

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**



**CENTRO DE INFORMATIZACIÓN DE LA GESTIÓN DOCUMENTAL**

**Personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS**

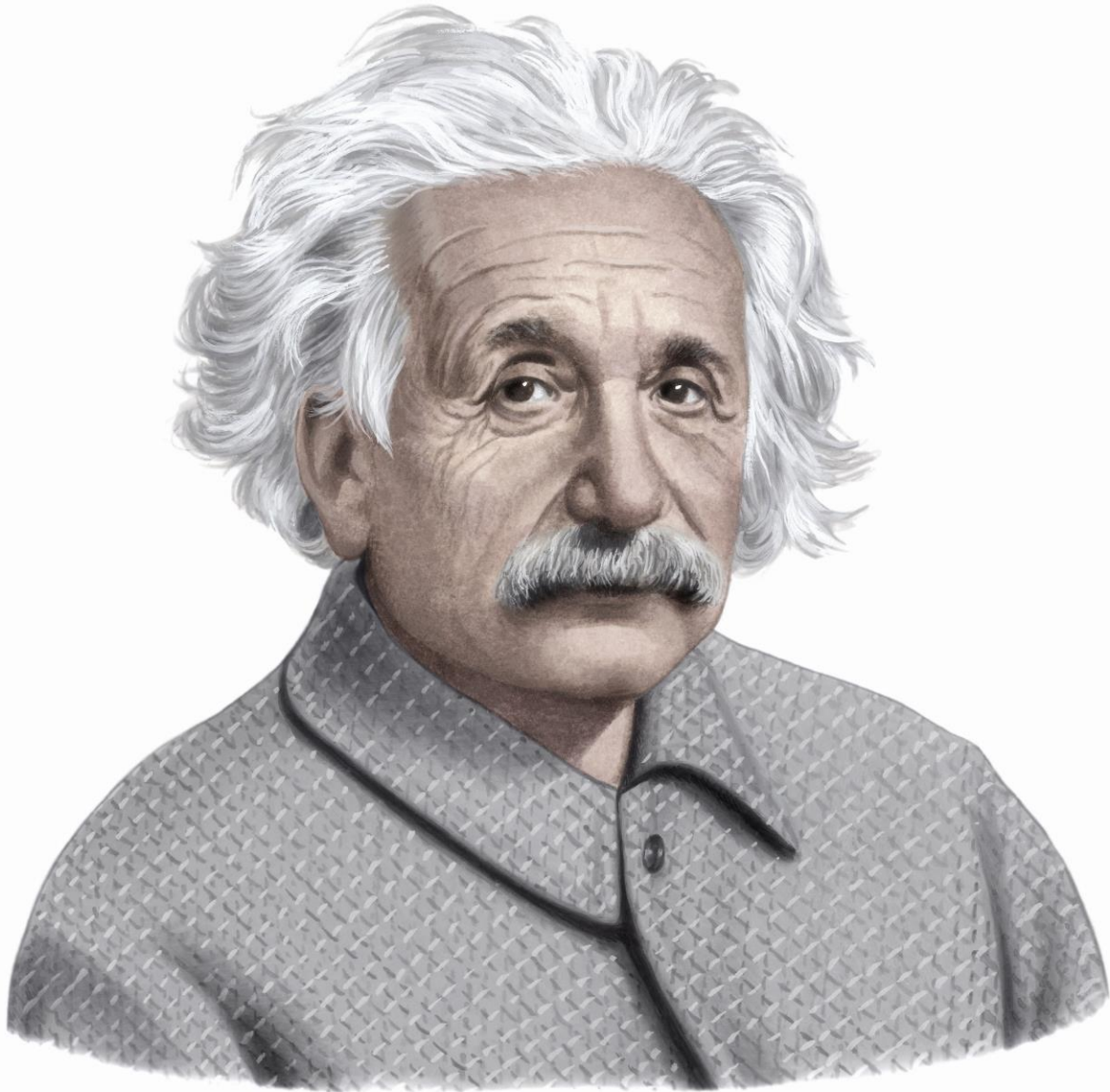
**Autor: Anamelys Lugo Zulueta.**

**Tutor: Ing Geidis Arévalo Quintana.**

**La Habana  
2018**

## *Fraser*

*“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”*



*Albert Einstein*

## DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que tiene por título: Personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Firma del autor

Anamelys Lugo Zulueta

\_\_\_\_\_

Firma del tutor

Geidis Arévalo Quintana

## **Dedicatoria**

A mis padres por guiarme por el buen camino y por estar siempre pendiente de mí. A mi novio por todos estos años que hemos pasado juntos. A ustedes les dedico esta tesis y todo mi esfuerzo. Los Amo.

## Resumen

En la Universidad de las Ciencias Informáticas se realizan revisiones técnicas formales a los proyectos. Estas revisiones son actividades de calidad para prevenir, controlar y mejorar la calidad de los productos. Para la realización de estas revisiones se necesita la información que contiene el expediente del proyecto a revisar, la cual es almacenada en memorias usb o es enviada por correo electrónico provocando que la información no llegue a su destino ya sea por la pérdida de la memoria o por problemas de conexión con el servidor de correo electrónico. Como solución a esta problemática la presente investigación propone la personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 con el fin de gestionar y controlar el proceso Revisión Técnica Formal. Para la personalización del sistema se realizó un estudio de algunas soluciones desarrolladas con fines similares. Se determinó la metodología a utilizar para guiar el desarrollo de la personalización. Se seleccionaron las herramientas y tecnologías a utilizar en el desarrollo de la propuesta de solución. Además se hizo un análisis del proceso para así lograr una mayor comprensión del mismo. Luego se diseñó, implementó y por último se le realizaron pruebas a la personalización para verificar su correcto funcionamiento.

**Palabras claves:** Calidad, personalización, proceso, Revisiones Técnicas Formales.

## Summary

At the University of Informatics Sciences, formal technical reviews are carried out on the projects. These reviews are quality activities to prevent, control and improve the quality of products. In order to carry out these reviews, the information contained in the case file of the project to be reviewed is needed, which is stored in usb memories or sent by e-mail, causing the information not to reach its destination, either due to memory loss or due to connection problems with the email server. As a solution to this problem, this research proposes the personalization of the Xabal eXcriba 3.1 Administrative Documents Manager in order to manage and control the Formal Technical Review process. For the personalization of the system, a study was made of some solutions developed with similar purposes. The methodology to be used to guide the development of personalization was determined. The tools and technologies to be used in the development of the solution proposal were selected. In addition, an analysis of the process was made in order to achieve a better understanding of it. Then it was designed, implemented and finally tests were carried out on the personalization to verify its correct functioning.

**Keywords:** Quality, personalization, process, Formal Technical Reviews.

**Índice**

Introducción .....	1
Capítulo 1: Fundamentación teórica.....	5
1.1 Principales conceptos asociados al dominio del problema.....	5
1.2 Estudio de las tendencias actuales del proceso RTF.....	8
1.3 Metodología de desarrollo de software.....	11
1.4 Gestor documental a utilizar para la personalización.....	11
1.4.1 GDA eXcriba o Gestor de Documentos Administrativos Xabal eXcriba 3.1.....	11
1.4.2 ECM Alfresco Community 4.2.f.....	12
1.5 Lenguaje de modelado de procesos BPMN v 2.0.....	12
1.6 Lenguaje de programación.....	13
1.7 Lenguaje de marcado.....	13
1.8 Herramientas y tecnologías.....	14
1.8.1 Eclipse Kepler 2.0.2.....	14
1.8.2 Activiti Designer 5.15.....	14
1.8.3 Visual Paradigm v 8.0.....	14
1.8.4 Axure RP v 7.0.....	15
1.9 Conclusiones parciales.....	15
Capítulo 2: Propuesta de solución.....	17
2.1 Descripción de la propuesta de solución.....	17
2.2 Mapa de proceso.....	18
2.3 Descripción y modelado del proceso RTF de la DCSW.....	18
2.4 Patrones de modelado de procesos.....	20
2.5 Requisitos del sistema.....	21
2.5.1 Técnicas para la captura de requisitos.....	21
2.5.2 Requisitos funcionales: .....	21
2.5.3 Requisitos no funcionales: .....	23

2.6 Descripción de requisitos por procesos. ....	24
2.7 Validación de los requisitos. ....	25
2.8 Matriz de trazabilidad. ....	26
2.9 Modelo conceptual. ....	27
2.10 Arquitectura del software. ....	29
2.11 Patrones de diseño. ....	30
2.12 Conclusiones parciales. ....	31
Capítulo 3: Implementación y pruebas. ....	33
3.1 Tipología de los documentos. ....	33
3.1.1 Tipificación de los documentos. ....	33
3.2 Estructura organizativa de los documentos. ....	34
3.3 Tabla de acceso y seguridad. ....	35
3.4 Configuración de los grupos y permisos. ....	36
3.5 Configuración del flujo de trabajo y modelado del proceso en Activiti. ....	37
3.5.1 Configuración del flujo de trabajo. ....	37
3.5.2 Despliegue del flujo de trabajo en Alfresco. ....	43
3.6 Tarea programada (Scheduled jobs). ....	48
3.7 Reglas de contenido. ....	48
3.8 Estándares de codificación. ....	49
3.9 Diagrama de despliegue. ....	51
3.10 Pruebas de software. ....	52
3.11 Conclusiones parciales. ....	57
Conclusiones generales. ....	58
Referencias bibliográficas. ....	59
Bibliografía. ....	62

Anexos.....65



## Índice de figuras

Fig. 1: Mapa de procesos de la DCSW. ....	18
Fig. 2: Patrón de modelado secuencia. ....	20
Fig. 3: Patrón de modelado decisión exclusiva. ....	20
Fig. 4: Prototipo del requisito funcional Iniciar flujo de trabajo RTF. ....	26
Fig. 5: Matriz de trazabilidad requisito – requisito. ....	27
Fig. 6: Modelo conceptual. ....	28
Fig. 7: Arquitectura de la propuesta de solución. ....	30
Fig. 8: Estructura organizativa de los documentos. ....	34
Fig. 9: Grupos definidos. ....	36
Fig. 10: Modelado del flujo de trabajo RTF. ....	38
Fig. 11: Configuración de la tarea revisar expediente de proyecto. ....	45
Fig. 12: Configuración del proceso RTF. ....	46
Fig. 13: Configuración del fichero RevisionesTécnicasFormales-generic-context.xml. ....	46
Fig. 14: Configuración del fichero RevisionesTécnicasFormales-message-workflow.properties. ....	47
Fig. 15: Configuración del fichero RevisionesTécnicasFormales-form-config.xml. ....	47
Fig. 16: Configuración del fichero RevisionesTécnicasFormales-context.xml. ....	48
Fig. 17: Tarea programada. ....	48
Fig. 18: Regla de contenido CSV. ....	49
Fig. 19: Estándar de codificación declaración de funciones. ....	49

Fig. 20: Estándar de codificación comentarios. ....	50
Fig. 21: Estándar de codificación declaración de variables. ....	50
Fig. 22: Estándar de codificación instrucción for. ....	51
Fig. 23: Diagrama de despliegue. ....	52
Fig. 24: Método para la prueba de caja blanca. ....	56
Fig. 25: Grafo de flujo del método. ....	56

## **Índice de tablas**

Tabla. 1: Resumen de las herramientas similares.....	10
Tabla. 2: Descripción del proceso RTF. ....	19
Tabla.3: Especificación de requisito Iniciar flujo de trabajo RTF.....	24
Tabla. 4: Diccionario de datos de RTF .....	28
Tabla. 5: Tabla de acceso y seguridad.....	35
Tabla.6: Permisos. ....	37
Tabla. 7: Descripción del flujo de trabajo. ....	38
Tabla. 8: Caso de prueba # 1.....	54
Tabla. 9: Resultados de las pruebas. ....	54
Tabla. 10: Caso de prueba camino básico 1. ....	57
Tabla. 11: Caso de prueba camino básico 2. ....	57

## **Introducción**

En la actualidad la industria del software ha logrado alcanzar una posición relevante dentro de las Tecnologías de la Información y las Comunicaciones (TIC). Sus aplicaciones cada vez son más utilizadas en cualquier rama de la ciencia o de la economía. Las empresas de desarrollo de software buscan alternativas para mejorar la producción, satisfacer las necesidades del cliente y garantizar la calidad del producto.

Hoy en día las empresas productoras de software reconocen que la calidad del producto se traduce en ahorro de costos, de tiempo y de recursos (Figueroa, 2000). La calidad del software se consigue por medios de la aplicación de métodos de la ingeniería de software, prácticas adecuadas de administración y un control de la calidad exhaustivo, todo lo cual es apoyado por la infraestructura de aseguramiento de la calidad (Pressman, 2010).

El aseguramiento de la calidad del software envuelve al proceso de desarrollo de software completo: monitoreando y mejorando el proceso; asegurándose que cualquier estándar y procedimientos adoptados sean seguidos; y que los problemas sean encontrados y tratados. Implica revisar y auditar los productos y proveer a las gerencias (incluyendo a la de proyectos) con los resultados de estas revisiones (Pressman, 2010).

Cuba no está ajena a este proceso, puesto que las organizaciones productoras de software se han planteado como meta obtener un producto que cumpla con las normas y estándares internacionales de calidad y para ello forma profesionales calificados en esta rama (Alvarado, 2016). La Universidad de las Ciencias Informáticas (UCI) es un ejemplo vigente de desarrollo de software de alta calidad.

