td>Representa las diferentes tipologías identificadas en la empresa en cuestión.</td> <td>Carta, Instrucción, Procedimiento, DHL, Indicación</td> <td>  </td> </tr> <tr> <td>Procedencia</td> <td>Contiene los orígenes de los documentos recibidos en la entidad. Este nomenclador no solo tiene los orígenes externos, sino también los internos.</td> <td>MES, ONEI, ANTEX, MINDUS, MINSAP</td> <td>  </td> </tr> <tr> <td>Destino</td> <td>Contiene los destinos de los documentos recibidos en la entidad. Este nomenclador no solo tiene los destinos externos, sino también los internos.</td> <td>PI, UH, UCI, IPVC, ECOAIND, UCI, IPUEC</td> <td>  </td> </tr> <tr> <td>Sexo</td> <td>Contiene los diferentes valores del nomenclador.</td> <td>Femenino, Masculino, Otro</td> <td>  </td> </tr> </tbody> </table>			Nombre	Descripción	Valores	Acciones	Tipo de documento	Representa las diferentes tipologías identificadas en la empresa en cuestión.	Carta, Instrucción, Procedimiento, DHL, Indicación	  	Procedencia	Contiene los orígenes de los documentos recibidos en la entidad. Este nomenclador no solo tiene los orígenes externos, sino también los internos.	MES, ONEI, ANTEX, MINDUS, MINSAP	  	Destino	Contiene los destinos de los documentos recibidos en la entidad. Este nomenclador no solo tiene los destinos externos, sino también los internos.	PI, UH, UCI, IPVC, ECOAIND, UCI, IPUEC	  	Sexo	Contiene los diferentes valores del nomenclador.	Femenino, Masculino, Otro	  
Nombre	Descripción	Valores	Acciones																				
Tipo de documento	Representa las diferentes tipologías identificadas en la empresa en cuestión.	Carta, Instrucción, Procedimiento, DHL, Indicación	  																				
Procedencia	Contiene los orígenes de los documentos recibidos en la entidad. Este nomenclador no solo tiene los orígenes externos, sino también los internos.	MES, ONEI, ANTEX, MINDUS, MINSAP	  																				
Destino	Contiene los destinos de los documentos recibidos en la entidad. Este nomenclador no solo tiene los destinos externos, sino también los internos.	PI, UH, UCI, IPVC, ECOAIND, UCI, IPUEC	  																				
Sexo	Contiene los diferentes valores del nomenclador.	Femenino, Masculino, Otro	  																				

Tabla 11: Descripción del caso de uso: Buscar nomenclador

Objetivo	Buscar nomenclador.	
Actores	Administrador	
Resumen	El caso de uso inicia cuando el usuario presiona el input de texto “Buscar”	
Complejidad	Media	
Prioridad	Media	
Precondiciones	Debe existir al menos un nomenclador.	
Postcondiciones	Se buscan los nomencladores que coincidan con la búsqueda.	
Flujo de eventos		
Flujo básico Buscar nomenclador		
	Actor	Sistema
1.	Introduce un término de búsqueda.	
2.		El sistema muestra todos los nomencladores que coincidan con el término de búsqueda.
3.		Termina el caso de uso.
Flujos alternos		
2.a Evento Entrada incorrecta de los datos		
	Actor	Sistema
1.		El sistema muestra el mensaje “No se encontraron resultados.”.
Relaciones	CU incluidos	N/A
	CU extendidos	N/A
Requisitos no funcionales	N/A	

Asuntos pendientes	N/A
Prototipo elemental de interfaz gráfica de usuario	
Buscar: <input type="text"/>	

Anexo 4. Matrices de trazabilidad

Tabla 12: Matriz de trazabilidad de RF-Concepto.

		(3) Class		
By: Transitor		Nomenclador	Valor	Administrador
(16) Requirement <div style="float: right; text-align: right;"> ↕ ↕ ↕ </div>		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF1. Crear nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF2. Editar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF3. Ver detalles del nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF4. Eliminar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF5. Listar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF6. Buscar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF7. Importar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF8. Exportar nomenclador	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF9. Eliminar valor	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF10. Ordenar nomencladores por la columna nombre	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF11. Ordenar nomencladores por la columna descripción	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF12. Ordenar nomencladores por la columna valores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF13. Configurar cantidad de elementos a visualizar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF14. Filtrar nomenclador por nombre	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF15. Filtrar nomenclador por descripción	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	RF16. Filtrar nomenclador por valores	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Anexos

Tabla 13: Matriz de trazabilidad de RF-RF.

(16) Requirement	RF1	RF2	RF3	RF4	RF5	RF6	RF7	RF8	RF9	RF10	RF11	RF12	RF13	RF14	RF15	RF16
RF1									✓							
RF2					✓				✓							
RF3					✓											
RF4					✓											
RF5		✓	✓	✓												
RF6																
RF7																
RF8																
RF9	✓	✓														
RF10																
RF11																
RF12																
RF13																
RF14																
RF15																
RF16																

Tabla 14: Matriz de trazabilidad de CU-CP.