Esta universidad fue creada el 23 de septiembre del 2002 tiene como misión formar profesionales comprometidos con su patria y altamente calificados en la rama de la Informática, así como producir aplicaciones y servicios informáticos, a partir de la vinculación estudio-trabajo como modelo de formación. En el año 2015 la institución recibió el certificado que avala la actividad de producción y desarrollo de software con el Modelo de Madurez de Capacidad Integrada (CMMI, del inglés Capability Maturity Model Integration) nivel 2. Este certificado evidencia las buenas prácticas en la gestión de software (Correa, 2017). Dentro de la estructura productiva de la universidad destacan la existencia de varios centros que desarrollan actividades de investigación, desarrollo e innovación (I+D+i). Entre ellos se encuentra el Centro de Informatización de 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, en el cual se desarrolla el Gestor de Documentos Administrativos (GDA) Xabal eXcriba 3.1. Este producto está destinado a la gestión del expediente de proyecto en la UCI y tiene como objetivo principal facilitar el control de los procesos documentales que se ejecutan dentro de cualquier entidad, desde la elaboración de un

documento en su fase de inicio hasta su conservación o expurgo en el Archivo de gestión (Informáticas, 2017). Incluye todas las acciones sobre los documentos tales como: crear, clasificar, control de versiones, definir Tipologías Documentales, gestionar flujos documentales, almacenar documentos en diferentes formatos electrónicos, gestionar los trámites de los documentos que se generan o reciben, salvaguardar el patrimonio documental y automatizar el Cuadro de clasificación entre otras (Informáticas., 2017).

Para el aseguramiento de la calidad del software, la UCI cuenta con una Dirección de Calidad de Software (DCSW) que tiene como misión principal garantizar la calidad técnica de los productos de trabajo de planificación, requisitos, arquitectura y base de datos a través de Revisiones Técnicas Formales (RTF) y que el software desarrollado cumpla con los requisitos del cliente mediante la ejecución de pruebas de liberación (Marín, 2015). La dirección está compuesta por un grupo de revisores, un departamento de prueba y un planificador, el cual es el encargado de llevar el control de todas las tareas y actividades del área de calidad. A su vez la dirección se subordina a la Vicerrectoría de Producción, la cual implementa estrategias y políticas que garanticen el proceso de desarrollo de soluciones informáticas. Toma como base las proyecciones del país en el fomento de la industria cubana de software y la exportación de productos y servicios informáticos (Informáticas., 2017). La DCSW realiza varias actividades para prevenir, controlar y mejorar la calidad de los productos, entre ellas se encuentra las RTF. Estas son un método probado para eliminar defectos de manera temprana, proporcionar una visión de valor sobre los productos de trabajo (Informáticas, 2017).

Actualmente el proceso de revisiones técnicas formales presenta limitantes en su realización, debido a que los revisores de la DCSW deben solicitar la información requerida mediante el correo electrónico o de lo contrario deben solicitar de manera presencial la información. Los resultados de la revisión deben ser enviados por correo electrónico, esto viene dado porque existe un GDA Xabal eXcriba 3.1 para calidad y otro para los centros de producción. Además, no todos los centros tienen un área con su estructura de carpeta creada en el GDA Xabal eXcriba de los centros de producción.

Estos factores afectan y retrasan la realización de dicha actividad, debido a que pueden ocurrir problemas de conexión con el servidor de correo electrónico provocando que la información requerida no llegue a su destino en el tiempo estimado. Cuando los revisores solicitan la información de manera presencial almacenan la información en memorias usb, las cuales son un dispositivo físico susceptible a daños. La pérdida de la memoria usb trae consigo que personas no autorizadas tengan acceso a la información. Las memorias usb pueden presentar problemas de conectividad debido al deterioro de los conectores quedando totalmente inutilizables y provocando la pérdida parcial o total de la información. El proceso tiene una deficiente centralidad y seguridad de la información, lo que dificulta la integridad de la información que se genera en dicho proceso.

Dada la situación problemática expuesta anteriormente se identifica como **problema de investigación**: ¿Cómo contribuir a la gestión y control del proceso documental revisión técnica formal de la Dirección de Calidad de Software?

Se define como **objeto de estudio**: el proceso revisión técnica formal. Para dar respuesta al problema antes mencionado se traza como **objetivo general**: Personalizar el Gestor de Documentos Administrativos Xabal eXcriba 3.1 para la gestión y control de las RTF de la DCSW.

Se define como **campo de acción**: Herramientas para la gestión y control de las RTF.

En correspondencia con el problema de investigación, el objeto de estudio, el objetivo general y el campo de acción se determinaron las siguientes **preguntas científicas**:

1. ¿Qué fundamentos teóricos y metodológicos sustentan las tendencias actuales del estudio del desarrollo de las Revisiones Técnicas Formales?
2. ¿Cuál es el estado actual del proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software?
3. ¿Qué estructura tendrá la personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso de RTF de la DCSW en la UCI?
4. ¿Qué resultados se alcanzarán con la factibilidad práctica de la personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso RTF de la DCSW en la UCI?

Para darle solución a las preguntas científicas se proponen las siguientes **tareas de investigación**:

1. Análisis de los principales fundamentos teóricos y metodológicos que sustentan las tendencias actuales acerca del proceso RTF de la DCSW.
2. Diagnóstico del estado actual del proceso RTF de la DCSW.
3. Diseño de la propuesta de solución utilizando la metodología de desarrollo de software AUP- UCI.
4. Constatación de los resultados alcanzados con la personalización del Gestor de Documentos Administrativo Xabal eXcriba 3.1 para el proceso RTF de la DCSW en la UCI.

Durante el desarrollo de esta investigación se hizo necesario profundizar en el estudio de los temas abordados, por lo que se utilizaron varios **métodos científicos de investigación**:

### **Métodos teóricos:**

- **Analítico-Sintético**: utilizado para realizar un análisis de los conceptos fundamentales relacionados con la calidad y de esta forma realizar una profundización de los mismos y arribar a conclusiones, lo cual contribuye a una mejor elaboración de la propuesta de solución.

- **Histórico-lógico:** utilizado para realizar un estudio sobre el desarrollo histórico de los antecedentes de la actividad de calidad Revisiones Técnicas Formales (RTF).
- **Modelación:** utilizado para realizar el modelado del proceso documental identificado en el área, posibilitando un mejor entendimiento del negocio. También es utilizado para realizar las matrices de trazabilidad.

### **Métodos empíricos:**

- **Entrevista:** Se desarrolla una entrevista al personal del área en cuestión, con el objetivo de obtener toda la información necesaria, así como sus necesidades y posibles resultados esperados (Ver anexo 1).

Para un mejor entendimiento tanto del problema como de la solución que se propone, el trabajo de diploma se ha estructurado de la siguiente forma: Resumen, Introducción, 3 Capítulos, Conclusiones, Recomendaciones, Glosario de términos, Referencias bibliográficas Bibliografía y Anexos. A continuación se desglosa brevemente el contenido a tratar en cada capítulo.

**Capítulo 1: Fundamentación teórica.** En este capítulo se abordarán los conceptos fundamentales relacionados con el problema a resolver. Se realiza un estudio de la metodología de desarrollo de software y de las herramientas que realizan RTF. Se exponen los lenguajes de programación, el lenguaje de modelado, las herramientas y tecnologías fundamentales a utilizar en la personalización.

**Capítulo 2: Propuesta de solución.** En este capítulo se elabora una presentación de la propuesta de solución a la problemática. Se determinan las funcionalidades y características del sistema. Se realiza la descripción y modelado del proceso de negocio. Se determinan los patrones de modelado a utilizar y se elabora la descripción de los requisitos por procesos y el modelo conceptual.

**Capítulo 3: Implementación y pruebas.** En este capítulo se implementa la propuesta de solución, se define la estructura organizativa de los documentos. Se elabora la tabla de acceso y seguridad del sistema, la cual define los permisos de los usuarios sobre los documentos en dependencia de los roles de cada uno de los archivos. Se muestra como queda configurado el flujo de trabajo y el modelado del proceso en la herramienta Activiti.

## **Capítulo 1: Fundamentación teórica.**

En este capítulo se expone un estudio de los conceptos fundamentales que se deben tener en cuenta para la comprensión del problema a resolver. Se determina la metodología de desarrollo de software, los lenguajes de programación y modelado, las herramientas y tecnologías a utilizar.

### **1.1 Principales conceptos asociados al dominio del problema.**

En el presente epígrafe se evidencian algunos conceptos fundamentales relacionados con la investigación. Se considera necesario el conocimiento de los mismos para lograr una mayor comprensión de cada uno de ellos, dado que es necesario su empleo en el desarrollo del trabajo.

#### **Personalización**

Según el Diccionario de la Lengua Española, personalización es dar carácter personal a algo. Es adaptar algo al gusto o necesidades del usuario (RAE, 2014).

#### **Calidad**

Según E.W. Deming, (1989). Calidad es el grado predecible de uniformidad y fiabilidad a un bajo costo y que se ajuste a las necesidades del mercado. La calidad no es otra cosa más que "una serie de cuestionamientos hacia una mejora continua" (Deming, 1989).

Según el consultor de gestión del siglo XX Joseph Moses Juran (1993): calidad es el conjunto de características que satisfacen las necesidades de los clientes, además calidad consiste en no tener deficiencias. La calidad es "la adecuación para el uso, satisfaciendo las necesidades del cliente" (Moses, 1993).

Según el empresario norteamericano Phil Crosby (1996) la calidad es el cumplimiento de normas y requerimientos precisos, es ajustarse a las especificaciones. Su lema es "hacerlo bien, a la primera vez y conseguir cero defectos", confirmando que la calidad está basada en cuatro principios absolutos: cumplimiento de requisitos, sistema de prevención, su estándar de realización es cero defectos y su medida es el precio del incumplimiento (Crosby, 1996).

Según la NC/ISO9000:2005: Es el grado en el que un conjunto de características inherentes cumple con los requisitos.

Luego del análisis de los conceptos expuestos anteriormente se toma como referencia el concepto dado por el autor Phil Crosby, debido a que es la definición más completa cuando se refiere al cumplimiento de los



requerimientos.

### **Calidad de software:**

La definición de la calidad del software según la IEEE, Std. 610-1990<sup>1</sup>, es “el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

La norma ISO 8402:1994<sup>2</sup> plantea que es “el conjunto de características de una entidad que le confiere su aptitud para satisfacer las necesidades expresadas y las implícitas.

Según Roger Pressman<sup>3</sup> la calidad del software es la: “Concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”.

Luego del análisis realizado de las definiciones expuestas anteriormente se toma como referencia la definición dada por la IEEE, debido a que es la que más se relaciona con el proceso de calidad RTF. Además es concisa cuando se refiere al cumplimiento de los requerimientos especificados y las necesidades del cliente.

### **Aseguramiento de la calidad**

Según la norma ISO 9000:2005 el aseguramiento de la calidad es la parte de la gestión de la calidad orientada a proporcionar confianza en que se cumplirán los requisitos de la calidad.

El control y aseguramiento de la calidad son actividades esenciales para cualquier negocio que genere productos que generen otras personas. Actualmente, toda compañía tiene mecanismos para asegurar la calidad en sus productos (Pressman, 2010).

El aseguramiento de la calidad establece la infraestructura de apoyo a los métodos sólidos de la ingeniería de software, la administración racional de proyectos y las acciones de control de calidad, todo de importancia crucial si se trata de elaborar software de alta calidad. Además el aseguramiento de la calidad consiste en un conjunto de funciones de auditoría y reportes para evaluar la eficacia y completitud de las acciones de

---

<sup>1</sup> *Standard Glossary of Software Engineering Terminology, IEEE STD 610-1990, IEEE 1990*

<sup>2</sup> *8402-1994, ISO. Calidad de software. 1994.8402.*

<sup>3</sup> *Ingeniero de Software estadounidense graduado de la Universidad de Connecticut, autor del libro Ingeniería de Software, un enfoque práctico.*

control de calidad (Pressman, 2010).

El aseguramiento de la calidad tiene como meta proveer al equipo administrativo y técnico los datos necesarios para mantenerlo informado sobre la calidad del producto, con lo que obtiene perspectiva y confianza en que las acciones necesarias para lograr la calidad del producto funcionan (Pressman, 2010).

### **Revisiones del software**

Las revisiones del software son un filtro para el proceso del software. Se aplican en varios puntos durante la ingeniería del software y sirven para descubrir errores y defectos a fin de poder eliminarlos. Las revisiones del software purifican los productos de la ingeniería del software, incluso los modelos de requerimientos y diseño, código y datos de prueba (Pressman, 2010).

El objetivo principal de las revisiones del software es encontrar errores durante el proceso a fin de que no se conviertan en defectos después de liberar el software. Un beneficio de las revisiones técnicas es el descubrimiento temprano de los errores, de modo que no se propaguen a la siguiente etapa del proceso del software.

### **Revisiones Técnicas Formales**

Una revisión técnica formal (RTF) es una actividad del control de calidad de software realizada por ingenieros de software (Pressman, 2010). Es un método probado para eliminar defectos de manera temprana y proporcionar una visión de valor sobre los productos de trabajo y los componentes de producto que están siendo desarrollados y mantenidos (Informáticas, 2017).

Los objetivos de una RTF son:

1. Descubrir los errores en funcionamiento, lógica o implementación de cualquier representación del software.
2. Verificar que el software que se revisa cumple sus requerimientos.
3. Garantizar que el software está representado de acuerdo con estándares predefinidos.
4. Obtener software desarrollado de manera uniforme.
5. Realizar proyectos manejables.

### **Lista de chequeo**

Es un formato creado para realizar actividades repetitivas, controlar el cumplimiento de una lista de requisitos o recolectar datos ordenadamente y de forma sistemática. Se usa para hacer comprobaciones sistemáticas de actividades o productos asegurándose de que el trabajador o inspector no olvide nada importante (Ledesma , 2013).

Los principales usos de las listas de chequeo:

- ❖ Realizar actividades en las que es importante que no se olvide ningún paso y/o deben hacerse las tareas con un orden establecido.
- ❖ Realiza inspecciones donde se debe dejar constancia de cuáles han sido los puntos inspeccionados.
- ❖ Verificar o examinar artículos.
- ❖ Examinar o analizar la localización de defectos.
- ❖ Verificar las causas de los defectos.
- ❖ Recopilar datos para su futuro análisis.

## Proceso

Según la ISO 9001: 2015 un proceso es un conjunto de actividades que están interrelacionadas y que pueden interactuar entre sí. Estas actividades transforman los elementos de entrada en resultados, para ello es esencial la asignación de recursos ((ISO), 2015).

## Flujo de trabajo

*Alfresco Enterprise Content Management Implementation* define que:

“...un flujo de trabajo es una automatización de un proceso de negocio, durante el cual los documentos pasan de una persona a otra para una acción, de acuerdo con un conjunto de procedimientos y reglas...” (Shariff, 2013).

### 1.2 Estudio de las tendencias actuales del proceso RTF.

Para la informatización del proceso es necesario estudiar herramientas que realicen revisiones de software o revisiones técnicas formales, con el objetivo de analizar cómo aseguran la calidad de sus productos, para así tomar las características que se puedan adaptar a la solución que se desea implementar. Para ello se tuvieron en cuenta las siguientes características:

1. Lenguaje de programación con que fue desarrollado.
2. Actividad de calidad que realiza.
3. Es aplicación de escritorio (Sí/No).
4. Integración con Alfresco (Sí/No).

## Propuesta de la Facultad de Estadística e Informática de la Universidad Veracruzana, México.

La Facultad de Estadística e Informática de la Universidad Veracruzana en México desarrolló una aplicación de escritorio para la informatización de los procesos de RTF que se desarrollan en dicho centro. La

herramienta que se presenta aligera la carga de los revisores al realizar una evaluación preliminar de aspectos automatizables de los diferentes artefactos, de acuerdo con reglas predefinidas. Se espera favorecer la realización de revisiones, con lo cual mejorará la calidad del software (Hernández, 2012).

El objetivo principal de esta herramienta es mejorar la calidad del software, introduciendo a las RTF como parte del ciclo de vida de un proyecto, mediante la reducción significativa del tiempo necesario para realizarlas. Además de la disminución del número de errores que surgen del manejo de tal información, de manera que influyan positivamente en el desarrollo de software.

La herramienta cuenta con un prototipo que consta de dos aplicaciones:

- ❖ RETO.exe que básicamente permite la administración de los proyectos y la importación o asociación de modelos a cada proyecto para la realización de la RTF como tal.
- ❖ AdministraRETO.exe que proporciona flexibilidad a la herramienta para agregar o actualizar las guías de RTF y sus herramientas evaluadoras asociadas.

Esta herramienta se desarrolló con XMI versión 2.1 (XML Metadata Interchange), un nuevo estándar del Object Management Group basado en XML, pero con estructura propia para representar UML (Unified Modeling Language). Sólo se llegó a desarrollar la primera etapa de la herramienta.

### **Rational RequisitePro**

Herramienta centrada en documentos, que almacena los requisitos asociándolos a documentos directamente en la base de datos. Se auxilia especialmente en el control de cambio de requisitos con trazabilidad para especificaciones de software y pruebas. También, ayuda a los equipos de proyectos para gestionar sus necesidades, escribir casos de uso, mejorar la trazabilidad, fortalecer la colaboración, reducir la reanudación del proyecto, y aumentar la calidad (IBM Rational RequisitePro, 2010). Las principales funcionalidades de la herramienta Rational RequisitePro están guiadas a la gestión de requerimientos. Esta herramienta es una aplicación de escritorio desarrollada con el lenguaje de programación java centrada solamente en el análisis de los requisitos.

Esta herramienta trabaja con los sistemas operativos Microsoft Windows ® 2000 Professional, Microsoft Windows XP Professional. También trabaja con Word processor: Microsoft Word 2000,2002 or 2003. Tiene como principales características:

- ❖ Evitar la duplicación de trabajo utilizando integración avanzada, en tiempo real con Microsoft Word.
- ❖ Gestionar la complejidad con vistas detalladas de trazabilidad que muestran relaciones padres / hijos.
- ❖ Mitigar el riesgo del proyecto con indicación de los requisitos que pueden ser afectados por los cambios ascendentes o descendientes de requisitos.

**PMD**

Es una herramienta de calidad de código encargada de validar los estándares de construcción de un desarrollo. Es decir, chequea la sintaxis del código fuente que ha sido desarrollado, encontrando las ocurrencias de un determinado problema que haya sido previamente configurado para ser detectado (ANDALUCÍA, 2013).

Esta herramienta es una aplicación de escritorio que trata de definir entornos normalizados que ayuden a mejorar la integración y herencia de código ajeno, así como a facilitar el establecimiento de un entorno común de desarrollo, favoreciendo la comprensión del código y, sobre todo, la minimización del impacto de los errores de integración. Trabaja principalmente con lenguaje Java, aunque, con menos soporte, también posee conjuntos de reglas para JavaScript, xsl y ecmascript (Escalona, 2008). Es un software libre que se integra fácilmente en entornos como eclipse o netbeans.

Luego del estudio de los sistemas similares se concluye lo siguiente:

Tabla. 1: Resumen de las herramientas similares.

Características	Herramienta de la Universidad Veracruzana	Rational RequisitePro	PMD	GDA Xabal eXcriba 3.1
Lenguaje de programación	XMI	Java	Java	Java
Actividad de calidad	RTF	Revisión de requisitos	Revisión de código	-
Aplicación de escritorio	Sí	Sí	Sí	Si
Integración con Alfresco	No	No	No	Si

- ❖ Las herramientas que se desarrollaron con el lenguaje de programación java se centran en el análisis de los requisitos y en la revisión de código, por lo tanto no pueden utilizarse en el desarrollo de la propuesta de solución debido a que las RTF se le realizan a la arquitectura, base de datos, a la planificación y a los requisitos.
- ❖ Solamente la herramienta de la universidad Veracruzana de México realiza la actividad de calidad RTF. Debido a que no se tiene acceso al código fuente no se puede obtener ningún método que pueda reutilizarse en la propuesta de solución.
- ❖ Ninguna de las herramientas se integra con Alfresco debido a que no existe un servicio mediante el cual se pueda consumir la información relacionada con las RTF.

## **1.3 Metodología de desarrollo de software.**

Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas. La metodología de desarrollo de software a utilizar es la AUP-UCI, debido a que es la definida por la universidad para emplearse en los proyectos productivos. Dicha metodología es una variación de la metodología “Proceso Unificado Ágil” (AUP por sus siglas en inglés) en unión con el modelo CMMI-DEV v 1.3 (Capability Maturity Model Integration) (Sánchez, 2014).

La metodología AUP-UCI tiene 3 fases las cuales son inicio, ejecución y cierre. En la fase de ejecución trabaja con 7 disciplinas para el ciclo de vida de los proyectos las cuales son: modelo de negocio, requisitos, análisis y diseño, implementación, pruebas internas, pruebas de liberación, pruebas de aceptación. Además utiliza 4 escenarios para modelar el sistema en los proyectos.

Escenario no.1: Proyectos que modelen el negocio con Caso de Uso del Negocio (CUN) solo pueden modelar el sistema con Caso de Uso del Sistema (CUS).

Escenario no.2: Proyectos que modelen el negocio con Modelo Conceptual solo pueden modelar el sistema con CUS.

Escenario no.3: Proyectos que modelen el negocio con Descripción de Proceso de Negocio (DPN) solo pueden modelar el sistema con Descripción de Requisitos por Proceso (DRP).

Escenario no.4: Proyectos que no modelen negocio solo pueden modelar el sistema con Historias de Usuario (HU).

Debido a que la investigación se va a enfocar en procesos, específicamente el proceso de RTF se utilizará el escenario 3.

## **1.4 Gestor documental a utilizar para la personalización.**

Se utilizará como gestor documental para la realización de la personalización el Gestor de Documentos Administrativos Xabal eXcriba 3.1. Este gestor documental es la solución creada en el centro CIGED de la UCI para la gestión de documentos y la automatización de los flujos documentales de una organización. Además, utiliza como núcleo de su sistema informático el Gestor de Contenido Empresarial (ECM) Alfresco Community en su versión 4.2.f.

### **1.4.1 GDA eXcriba o Gestor de Documentos Administrativos Xabal eXcriba 3.1.**

El GDA Xabal eXcriba 3.1 es un software para la gestión documental, diseñado para tramitar los documentos administrativos que se generan o reciben dentro de las organizaciones a partir de sus funciones. 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,

evitando la pérdida de información y tributando como herramienta de apoyo para la toma de decisiones. Permite trabajar en un entorno de colaboración entre los usuarios de la organización, compartiendo sus 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: captura, creación, clasificación, descripción, protección, retención de archivo y expurgo (Informáticas., 2017).

### **1.4.2 ECM Alfresco Community 4.2.f.**

De acuerdo con la definición planteada por Jesús Salinas Revelles, Alfresco es un sistema de Gestión de Contenido Empresarial (ECM), normalmente utilizado para la gestión de documentos digitales en entornos exigentes (sometidos a importantes requisitos de escalabilidad, alta disponibilidad, capacidades de integración y auditabilidad.) (Revelles, 2014).

Alfresco gestiona todos los contenidos en las empresas: documentos, imágenes, fotos, páginas web, registros, documentos xml y cualquier otro fichero semiestructurado o no estructurado.

Los servicios que ofrece son unas de sus grandes ventajas. Permiten gestionar el contenido de los documentos y características, tales como, gestión de metadatos, control de versiones, gestión del ciclo de vida, flujo de trabajo, búsquedas, asociaciones a otros contenidos, etiquetados, comentarios, etc. Posee un único repositorio unificado para gestionar cualquier tipo de contenido: documentos, imágenes, videos y audio. Ofrece servicios tales como: gestión de documentos, gestión de contenido web, versionado a nivel de repositorio, superposición transparente, gestión de registros, flujos de trabajo basados en el motor Activiti 5.15, gestión de imágenes y soporte de varios idiomas.

### **1.5 Lenguaje de modelado de procesos BPMN v 2.0.**

Para el modelado de los procesos se escogió BPMN 2.0 (del inglés Business Process Modeling Notation), el cual es soportado por la plataforma Activiti Designer 5.15, plataforma escogida para el modelado del flujo de trabajo. Es una notación gráfica que describe la lógica de los pasos en un proceso de negocio. Esta notación ha sido diseñada especialmente para coordinar la secuencia de procesos y mensajes que fluyen entre participantes de actividades distintas.

Tiene como objetivo proporcionar una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. BPMN tiene la finalidad de servir como lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación (BizAgi, 2014).

La notación BPMN 2.0 se visualiza gráficamente en un diagrama que muestra el flujo de trabajo de un proceso de negocio. Esto facilita el diseño del diagrama por medio de objetos gráficos con apariencia y

nombres amigables. Los objetos en un proceso de negocio se dividen en cuatro categorías: objetos de flujo, objetos de conexión, carriles, y artefactos (Cruz, 2014).

### **1.6 Lenguaje de programación.**

Se utiliza el lenguaje de programación JavaScript, para implementar los flujos de trabajo sobre el GDA Xabal eXcriba 3.1.

#### **JavaScript 1.6**

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Es un lenguaje de programación interpretado, o sea, no requiere compilación (Alegsa, 2014). Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2015). Permite crear visualizaciones más atractivas y mayor interactividad y gestionar diferencias de implementación entre los distintos navegadores (Mestras, 2013).

Utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con JavaScript se pueden crear diferentes efectos e interactuar con los usuarios (Pérez, 2015). Es un lenguaje de programación completo, con una sintaxis, un conjunto de sentencias e instrucciones similares a las de otros lenguajes.

### **1.7 Lenguaje de marcado.**

Se hace uso del lenguaje XML (Extensible Markup Language) para realizar los ficheros de configuración, tales como: form-config.xml y model.xml.

#### **XML 1.0**

XML significa lenguaje de marcas generalizado (Extensible Markup Language). Es un lenguaje usado para estructurar información en un documento o en general en cualquier fichero que contenga texto, como por ejemplo ficheros de configuración de un programa o una tabla de datos. Ha ganado muchísima popularidad en los últimos años debido a ser un estándar abierto y libre, creado por el Consorcio World Wide Web, W3C, en colaboración con un panel que incluye representantes de las principales compañías productoras de software (Villate., 2001).

Es un lenguaje de meta-marcado, ya que se usa para describir metadatos a través de etiquetas de marcado. Provee un conjunto de reglas simples para diseñar formatos de texto que permiten estructurar los datos y un método uniforme para describir e intercambiar datos estructurados. Describe estructura y semántica, no formato. Este lenguaje no solo presenta el contenido de un texto, sino que lo dota de significado, además de que el contenido de un documento es separado de cualquier noción de presentación.



## **1.8 Herramientas y tecnologías.**

### **1.8.1 Eclipse Kepler 2.0.2.**

Eclipse es una plataforma de desarrollo de código abierto basada en Java. Por si misma, es simplemente un marco de trabajo y un conjunto de servicios para la construcción del entorno de desarrollo de los componentes de entrada. Afortunadamente, Eclipse tiene un conjunto de complementos, incluidas las Herramientas de Desarrollo de Java (JDT). Eclipse también incluye el Entorno de Desarrollo de Complementos (PDE), que es de interés principalmente para los desarrolladores que quieren extender Eclipse, dado que les permite construir herramientas que se integran sin dificultades con el entorno de Eclipse. Dado que todo en Eclipse es un complemento, todos los desarrolladores de herramientas tienen un campo de juego de nivel para ofrecer extensiones a Eclipse y para proporcionar un entorno de desarrollo integrado y unificado para los usuarios (CORPORATION, 2012.).