(9) Class	CP1. Gestionar nomenclador	CP2. Listar nomenclador	CP3. Buscar nomenclador	CP4. Importar nomenclador	CP5. Exportar nomenclador	CP6. Eliminar valor	CP7. Ordenar nomencladores por columna	CP8. Configurar cantidad de elementos a visualizar	CP9. Filtrar nomenclador por propiedades
Gestionar nomenclador	✓								
Listar nomenclador		✓							
Buscar nomenclador			✓						
Importar nomenclador				✓					
Exportar nomenclador					✓				
Eliminar valor						✓			
Ordenar nomencladores p...							✓		
Configurar cantidad de ele...								✓	
Filtrar nomenclador por pro...									✓

Anexo 5. Casos de prueba

Tabla 15: Caso de Prueba Gestionar nomenclador SC2 Editar nomenclador.

Escenario	Descripción	Nombre	Descripción	Valores	Respuesta del sistema	Flujo central
EC 2.1 Editar nomenclador	El escenario de prueba permite al usuario autenticado editar un nomenclador seleccionado	V	V	V	El sistema crea el nuevo elemento y muestra el mensaje: "Nomenclador creado satisfactoriamente."	1- Presionar el menú Administración, 2- Seleccionar Administración de nomencladores.3- Seleccionar la funcionalidad "Editar nomenclador".4- Edita los datos. 5- Presiona el botón "Aceptar"
		Carro	Moderno	Audi, Mercedes		
EC 2.2 Validar campos vacíos	El escenario de prueba permite al usuario validar los campos vacíos.	I	V	V	El sistema señala el campo obligatorio vacío y muestra el mensaje: "El valor no puede estar vacío."	
		\$#carro	Moderno	Audi, Mercedes		
		V	I	V		
EC 2.3 Campos incorrectos	El escenario de prueba permite al usuario autenticado validar los campos incorrectos.	Carro	Es un carro\$%#	Audi, Mercedes	El sistema muestra el mensaje: "El valor contiene caracteres no admitidos."	
		I	V	V		
		Carro21%\$	Moderno	Audi, Mercedes		
		V	I	V		
EC 2.4 Cancelar acción	El escenario de prueba permite al usuario	Carro	#\$%es un	Audi, Mercedes	El sistema cancela la acción.	1- Presionar el menú Administración, 2- Seleccionar Administración de
		NA	NA	NA		

Tabla 16: Caso de Prueba Gestionar nomenclador SC3 Ver detalles del nomenclador.

Escenario	Descripción	Respuesta	Flujo central
EC 3.1 Ver detalles del nomenclador	El escenario de prueba permite al usuario ver los detalles del nomenclador seleccionado	El sistema muestra los nomencladores	1- Presionar el menú Administración, 2- Seleccionar Administración de nomencladores.3- Seleccionar la funcionalidad "Ver detalles del nomenclador".4- Presiona el botón "Aceptar"

Anexo 6. Acta de evaluación interna de producto de software

Acta de Evaluación Interna de Productos Software

CIGED

Fecha de emisión del acta: 01/06/2018

Emitida a favor de: Módulo de Gestión de Nomencladores para el GDA XABAL eXcriba 3.1.

Datos del producto:

Artefacto	Versión	Estado final	Cantidad de Iteraciones	Tipos de pruebas realizadas
Aplicación	3.0	0	4	Funcionales

Anexo:

NC Pendientes / No Procede	Clasificación de la NC	Respuesta del Equipo de desarrollo
En CP Importar nomenclador el sistema permite la inserción de caracteres no válidos en la propiedades Nombre y Descripción.	Validación	Pendiente. Dado que el documento se debía mandar para el tribunal, se queda pendiente a solucionar antes del día de la defensa de tesis.

Participantes


 Lizandra Campesino Rodriguez

Asesora de Calidad


 Elva del Carmen Pineda

Jefe de proyecto


 Alejandro Pérez Pinedas



Fig 16: Acta de evaluación interna de producto de software.

Anexo 7. Acta de aceptación del módulo



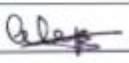
Acta de aceptación

ACTA DE ACEPTACIÓN

En cumplimiento del **Convenio de colaboración con el centro CIGED** y en función de la ejecución del proyecto de tesis: **Módulo de Gestión de nomencladores para el GDA XABAL eXcriba 3.1**, se hace entrega del producto que se relaciona a continuación:

- Módulo de Gestión de nomencladores para el GDA XABAL eXcriba 3.1.

La Parte Cliente, luego de haber revisado el producto de trabajo determina su aceptación.

Entrega Tesista	Recibe Centro CIGED
Nombre y apellidos: Alejandro Pirez	Nombre y apellidos: Solangel Rodríguez
Cargo: Desarrollador 	Cargo: Jefe de Dpto. Aplicaciones 

Fecha: 01/09/2018



Fig 17: Acta de aceptación del módulo.