### **1.8.2 Activiti Designer 5.15.**

Activiti Designer es un plugin de Eclipse Kepler 2.0.2. De esta forma el desarrollador puede trabajar tanto en el proceso en XML como en el esquema gráfico. Es decir con el código en el mismo entorno de desarrollo y acceder fácilmente entre las diferentes piezas relacionadas con la lógica (Calleja, 2010).

Activiti es una plataforma ligera de flujos de trabajo y BPM dirigida a empresarios, desarrolladores y administradores de sistemas. Ofrece una herramienta de flujo de trabajo centralizada con la que gestionar las necesidades de contenidos de Alfresco Community. Con Activiti, los desarrolladores pueden especificar nuevas definiciones de flujos de trabajo en BPMN 2.0 y añadirlas a Alfresco para gestionar procesos basados en documentos (Calleja, 2010).

### **1.8.3 Visual Paradigm v 8.0.**

Visual Paradigm for UML es una herramienta de Ingeniería de Software Asistida por Computadora (CASE: Computer Aided Software Engineering), diseñada para la ayuda al desarrollo de software. Soporta estándares de la industria clave, tales como Lenguaje de Modelado Unificado (UML) (López, 2017).

Esta herramienta ofrece un diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad, disponibilidad de múltiples versiones y puede integrarse con los principales IDEs. Es extensible mediante desarrollo de nuevos plug-ins (López, 2017). Visual Paradigm para UML se ha actualizado rápidamente en sintonía con el nuevo desarrollo de técnicas de modelado UML 2.1 con el propósito de generar un entorno de modelados visuales en el que se reúnen hoy todas las necesidades tanto de software y tecnología, como de las necesidades de comunicación.

### **1.8.4 Axure RP v 7.0.**

Axure RP es una herramienta rápida de creación de diagramas, prototipos y especificaciones para sitios web (Scientec., 2016). Axure RP se distingue principalmente por su facilidad de uso y riqueza de su diseño, tendrá acceso a:

Especificaciones automáticas pre-configurables.

- ❖ Simulación de ideas de forma rápida.
- ❖ Comunicación de ideas más claras.
- ❖ Diseño eficiente.
- ❖ Ahorro de tiempo con la documentación.

Respecto a la interactividad con otras herramientas, Axure permite la exportación de sus propuestas de prototipado a los siguientes formatos: Html, Word, Cvs, algunos formatos de imagen (Bitmap, PNG, JPG o GIF) y RP (formato propio de Axure). En cambio, sólo permite la importación directa de ficheros en formato RP de Axure.

### **1.8.5 Visual Studio Code v 1.17.**

Visual Studio Code es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, MacOS y Linux. Tiene soporte integrado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity) (Microsoft, 2018). VS Code soporta múltiples cursores para ediciones simultáneas rápidas. Puede agregar cursores secundarios (diluidos) con Alt + Click. Cada cursor funciona de forma independiente en función del contexto en el que se encuentra. Una forma común de agregar más cursores es presionando Ctrl + Alt + Abajo o Ctrl + Alt + Arriba que insertan cursores debajo o arriba (Microsoft, 2018).

### **1.9 Conclusiones parciales.**

En este capítulo se arribaron a las siguientes conclusiones:

- ❖ El estudio realizado sobre los conceptos fundamentales facilitó sentar las bases teóricas de la investigación.
- ❖ El estudio del estado del arte se realizó con el objetivo de identificar posibles soluciones. Teniendo como resultado que en la bibliografía estudiada no se encontraron herramientas que solucionen parcial o totalmente el problema de investigación anteriormente expuesto.

## *Capítulo 1: Fundamentación teórica.*

- ❖ El uso de la metodología de desarrollo de software permitirá organizar y guiar por fases de desarrollo la solución propuesta.
- ❖ La selección de las herramientas y tecnologías permitió una mayor comprensión de las mismas para su correcto uso.

## **Capítulo 2: Propuesta de solución.**

Se describen, en este capítulo, la propuesta de solución, las funcionalidades y características del sistema. Se elabora la descripción y modelado del proceso de negocio. Se identificaron los patrones de modelado a utilizar. Se realizan el modelo conceptual, las matrices de trazabilidad, la arquitectura y los patrones de diseños utilizados.

### **2.1 Descripción de la propuesta de solución.**

Con el presente trabajo se pretende personalizar el proceso RTF en el GDA Xabal eXcriba 3.1 que contribuirá a evitar la pérdida de los documentos en formato electrónico. Además facilita el acceso a los documentos, garantiza la seguridad de los documentos en formato electrónico estableciendo permisos a usuarios y centraliza los documentos que se generan en las RTF. La propuesta de solución permitirá guardar todos los resultados encontrados en cada revisión.

El proceso de la revisión inicia una vez que el asesor de calidad del centro realiza la solicitud de prueba de RTF, la cual es planificada por la DCSW en la herramienta Gespro. Luego a través de un servicio, previamente creado por la Dirección de Producción de Software se consultan los datos necesarios de la revisión. La nueva revisión iniciará de forma automática cuando se verifique todos los días a las 6:00 am si existe una nueva planificación de revisión RTF, en caso de que haya una nueva planificación se les notifica a los expertos implicados en la revisión.

Una vez realizada la notificación se copia en el espacio de trabajo del usuario una copia del expediente del proyecto para que los documentos originales no sufran modificaciones. Primeramente el experto líder emite la minuta de reunión de inicio, documento que registra las principales cuestiones tratadas en la reunión. Al finalizar la realización de la minuta de reunión el grupo de expertos revisa el expediente de proyecto. Cada experto realiza un tipo de RTF, las cuales son: planificación, requisito, base de datos y arquitectura. En la revisión realizan lista de chequeo dependiendo del tipo de RTF que realizan.

Las listas de chequeo describen los indicadores necesarios para evaluar los documentos. Los indicadores se basan en preguntas, las cuales generan un método de evaluación, especificando la descripción de la puesta en práctica de cada criterio, según se encuentre en los documentos del expediente del proyecto. También se genera una minuta de reunión de cierre y un dictamen técnico, el cual es la constancia de que al proyecto le fueron realizadas las RTF.

Además en la personalización se administran los usuarios y se definen los permisos de cada uno sobre los documentos que van a revisar. El acceso a la personalización se realizará mediante la autenticación a través de usuario y contraseña. Se creará una carpeta con la fecha del día en que finalizó la revisión, en la cual serán guardados todos los documentos generados. Esta carpeta será guardada en una carpeta del

expediente de proyecto nombrada Evaluaciones. En caso de no poderse iniciar el proceso de forma automática se determinará un usuario para que inicie el proceso de forma manual para que la realización de la revisión no sea afectada. La personalización le permitirá a los usuarios crear documentos a partir de plantillas tales como: minuta de reunión de inicio y de cierre, el dictamen técnico y las listas de chequeo.

### 2.2 Mapa de proceso.

El mapa de procesos es una gráfica que permite visualizar fácilmente cuáles son y cómo se relacionan los procesos de una organización (Valencia). Tipos de procesos en cualquier organización tales como (Europea de excelencia, 2015):

**Procesos estratégicos:** Son procesos destinados a definir y controlar las metas de la organización, sus políticas y estrategias. Permiten llevar adelante la organización. Están en relación muy directa con la misión/visión de la organización. Involucran personal de primer nivel de la organización.

**Procesos claves:** Son procesos que permiten generar el producto/servicio que se entrega al cliente, por lo que inciden directamente en la satisfacción del cliente final. Generalmente atraviesan muchas funciones. Son procesos que valoran los clientes y los accionistas.

**Procesos de apoyo:** Dan apoyo o soporte a los procesos clave. Su valor es indirecto y generalmente sus clientes son internos.

La DCSW clasifica sus procesos internos en estratégicos, claves y de apoyo, quedando su mapa de procesos como se muestran en la siguiente figura:

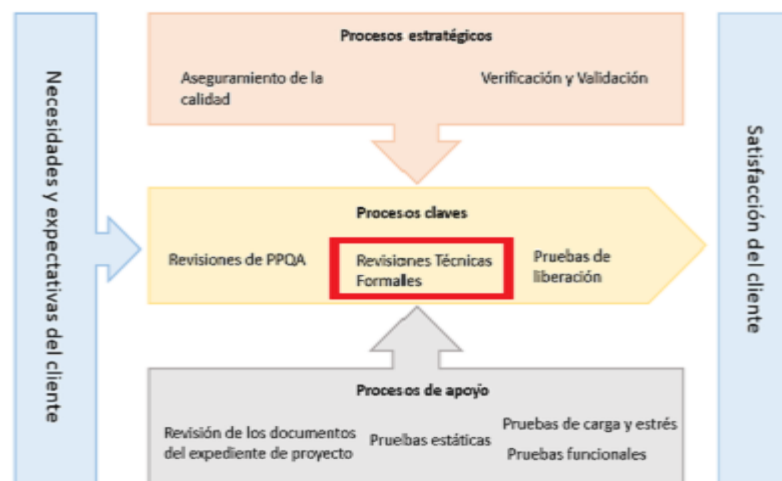


Fig. 1: Mapa de procesos de la DCSW.

### 2.3 Descripción del proceso RTF de la DCSW.

A continuación se muestra la descripción actual del proceso Revisión Técnica Formal, el cual es el definido por la universidad. Este proceso está publicado en el sitio mejoras.prod.uci.cu de la universidad.

Tabla. 2: Descripción del proceso RTF.

<b>Objetivo</b>	Mejorar la calidad de los productos de trabajo teniendo en cuenta la revisión entre colegas.
<b>Evento(s) que lo genera(n)</b>	Solicitud de pruebas de calidad RTF.
<b>Pre condiciones</b>	
<b>Marco legal</b>	Procedimiento para el buen funcionamiento de los productos de software.
<b>Reglas de negocio</b>	
<b>Responsable</b>	Asesor de Calidad del centro.
<b>Clientes internos</b>	Red de centros.
<b>Clientes externos</b>	NA
<b>Entradas</b>	Expediente de proyecto.
<b>Salidas</b>	Expediente de revisión.
<b>Flujo de eventos</b>	
<b>Flujo básico</b>	
1.	Determinar viabilidad: el asesor de calidad determina la correcta disponibilidad de documentación necesaria para planificar las pruebas, además de, el tiempo y los recursos necesarios.
2.	Diseñar la revisión: Para diseñar las revisiones el experto líder tiene en cuenta el paquete de documentación requerida, la cantidad de recursos disponibles y la técnica a usar. Luego se asigna a los expertos las tareas a realizar y se precisan las herramientas necesarias para ejecutar el proceso. Seguido a ello es realizada la reunión de inicio donde participan los involucrados en la revisión.
3.	Notificar centro y proyecto: el experto líder notifica la realización de la revisión.
4.	Ejecutar revisión: los expertos ejecutan las revisiones y registran los defectos en la herramienta de gestión de proyecto (GESPRO).
5.	Preparar resultado RTF: El experto líder realiza un análisis de la lista de defectos, plasmada en la Herramienta de Gestión de Proyecto (GESPRO), detectada en la revisión.
6.	Notificar resultado para RTF: El experto líder notifica al jefe de proyecto el resultado de la revisión.
7.	Solucionar hallazgos: La lista de defectos es asignada a los involucrados del proyecto para tomar acciones correctivas y de mejora para un mejor resultado.
8.	Seguimiento a los hallazgos: El experto líder confecciona el Dictamen técnico, plasmando en el documento los defectos detectados en la revisión, a estos defectos se le realiza un seguimiento hasta su solución.
9.	Evaluar RTF: El experto líder, teniendo en cuenta los resultados obtenidos en la revisión, emite una evaluación, asegurándose que el producto de trabajo satisface los criterios de entrada de la revisión entre pares antes de su distribución
10.	Archivar resultados RTF: El experto líder archiva toda la evidencia generada en el proceso de revisión, mediante la herramienta de gestión documental (eXcriba).
11.	Termina el proceso.
<b>Flujos alternos</b>	
<b>1.a Si no es viable aplicar criterios de criticidad</b>	
1.	Abortar revisión si cumple con los criterios para abortar.
2.	Detener revisión si cumple con los criterios para detener.
3.	Termina el proceso.

## 2.4 Patrones de modelado de procesos.

Cada patrón describe un problema que ocurre una y otra vez en el entorno y describe también el núcleo de la solución al problema, de forma que puede utilizarse un millón de veces sin tener que hacer dos veces lo mismo. Existen diferentes tipos de patrones: de diseño, arquitectónicos, de codificación, de análisis y de procesos. Este último describe los diversos casos de flujos de procesos que surgen durante el diseño que resuelven situaciones deseadas y, en muchos casos, necesarias. Además de proveer independencia de la implementación tecnológica (Garzás, 2014). A continuación se describen los patrones usados durante el modelado del proceso RTF de la DCSW.

### Patrón de secuencia

Este patrón es utilizado para modelar dependencia entre tareas, es decir, una tarea no puede empezar hasta que otra no haya terminado (ejecución en secuencia). Este patrón se evidencia en la relación entre las tareas diseñar revisión y notificar centro y proyecto. Ver figura 3.

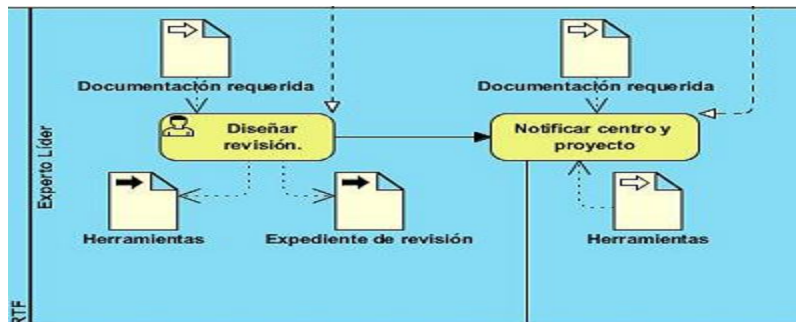


Fig. 2: Patrón de modelado secuencia.

### Patrón de decisión exclusiva

La decisión exclusiva representa un punto en el proceso donde se debe escoger un solo camino de varios disponibles dependiendo de una decisión o de datos del proceso (BizAgi, 2014). Puede ser modelado a través de una compuerta exclusiva. Este patrón se evidencia entre las tareas determinar viabilidad y aplicar criterios de criticidad. Ver figura 4.

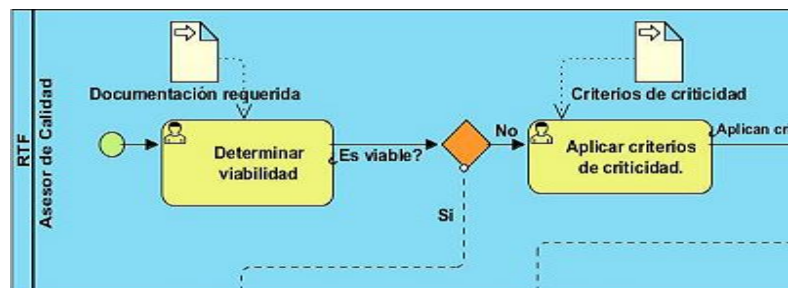


Fig. 3: Patrón de modelado decisión exclusiva.

### **2.5 Requisitos del sistema.**

Los requisitos para un sistema son las descripciones de lo que el sistema debe hacer, los servicios que brinda y las limitaciones de su funcionamiento (Sommerville, 2011).

Los requerimientos permiten que los analistas describan los requisitos del cliente para que los desarrolladores entiendan lo que deben implementar. También, indican a los diseñadores qué funcionalidad y que características va a tener el sistema resultante (Gómez, 2011).

El análisis de requerimientos proporciona una vía para que los clientes y los desarrolladores lleguen a un acuerdo sobre lo que debe hacer el sistema. La especificación, producto de este análisis proporciona las pautas a seguir a los diseñadores del sistema. La captura y el análisis de los requerimientos del sistema es una de las fases más importantes para que el proyecto tenga éxito.

#### **2.5.1 Técnicas para la captura de requisitos.**

La captura de los requisitos tiene como objetivo principal la comprensión de lo que los clientes y los usuarios esperan que haga el sistema (Gómez, 2011). Para la captura de los requisitos existen diferentes técnicas entre ellas las entrevistas, los cuestionarios, encuestas, las técnicas grupales de creatividad, tales como: tormenta de ideas, mapas conceptuales, entre otras. Para el levantamiento de requisitos de la propuesta de solución se aplica la técnica de la entrevista.

Una entrevista es una manera formal o informal de obtener información de los interesados, a través de un diálogo directo con ellos. Se lleva a cabo habitualmente realizando preguntas, preparadas o espontáneas y registrando las respuestas. Las entrevistas se realizan a menudo de manera individual entre un entrevistador y un entrevistado, pero también pueden implicar a varios entrevistadores y/o entrevistados. Entrevistar a participantes con experiencia en el proyecto, a patrocinadores y otros ejecutivos, así como a expertos en la materia, puede ayudar a identificar y definir las características y funciones esperadas de los entregables del producto. Las entrevistas también son útiles para obtener información confidencial (Project Management Institute, 2013).

Se realizó una entrevista a la planificadora de la DCSW y a través de un conjunto de preguntas preparadas se lograron identificar los requisitos que debería cumplir la personalización. Además las respuestas obtenidas en la entrevista permitieron comprender mejor como se realiza el proceso RTF (Ver Anexo 1).

#### **2.5.2 Requisitos funcionales:**

Describen la funcionalidad o los servicios que se espera que éste proveerá. Son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares (Gómez, 2011).

A continuación se muestran los requisitos funcionales obtenidos:



**RF\_1:** Iniciar flujo de trabajo RTF.

- ❖ **Descripción:** El sistema debe permitir iniciar los flujos de trabajos RTF que se deseen realizar.
- ❖ **Prioridad:** Alta

**RF\_2:** Aprobar el expediente de proyecto.

- ❖ **Descripción:** El sistema debe permitir a los usuarios aprobar el expediente de proyecto.
- ❖ **Prioridad:** Alta

**RF\_3:** Rechazar el expediente de proyecto.

- ❖ **Descripción:** El sistema debe permitir a los usuarios rechazar el expediente de proyecto, esto ocurre cuando se encuentran defectos o no conformidades.
- ❖ **Prioridad:** Alta

**RF\_4:** Crear dictamen técnico a partir de una plantilla.

- ❖ **Descripción:** El sistema debe permitir a los usuarios crear el dictamen técnico a partir de una plantilla, en caso que se requiera este documento.
- ❖ **Prioridad:** Alta

**RF\_5:** Crear minuta de reunión de inicio a partir de una plantilla.

- ❖ **Descripción:** El sistema debe permitir crear este documento a partir de una plantilla.
- ❖ **Prioridad:** Alta

**RF\_6:** Crear minuta de reunión de cierre a partir de una plantilla.

- ❖ **Descripción:** El sistema debe permitir crear este documento a partir de una plantilla.
- ❖ **Prioridad:** Alta

**RF\_7:** Crear listas de hallazgos a partir de una plantilla.

- ❖ **Descripción:** El sistema debe permitir crear estos documentos a partir de una plantilla.
- ❖ **Prioridad:** Alta

**RF\_8:** Iniciar flujo de trabajo RTF planificado.

- ❖ **Descripción:** El sistema debe iniciar de forma automática, verificando todos los días a las 6:00 am si existe una nueva RTF planificada. Luego debe notificar al grupo de revisores que deben realizar una nueva revisión.

- ❖ **Prioridad:** Alta

**RF\_9:** Cancelar flujo de trabajo RTF.

- ❖ **Descripción:** El sistema debe permitir al administrador cancelar el flujo de trabajo que se ha iniciado.
- ❖ **Prioridad:** Alta

**RF\_10:** Añadir documentos al flujo de trabajo RTF.

- ❖ **Descripción:** El sistema debe permitir seleccionar uno o varios documentos para que sea añadido al flujo de trabajo RTF.
- ❖ **Prioridad:** Alta

**RF\_11:** Quitar documentos del flujo de trabajo RTF.

- ❖ **Descripción:** El sistema debe permitir quitar el documento seleccionado del flujo de trabajo RTF.
- ❖ **Prioridad:** Alta

**RF\_12:** Asignar permisos a los usuarios.

- ❖ **Descripción:** El sistema debe permitir asignarles permisos a los usuarios que intervienen en el flujo de trabajo RTF.
- ❖ **Prioridad:** Alta

**RF\_13:** Notificar nueva revisión RTF.

- ❖ **Descripción:** El sistema debe notificar a los usuarios la nueva revisión RTF a realizar.
- ❖ **Prioridad:** Alta

### 2.5.3 Requisitos no funcionales:

Los requisitos no funcionales complementan a los funcionales y describen las condiciones ambientales y las cualidades necesarias para que el producto sea eficaz. A continuación se muestran los requisitos no funcionales obtenidos:

#### Restricciones de diseño:

- ❖ **RNF\_1:** El lenguaje de programación que se utilizará para el desarrollo de la personalización será: javascript 1.6 y el lenguaje de marcado xml 1.0.

**Interfaz:**

- ❖ **RNF\_2:** La personalización debe contar con una interfaz gráfica bien definida que permita a los usuarios interactuar de forma cómoda facilitando el trabajo con la herramienta.

**Legales:**

- ❖ **RNF\_3:** Se utilizarán herramientas de software libre y código abierto, o que funcionen bajo las licencias GNU/GPL, por lo que el sistema será desarrollado también en los términos de la licencia GNU/GPL.

**Requerimiento de software:**

- ❖ **RNF\_4:** Las computadoras deben tener instalado un navegador web, se recomienda Mozilla Firefox.

**Confiabilidad:**

- ❖ **RNF\_5:** En la personalización se imponen campos obligatorios para garantizar la integridad de la información que se introduce.

**2.6 Descripción de requisitos por procesos.**

A continuación se muestran la descripción del requisito Iniciar flujo de trabajo RTF asociados al proceso RTF. Consultar las restantes especificaciones de requisito en el anexo # 3.

Tabla.3: Especificación de requisito Iniciar flujo de trabajo RTF.

<b>Precondiciones</b>	Un servicio informático GET realizado por la Dirección de Producción de Software que permita consumir los datos de la revisión RTF planificada.
<b>Flujo de eventos</b>	
<b>Flujo básico Iniciar flujo de trabajo RTF</b>	
1.	El experto líder (usuario) se autentica en el sistema.
2.	El sistema muestra el panel de inicio.
3.	El experto líder selecciona el panel de tareas.
4.	El sistema muestra las opciones Mis tareas y Flujos de trabajo que he iniciado.
5.	El experto líder selecciona la opción flujos de trabajo que he iniciado.
6.	El sistema muestra la opción Iniciar un flujo de trabajo.
7.	El experto líder selecciona la opción Iniciar flujo de trabajo.
8.	El experto líder selecciona el flujo de trabajo Revisiones Técnicas Formales.
9.	El sistema muestra la interfaz del proceso con los campos a llenar.
10.	El experto líder llena los campos mostrados.
11.	EL experto líder selecciona el botón "Iniciar flujo de trabajo".
12.	El sistema muestra el flujo de trabajo iniciado.
13.	Concluye el requisito.
<b>Pos-condiciones</b>	

1.	N/A	
<b>Flujos alternativos</b>		
<b>Flujo alternativo</b>		
1.	N/A	
<b>Pos-condiciones</b>		
1	N/A	
<b>Validaciones</b>		
1	N/A	
<b>Conceptos</b>	<b>Iniciar flujo de trabajo RTF</b>	Visibles en la interfaz: <ul style="list-style-type: none"> <li>- Comentario</li> <li>- Elementos adjuntos</li> </ul> Utilizados internamente <ul style="list-style-type: none"> <li>- Fecha de inicio</li> <li>- Tipo</li> <li>- Descripción</li> </ul>
<b>Requisitos especiales</b>	N/A	
<b>Asuntos pendientes</b>	N/A	
<b>Prototipo elemental de interfaz gráfica de usuario</b>		
		

## 2.7 Validación de los requisitos.

La validación de los requisitos es el proceso de verificar que los requisitos realmente definen el sistema que el cliente realmente quiere. Es importante porque los errores en un documento de requisitos pueden conducir a costos de trabajos extensos, cuando estos problemas se descubren durante el desarrollo o después de que el sistema está en servicio (Sommerville, 2011).

Existen varias técnicas de validación de requerimientos, entre ellas:

- ❖ **Revisiones de requerimientos**

Son un proceso manual en la que intervienen tanto el cliente como personal involucrado en el desarrollo del sistema, ésta puede ser formal o informal, y tiene el fin de verificar que el documento de requerimientos no presente anomalías ni omisiones.

### ❖ Construcción de prototipos

Consiste en mostrar un modelo ejecutable del sistema a los usuarios finales y a los clientes, así éstos pueden experimentar con el modelo para ver si cumple con sus necesidades reales.

### ❖ Generación de casos de prueba

La generación de los casos de prueba consiste en probar los requisitos. Si una prueba es difícil o imposible de diseñar, normalmente significa que los requerimientos serán difíciles de implantar y deberían ser considerados nuevamente.

De las técnicas de validación de requisitos enunciadas anteriormente se aplicó la construcción de prototipos. Esta permitió verificar que se cumplen las necesidades del cliente. A continuación se muestra el prototipo del RF\_1. Consultar los restantes prototipos en los anexos.

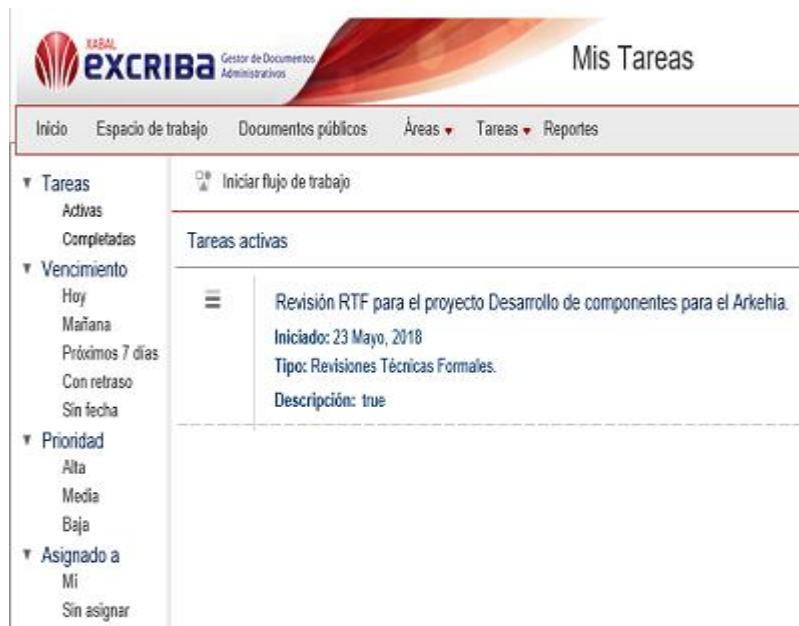


Fig. 4: Prototipo del requisito funcional Iniciar flujo de trabajo RTF.

## 2.8 Matriz de trazabilidad.

La matriz de trazabilidad de requisitos es un cuadro que vincula los requisitos del producto desde su origen hasta los entregables que los satisfacen. Proporciona un medio para realizar el seguimiento de los requisitos a lo largo del ciclo de vida del proyecto, lo cual contribuye a asegurar que al final del proyecto se entreguen

efectivamente los requisitos aprobados en la documentación de requisitos (Project Management Institute, 2013). A continuación se muestra la matriz de trazabilidad requisito- requisito.

(13) Requirement	Aprobar el expediente de ...	Asignar permisos a los usu...	Añadir documentos al flujo...	Cancelar flujo de trabajo ...	Crear dictamen técnico	Crear lista de hallazgos	Crear minuta de reunión d...	Crear minuta de reunión d...	Iniciar flujo de trabajo RTF	Iniciar flujo de trabajo RTF...	Notificar nueva revisión RTF	Quitar documentos del flujo...	Rechazar el expediente de...
Aprobar el expediente de ...							«d»	«d»					«d»
Asignar permisos a los usu...							«d»				«d»		
Añadir documentos al flujo...							«d»						
Cancelar flujo de trabajo ...							«d»						
Crear dictamen técnico						«d»	«d»						
Crear lista de hallazgos	«d»				«d»								«d»
Crear minuta de reunión d...				«d»	«d»								
Crear minuta de reunión d...	«d»	«d»	«d»										«d»
Iniciar flujo de trabajo RTF											«d»		
Iniciar flujo de trabajo RTF...											«d»		
Notificar nueva revisión RTF		«d»							«d»	«d»			

Fig. 5: Matriz de trazabilidad requisito – requisito.

### 2.9 Modelo conceptual.

El modelo conceptual o modelo de dominio es una representación visual estática del entorno real. Es un diagrama con los objetos que forman parte del proyecto a desarrollar y las relaciones que hay entre ellos, pero no son clases de software (Martínez, 2010). Este artefacto se realiza debido a que se va a utilizar en la implementación del modelo de contenido.

A continuación se presenta el modelo conceptual del proceso RTF.

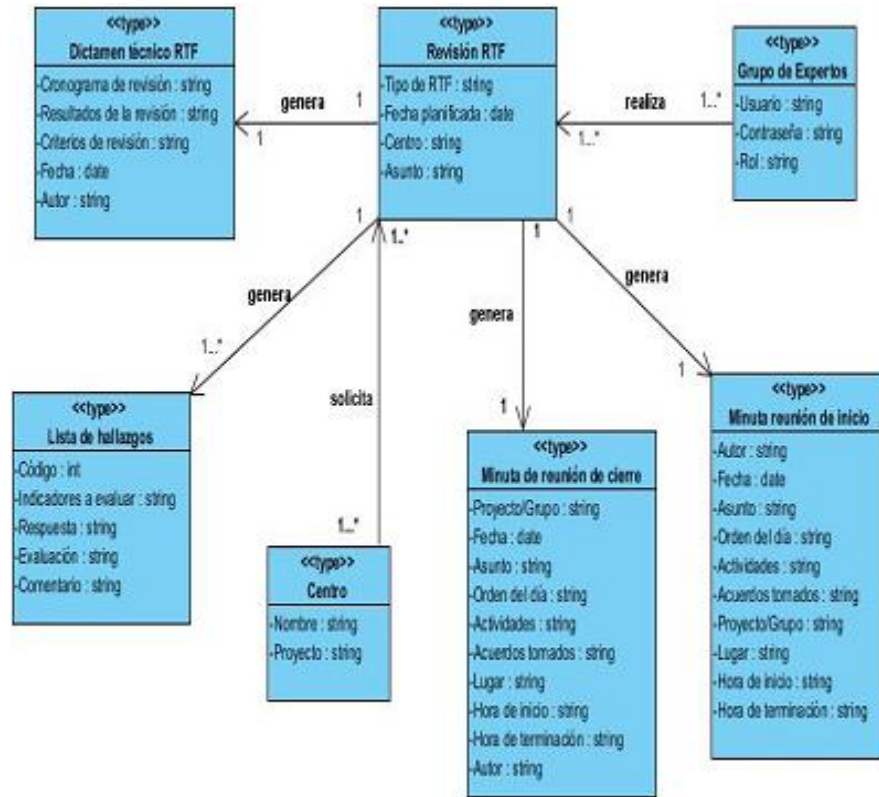


Fig. 6: Modelo conceptual.

**Diccionario de datos:**

Este artefacto define un conjunto de tipos de datos soportados por defecto, es decir, se pueden manejar textos, números, fechas, booleanos, sin necesidad de tener que definir nuevos tipos (Revelles, 2014). Consultar los restantes diccionarios de datos en el anexo # 2.

Tabla. 4: Diccionario de datos de RTF

Descripción	Prueba de calidad que se le realiza a los productos.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas
Tipo de RTF	Tipo de prueba RTF.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Números, otros caracteres especiales
Fecha	Fecha planificada para la RTF.	Fecha	No	Si	DD/MM/AAAA	Fecha diferente a la actual.
Centro	Centro al que se le va a realizar la RTF.	Cadena de caracteres	No	Si	A-Z, a-z, espacio	Números, otros caracteres especiales

Asunto	Descripción de RTF.	Cadena de caracteres	No	No	A-Z, a-z, espacio	Otros caracteres especiales
--------	---------------------	----------------------	----	----	-------------------	-----------------------------

### 2.10 Arquitectura del software.

La arquitectura de software alude a la estructura general de este y a las formas en la que esta da integridad conceptual a un sistema. En su forma más sencilla, la arquitectura es la estructura de organización de los componentes de un programa, la forma en la que estos interactúan y la estructura de datos que utilizan (Pressman, 2010).

#### Patrón de arquitectura.

Se relacionan con el diseño a gran escala y de grano grueso, y que se aplican típicamente durante las primeras iteraciones (la fase de elaboración) cuando se establecen las estructuras y conexiones más importantes (Pressman, 2010).

Para el desarrollo de la solución se utilizará la arquitectura en capas. El GDA Xabal eXcriba tiene definidas 3 capas las cuales son: Presentación, Web y Repositorio.

Capa de Presentación: En esta capa el cliente y la aplicación establecen comunicación mediante el conjunto de interfaces de usuarios que están presente en dicha capa.

Capa de Web: En esta capa se ejecutan todos los procesos de negocio que han sido previamente implementados, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados. Además se comunica con la capa de acceso a repositorio mediante un subsistema de servicios, el cual es el encargado de realizar las llamadas a los servicios.

Capa de Repositorio: En esta capa es donde se realiza la implementación de los servicios, los cuales son necesarios para gestionar los datos del repositorio. Entre los servicios que se implementan se encuentran los servicios de control, tales como la implementación de los flujos de trabajo. También están presente los servicios de búsquedas, entre ellas las búsquedas a texto completo y los servicios de contenidos entre ellos las reglas y las acciones. Además están presente las API javascript que permiten acceder a los servicios base. A continuación se muestra la arquitectura en capas de la propuesta de solución.



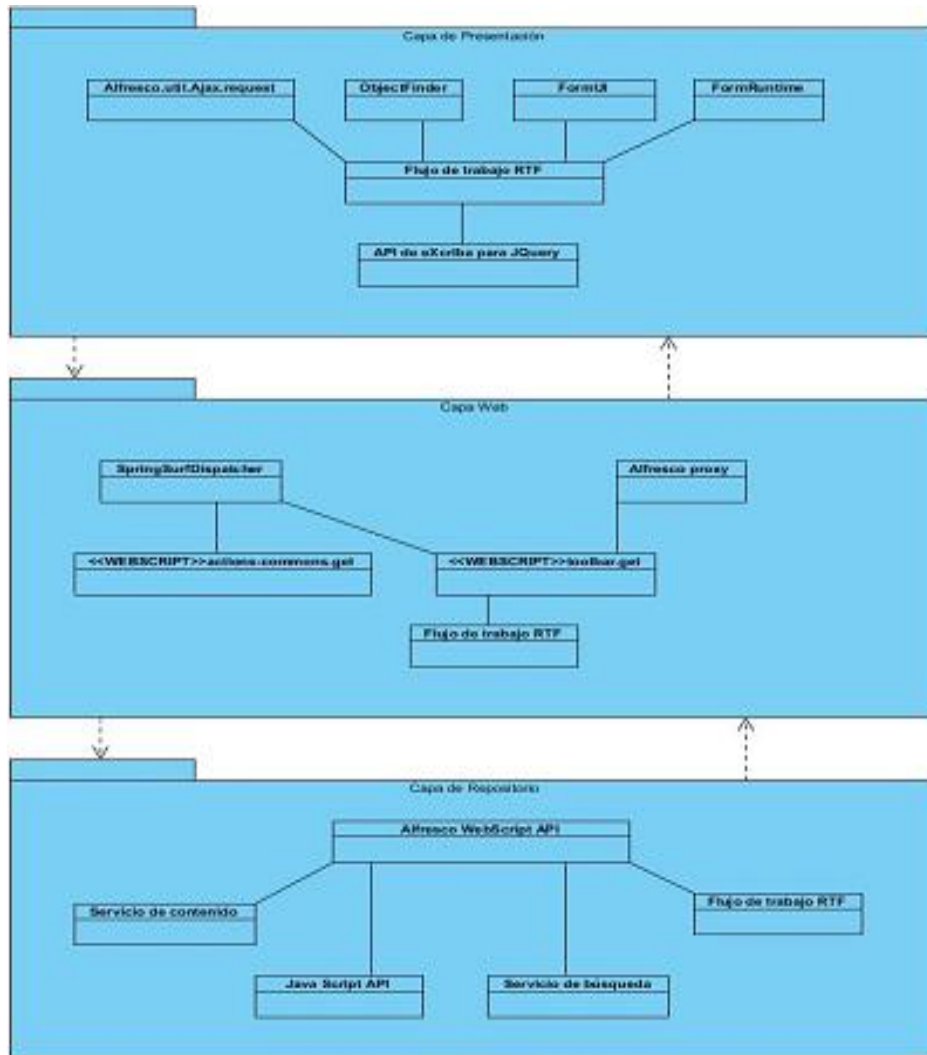


Fig. 7: Arquitectura de la propuesta de solución.

### 2.11 Patrones de diseño.

Un patrón es una forma de dar solución a un problema, con un nombre y que es aplicable a otros contextos, cuenta con una sugerencia sobre la manera de usarlo en situaciones nuevas. Los patrones de diseño son el esqueleto de las soluciones a problemas comunes en el desarrollo de software (Larman, 2003).

Los patrones que se utilizan en la propuesta de solución son los GRASP (Patrones de Principios Generales para Asignar Responsabilidades). Estos patrones describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

❖ Experto

Asigna una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir con la responsabilidad. Se conserva el encapsulamiento y el comportamiento se distribuye entre todas las clases implicadas en la ejecución de la función. Se soporta normalmente una alta cohesión (Larman, 2003). Este patrón se evidencia en la realización del fichero RevisionesTécnicasFormales.bpmn20.xml, el cual contiene toda la información referente a las tareas del proceso.

### ❖ Bajo acoplamiento

Asignar una responsabilidad de manera que el acoplamiento permanezca bajo. El acoplamiento es una medida de la fuerza con que un elemento está conectado a, tiene conocimiento de, confía en, otros elementos. Un elemento con bajo (o débil) acoplamiento no depende de otros elementos, los cuales pueden ser clases, subsistemas y sistemas.

Con este patrón las clases no se afectan por cambios de otros componentes. Son fáciles de entender por separado y fáciles de reutilizar (Larman, 2003). Este patrón se evidencia en el modelo de contenido, debido a que este sólo depende del fichero que contiene la definición del proceso.

### ❖ Alta cohesión

Asignar una responsabilidad de manera que la cohesión permanezca alta. Un elemento con responsabilidades altamente relacionadas, y que no hace una gran cantidad de trabajo, tiene alta cohesión. Estos elementos pueden ser clases, subsistemas y otros. Este patrón incrementa la claridad y facilita la comprensión del diseño. Se simplifican el mantenimiento y las mejoras. Se soporta a menudo bajo acoplamiento (Larman, 2003). Este patrón se evidencia en el fichero RevisionesTecnicasFormales-generic-context.xml el cual carga las configuraciones personalizadas de los flujos de trabajo y los modelos de contenidos personalizados.

### ❖ Controlador

Es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control (Larman, 2003). Este patrón se evidencia en la realización del fichero RevisionesTécnicasFormales.bpmn20.xml, el cual contiene toda la información referente a la definición del proceso.

## **2.12 Conclusiones parciales.**

En este capítulo se arribaron a las siguientes conclusiones:

- ❖ La realización del modelo conceptual proporcionó una mayor comprensión de los conceptos asociados al proceso RTF.
- ❖ La elaboración del diagrama de proceso de negocio permitió comprender con claridad el funcionamiento del proceso RTF en la universidad.
- ❖ Mediante el levantamiento de requisitos se determinaron las funcionalidades que la personalización del GDA Xabal eXcriba 3.1 debe cumplir.
- ❖ Se diseñaron los prototipos de interfaz con el fin de verificar si la personalización del GDA Xabal eXcriba 3.1 cumplirá con las necesidades del cliente.
- ❖ Mediante la arquitectura de software se estableció la estructura de la personalización y la forma en que sus componentes trabajan.
- ❖ El empleo de patrones de diseño garantizará una solución que tiene como premisa la reutilización de código y la solución a problemas.

### **Capítulo 3: Implementación y pruebas.**

En este capítulo se presenta la implementación y validación del Trabajo de Diploma, se establece la tipología de los documentos pertenecientes a la DCSW de la UCI, la política de permisos y los grupos de usuarios del sistema. Se define la estructura organizativa de los documentos de la DCSW y la tabla de acceso y seguridad. Por último se mostrarán los procesos modelados en Activiti así como la configuración del flujo de trabajo.

#### **3.1 Tipología de los documentos.**

La tipología de los documentos es una descripción de estructura y sintaxis de un documento, su objetivo es determinar la estructura de un documento. Definir la tipología documental para el desarrollo de la presente investigación tiene gran importancia debido a que dentro de los metadatos que sirven de búsqueda para los documentos, se encuentra el tipo de contenido, en este caso, lo único que se maneja en el sistema es documentos, debido a que cada tipo de documento constituye un tipo de contenido, a continuación se muestran algunas de las tipologías existentes:

- ❖ Minuta de reunión de inicio.
- ❖ Minuta de reunión de cierre.
- ❖ Dictamen técnico RTF.
- ❖ Lista de chequeo de arquitectura.
- ❖ Lista de chequeo de base de datos.
- ❖ Lista de chequeo de requisitos.
- ❖ Lista de chequeo de planificación.

##### **3.1.1 Tipificación de los documentos.**

La tipificación es el estudio de los tipos o clases que se encarga en diversos campos de estudio, de realizar una clasificación de diferentes elementos. El término también se puede referir a los tipos de letra, utilizados en cualquier soporte escrito (RAE, 2017). Según el documento Estándar de configuración del programa de mejora la nomenclatura de los documentos queda de la siguiente forma.

Nomenclatura definida para las RTF:

RTF- XX - Tipo

Tipo de RTF: planificación, requisito, arquitectura, base de datos.

XX: año.

La nomenclatura de los documentos generados de la ejecución del proceso estaría dada de la siguiente forma:

AAAA - MM - DD_RTF - XX - Tipo_Nombre del documento	AAAA – MM – DD: Fecha XX: año Tipo: Tipo de RTF
---	---

### 3.2 Estructura organizativa de los documentos.

La estructura organizativa define como van a estar ubicados los documentos en el GDA XABAL eXcriba 3.1. Dicha estructura contendrá todos los directorios del expediente de proyecto relativos a la documentación y ficheros de modelado. A continuación se muestra un fragmento de la estructura organizativa de los documentos.

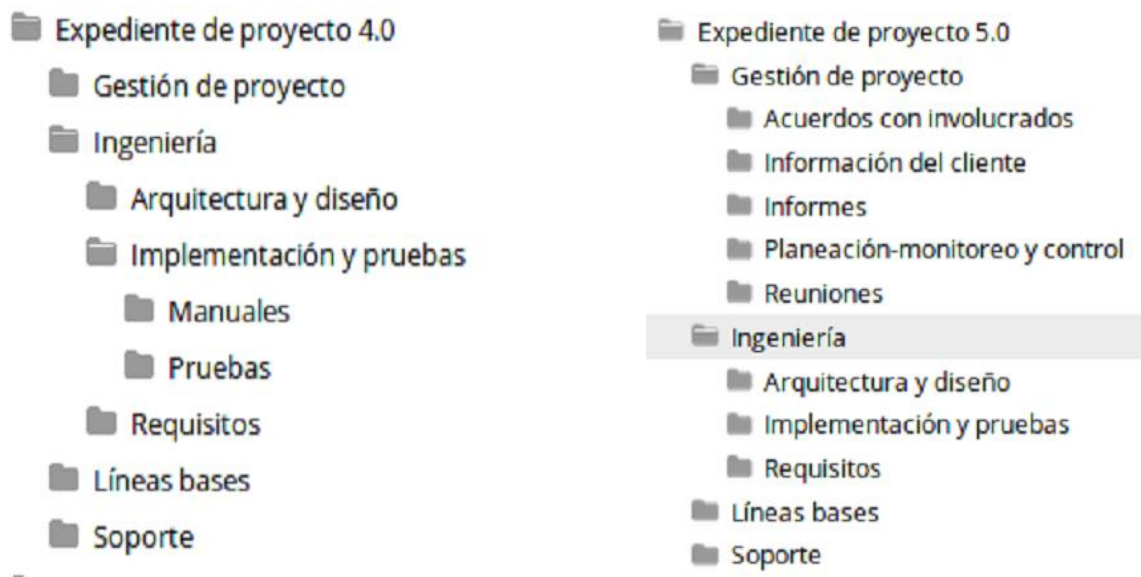


Fig. 8: Estructura organizativa de los documentos.

Estos expedientes de proyecto son utilizados debido a que va a hacer una copia del expediente de proyecto para el espacio de trabajo de los revisores. Esta copia del expediente de proyecto permitirá que no sean modificados los documentos originales del proyecto. Además los documentos generados en la revisión serán guardados en la siguiente dirección: Expediente de proyecto/Soporte/Aseguramiento de la calidad/Evaluaciones/RTF.

**3.3 Tabla de acceso y seguridad.**

Las tablas de acceso y seguridad identifican los derechos y las restricciones de acceso (creación, consulta, modificación, eliminación) de los miembros de la entidad en relación con los documentos.

Tabla. 5: Tabla de acceso y seguridad.

Tabla de acceso y seguridad			
No.	Documento	Archivo de gestión	
		Permisos	Rol
1	Dictamen_RTF	Consumidor	Revisores de calidad
			Administrador de la configuración
			Analista
			Arquitecto de información
			Arquitecto de software
			Planificador
			Programador
			Diseñador de BD
			Diseñador de pruebas
2	Minuta de reunión de apertura	Consumidor	Revisores de calidad
			Administrador de calidad
			Administrador de la configuración
			Analista
			Arquitecto de información
			Arquitecto de software
			Planificador
			Diseñador de pruebas
			Programador
Diseñador de BD			
3	Minuta de reunión de cierre	Consumidor	Revisores de calidad
			Administrador de calidad
			Administrador de la configuración

			Analista
			Arquitecto de información
			Arquitecto de software
			Planificador
			Diseñador de pruebas
			Programador
			Diseñador de BD

**3.4 Configuración de los grupos y permisos.**

Para facilitar la asignación de los permisos a las diferentes carpetas y documentos de la estructura establecida, se creó un sistema de grupos de usuarios.

**Grupos:**

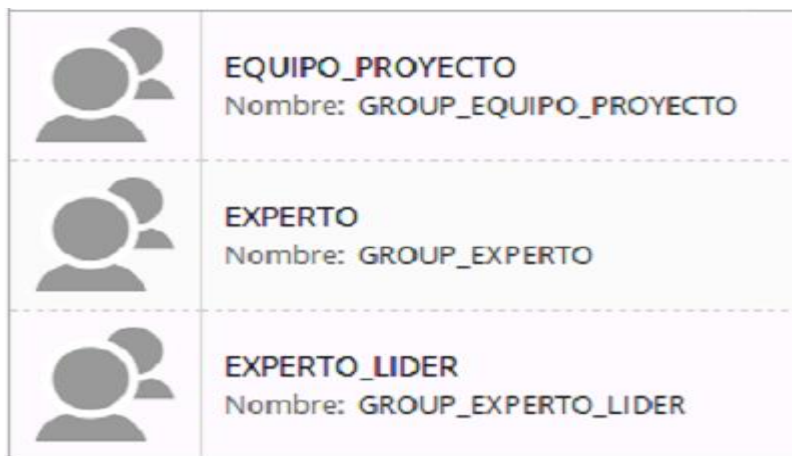


Fig. 9: Grupos definidos.

**Permisos:**

En aras de garantizar la seguridad y fiabilidad de la documentación existente se utilizaron los permisos que propone el sistema para gestionar el rol que tendrá asociado una persona sobre un documento y en función de ese rol serán las acciones que podrán realizar. Los permisos son asignados de forma manual por el administrador del sistema a los distintos usuarios del área, a continuación se ilustran los permisos definidos:

Tabla.6: Permisos.

Permisos	Coordinador	Contribuidor	Colaborador	Editor	Consumidor
Leer contenido	X	X	X	X	X
Editar contenido	X		X		
Crear contenido	X	X	X		
Editar propiedades	X		X	X	
Copiar contenido	X	X	X	X	X
Actualizar contenido	X				

Se le otorgó a los grupos de usuarios definidos el permiso de consumidor. Estos permisos fueron otorgados teniendo en cuenta el documento Expedientes 4.0-5.0\_Artefactos y Roles publicado en el sitio mejoras.prod.uci.cu de la universidad.

### **3.5 Configuración del flujo de trabajo y modelado del proceso en Activiti.**

En este epígrafe se define el flujo de trabajo del proceso a informatizar, además se muestra el proceso modelado en Activiti y los ficheros de configuración que se realizan.

Los flujos de trabajo se definen como un sistema de secuencia de tareas de un proceso de negocio. Organiza y controla tareas, recursos y reglas necesarias para completar el proceso de negocio. Los flujos de trabajo consisten en reducir el tiempo y acelerar la realización de un trabajo mediante el acercamiento de procesos, personas y maquinas, incluso permitiendo trabajar en equipo desde diferentes lugares (Potts, 2012).

#### **3.5.1 Configuración del flujo de trabajo.**

Para la informatización del proceso RTF fue necesaria la implementación del flujo de trabajo en la herramienta Activiti Designer. Para modelar el proceso se deben identificar los patrones que acepta la herramienta, modelando solamente las actividades a informatizar. A continuación se muestra el modelado del proceso en la herramienta.



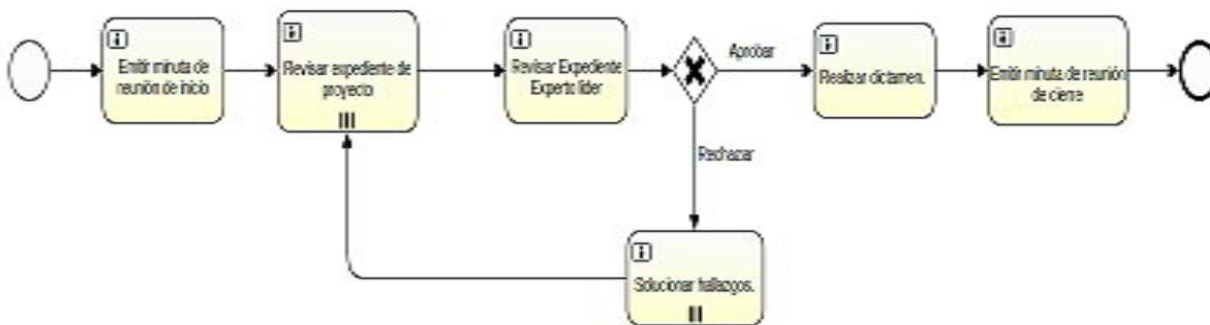


Fig. 10: Modelado del flujo de trabajo RTF.

Luego de realizar el modelado del flujo de trabajo se configuran las propiedades del proceso y de cada una de sus actividades. A continuación se muestran las configuraciones realizadas a cada uno de los elementos del flujo de trabajo.

Tabla. 7: Descripción del flujo de trabajo.

Evento/ Tarea/ Flujo de Secuencia	Propiedades	
	Main Config	Listeners
Start ( startevent1)	Form key: rtf: inicio	
Emitir minuta de reunión de inicio ( usertask1)	Form key: rtf:emitirmri Candidate groups: GROUP_EXPERTO_LIDER	Event = complete task.setVariable('bpm_comment', execution.getVariable('bpm_comment'));
Revisar expediente de proyecto ( usertask2)	Form key: rtf:revisarexpproy Candidate users: \${responsible}	Event = complete var permission = [['Coordinator','GROUP_EXPERTO_LIDER']]; setPermissionNode(permission); execution.setVariable('bpm_comment', task.getVariable('bpm_comment'));
Realizar dictamen ( usertask3)	Form key: rtf:realizardictamen Candidate groups: GROUP_EXPERTO_LIDER	Event = complete task.setVariable('bpm_comment', execution.getVariable('bpm_comment'));

### Capítulo 3: Implementación y pruebas.

<p>Emitir minuta de reunión de cierre ( usertask4)</p>	<p>Form key: rtf:emitirmrc</p> <p>Candidate groups: GROUP_EXPERTO_LIDER</p>	<pre>Event = complete var searchElemts= new java.util.ArrayList(); searchElemt=search.luceneSearch('PATH:"'+bpm_ package.children[0].parent.qnamePath+'/*' AND @cm\\:name:"Evaluaciones"); var f = new Date(); var anno = f.getFullYear(); var mes = f.getMonth()+1; var dia = f.getDate(); if(searchElemts != null &amp;&amp; searchElemts.length &gt; 0){ var nameFolder = anno+"-"+mes+"-"+día+ "- RTF"; var carpeta = searchElemts[0].createFolder(nameFolder); for(var j=0; j &lt; bpm_package.children.length; j++){ if(bpm_package.children[j].hasAspect("rtf:evaluaci ones") == true){ bpm_package.children[j].copy(carpeta); }} }</pre>
<p>Solucionar hallazgos ( usertask6)</p>	<p>Form key: rtf:solucionarhallazgos</p> <p>Candidate users: \${equipo}</p>	<pre>Event = complete task.setVariable('bpm_comment', execution.getVariable('bpm_comment')); var newTaskCount = taskdone + 1; execution.setVariable('taskdone', newTaskCount); var newCount1 = count1 + 1; execution.setVariable('count1', newCount1);  execution.setVariable('approveCount', 0); execution.setVariable('count', 0); execution.setVariable('bpm_comment', task.getVariable('bpm_comment'));</pre>
<p>Revisar Expediente Experto líder (usertask7)</p>	<p>Form key: rtf:revisarexplíder</p> <p>Candidate groups: GROUP_EXPERTO_LIDER</p>	<pre>Event = complete if(task.getVariable('rtff_approveRejectOutcome')==</pre>

		<pre>'Aprobar'){     execution.setVariable('resultado', true); } }else{     execution.setVariable('resultado', false);     var permission = [[['Consumer','GROUP_EQUIPO_PROYECTO']]];     setPermissionNode(permission); } execution.setVariable('bpm_comment', task.getVariable('bpm_comment'));</pre>
Aprobar (sequenceFlow4)	( Condition: \${resultado==true}	
Rechazar (sequenceFlow8)	Condition: \${resultado==false}	

A continuación se especifica un poco más la lógica del proceso modelado para lograr un mejor entendimiento.

Primeramente se configuran las propiedades del proceso que serían el nombre Revisiones técnicas formales y su id correspondiente (RevisionesTécnicasF). Después se pasa a configurar cada una de las tareas y eventos del proceso, a continuación se ilustran:

❖ Start

Al evento de inicio (Star) se le asigna en la propiedad Form Key el valor rtf: inicio pues el proceso inicia cuando se realiza una nueva petición de prueba RTF.

❖ Tarea Emitir minuta de reunión de inicio.

En el Main Config de la tarea se le asigna a la propiedad Candidate Groups el valor GROUP\_EXPERTO\_LIDER y a Form Key el valor rtf: emitirmri. En el Task Listeners, que son las acciones a realizar en la tarea dependiendo a determinadas condiciones, el valor dado a la propiedad event es complete, pues la acción se realizará al finalizar la tarea, y el script que se ejecutará será el siguiente:

```
task.setVariable ('bpm_comment', execution.getVariable ('bpm_comment'));
```

En este código se capturan los datos de la variable local de la tarea y luego son modificados por los datos de la variable global.

❖ Tarea Revisar expediente de proyecto.

En el Main Config de la tarea se le asigna a la propiedad Candidate Users el valor `${responsable}` y a Form Key el valor `rtf: revisarexproy`. En el Task Listeners el valor dado a la propiedad event es `complete` y el script a ejecutar será el siguiente:

```
var permission = [['Coordinator','GROUP_EXPERTO_LIDER']];
    setPermissionNode (permission);
    execution.setVariable ('bpm_comment', task.getVariable ('bpm_comment'));
```

Esta es una tarea multi - instancia que permite ejecutar la misma para cada elemento de una colección, en este caso la colección es `${responsable}`. En este código se les da permisos de coordinador a los usuarios que pertenecen al grupo experto líder, para que puedan acceder a los documentos adjuntados.

### ❖ Tarea Solucionar hallazgos.

En esta tarea se configura la propiedad Form Key con la expresión `rtf: solucionarhallazgos` y a Candidate users el valor de la colección `${equipo}`. Esta colección guarda todos los usuarios que pertenecen al grupo EQUIPO\_PROYECTO. En el Task Listeners, el valor dado a la propiedad event es `complete` y el script a ejecutar será el siguiente:

```
task.setVariable ('bpm_comment', execution.getVariable ('bpm_comment'));
var newTaskCount = taskdone + 1;
execution.getVariable ('taskdone', newTaskCount);
var newCount1 = count1 + 1;
execution.getVariable ('count1', newCount1);

execution.getVariable ('approveCount', 0);
execution.getVariable ('count', 0);
execution.getVariable ('bpm_comment', task.setVariable ('bpm_comment'));
```

A l igual que la tarea anterior, esta es una tarea multi – instancia. En este código se verifica a través de un contador la cantidad de usuarios que realizaron esta tarea.

### ❖ Tarea Realizar dictamen técnico

En el Main Config de la tarea se le asigna a la propiedad Form key el valor de `rtf: realizardictamen` y en el Task Listeners, el valor dado a la propiedad event es `complete` y el script a ejecutar será el siguiente:

```
task.setVariable ('bpm_comment', execution.getVariable ('bpm_comment'));
```

En este código se capturan los datos de la variable local de la tarea y luego son modificados por los datos de la variable global.

### ❖ Tarea Revisar Expediente Experto líder

En el Main Config de la tarea Revisar Expediente Experto líder, se le asigna a la propiedad Candidate groups: GROUP\_EXPERTO\_LIDER y a Form Key el valor rtf: revisarexplíder. En el Task Listeners, el valor dado a la propiedad event es complete, y el script a ejecutar será el siguiente:

```
if (task.getVariable('rtff_approveRejectOutcome')== 'Aprobar'){
    execution.setVariable ('resultado', true);
} else {
    execution.setVariable ('resultado', false);
    var permission = [['Consumer','GROUP_EQUIPO_PROYECTO']];
    setPermissionNode (permission);
}
execution.setVariable ('bpm_comment', task.getVariable ('bpm_comment'));
```

En este código se pide el valor de la variable 'rtff\_approveRejectOutcome', y si es Aprobar, se crea una variable de proceso resultado con valor true, en caso de ser Rechazar, se crea la misma variable con valor false y se le asignan permisos de consumidor a los usuarios que pertenecen al grupo equipo de proyecto sobre los documentos adjuntados.

### **Rechazar**

Se le asigna valor a la propiedad condition con el siguiente código: `${resultado == false}`, donde se pregunta si el valor de la variable resultado, creada en la tarea Revisar Expediente Experto líder es igual a false.

### **Aprobar**

Se le asigna valor a la propiedad condition con el siguiente código: `${resultado == true}`, donde se pregunta si el valor de la variable resultado, creada en la tarea Revisar Expediente Experto líder es igual a true.

### ❖ Tarea Emitir minuta de reunión de cierre

En el Main Config de la tarea Emitir minuta de reunión de cierre, se le asigna a la propiedad Candidate groups: GROUP\_EXPERTO\_LIDER y a Form Key el valor rtf: emitirmrc. En el Task Listeners, el valor dado a la propiedad event es complete, y el script a ejecutar será el siguiente:

```
var searchElemnts= new java.util.ArrayList ();
searchElemnts = search.luceneSearch ('PATH:"'+bpm_package.children[0].parent.qnamePath+'/*' AND
@cm\\:name: "Evaluaciones");
    var f = new Date();
    var anno = f.getFullYear();
    var mes = f.getMonth()+1;
    var día = f.getDate();
    if (searchElemnts != null && searchElemnts.length > 0){
    var nameFolder = anno+"-"+mes+"-"+día+ "-RTF";
    var carpeta = searchElemnts[0].createFolder(nameFolder);
    for (var j=0; j < bpm_package.children.length; j++){
        if (bpm_package.children[j].hasAspect("rtf:evaluaciones") == true){
            bpm_package.children[j].copy (carpeta);
        }
    }
}
```

En el código anteriormente expuesto primeramente se crea una variable `searchElemnts`, que guardará la dirección de la carpeta con el nombre Evaluaciones. Después se crea una variable `f` y se le asigna el valor de la fecha actual de la máquina y posteriormente se crean las variables `mes`, `año` y `día` obteniendo estos valores de la variable `f`. Luego se crea la variable `nameFolder`, la cual contiene el nombre de la carpeta donde serán guardados los documentos. El nombre de esta carpeta está compuesto por la fecha en que se realizó la revisión y el nombre RTF. Se crea la variable `carpeta` que va a contener la referencia de la ubicación donde va a ser creada la nueva carpeta con la documentación de la revisión. Por último se copian los documentos adjuntos en la referencia de la dirección que contiene la variable `carpeta`.

Una vez terminada la configuración de los flujos de trabajo en Activiti, el siguiente paso es integrarlos a Alfresco para que este los reconozca.

### 3.5.2 Despliegue del flujo de trabajo en Alfresco.

Luego de configurar las propiedades del proceso el siguiente paso es desplegar el archivo **RevisionesTécnicasFormales.xml** generado, éste es copiado en la carpeta `workflow` ubicada en la siguiente dirección **shared/classes/alfresco/workflow**. Luego de haber copiado el archivo es importante destacar que este tendrá como extensión **RevisionesTécnicasFormales.bpmn**, la que no es interpretada por Alfresco, para que el mismo interprete este fichero es necesario cambiarle la extensión:

**RevisionesTécnicasFormales.bpmn20.xml**. En este fichero se realiza la definición del proceso.

Una vez finalizado el paso anterior se procede a la creación del fichero **RevisionesTécnicasFormales-model.xml**, donde se asocia para cada tarea definida en el flujo, un modelo de contenido, donde se especifican las propiedades y aspectos que luego serán mostrados en el GDA Xabal eXcriba 3.1. Dicho fichero se encuentra en la carpeta `workflow` ubicada en la siguiente dirección: **shared/classes/alfresco/workflow**.

El modelo de contenidos por defecto se conoce en Alfresco como Diccionario de Datos o Data Dictionary y soporta ser extendido para que el repositorio pueda manejar nuevos modelos de contenido. El Diccionario de Datos describe las estructuras del contenido a través de propiedades, asociaciones y reglas o restricciones. El Diccionario de Datos es un metamodelo que permite describir uno o más modelos de contenido específicos (Revelles, 2014).

El siguiente paso es la creación del fichero **RevisionesTécnicasFormales-form-config.xml**, donde se personalizan las propiedades y aspectos definidos en el modelo de contenido, para ser mostradas mediante una interfaz en el GDA Xabal eXcriba 3.1. El mismo se encuentra en la carpeta `web-extension` ubicado en la siguiente dirección **shared/classes/alfresco/web-extension**.

A continuación se crea el fichero **RevisionesTécnicasFormales-message-workflow.properties**, en la carpeta `messages` ubicada en la dirección **shared/classes/alfresco/messages**, en el que se describen las etiquetas a mostrar en las interfaces del flujo. En caso de querer otros idiomas se pone el mismo fichero pero en el nombre le pones las siglas del idioma por ejemplo: **RevisionesTécnicasFormales-message-workflow-es.properties** y así las etiquetas son mostradas en español.

Luego se crea el fichero **RevisionesTécnicasFormales-generic-context.xml**, el cual es usado para cargar las configuraciones personalizadas de los flujos de trabajo y los modelos de contenidos personalizados, la ubicación del fichero es **tomcat/shared/classes/alfresco/extension**. Además se encarga de hacer referencia al modelo de contenido descrito en el fichero **RevisionesTécnicasFormales-model.xml** y al flujo de trabajo **RevisionesTécnicasFormales.bpmn20.xml**.

Por último se crea el fichero **RevisionesTécnicasFormales-context.xml** usado para cargar las vistas personalizadas de cada proceso, la ubicación para este fichero es **tomcat/shared/classes/alfresco/web-extension**. Se encarga de hacer referencia al formulario **RevisionesTécnicasFormales-form-config.xml** y al fichero de mensajes **RevisionesTécnicasFormales-message-workflow.properties**.

A continuación se muestra la implementación de cada uno de estos ficheros para el flujo de trabajo RTF:

- ❖ Fichero **RevisionesTécnicasFormales-model.xml**

En la siguiente figura se muestra el modelo de contenido de la tarea revisar expediente de proyecto.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
Filename: RevisionesTecnicasFormales-model.xml
Description: Modelo de contenido para revisiones tecnicas formales.
Author: Jasmelys Lago Salas.
-->
<model name="rtf:RevisionesTecnicasFormales-model" xmlns="http://www.alfresco.org/model/dictionary/1.0">
  <imports>
    <import uri="http://www.alfresco.org/model/dictionary/1.0" prefix="d"/>
    <import uri="http://www.alfresco.org/model/typ/1.0" prefix="typ"/>
    <!-- Import Alfresco Content Domain Model Definitions -->
    <import uri="http://www.alfresco.org/model/content/1.0" prefix="cm" />
    <import uri="http://www.exoriba.uoi.es/generic_model/workflow/1.0" prefix="rtff" />
  </imports>
  <namespaces>
    <namespace uri="http://www.exoriba.uoi.es/RevisionesTecnicasFormales_model/workflow/1.0" prefix="rtff"/>
  </namespaces>

  <type name="rtf:revisarexpproy">
    <title>Revisar expediente proyecto</title>
    <parent>rtff:reviewGeneric</parent>
  </type>

  <type name="rtff:reviewGeneric">
    <parent>rtff:genericTask</parent>
    <properties>
      <property name="rtff:approveRejectOutcome">
        <type>d:text</type>
        <default>Rechazar</default>
        <constraints>
          <constraint ref="rtff:OutcomeValue" />
        </constraints>
      </property>
    </properties>
    <overrides>
      <property name="typ:outcomePropertyName">
        <default>(http://www.exoriba.uoi.es/generic_model/workflow/1.0)approveRejectOutcome</default>
      </property>
    </overrides>
  </type>

```

Fig. 11: Configuración de la tarea revisar expediente de proyecto.

Como primer paso se define el nombre único del modelo contenido y el prefijo que se utilizará. Luego se declara el espacio de nombres, se declara el tipo de contenido de la tarea revisar expediente de proyecto y se pone la etiqueta **parent** porque hereda de ese modelo. Se le asigna a la propiedad **name** de la etiqueta **type** el valor del From key asociado a la tarea (rtf: revisarexpproy). Luego se configuran cada una de las propiedades que va a tener la tarea, una de estas es rtf: approveRejectOutcome la cual está asociada a la tarea de aprobar o rechazar. La etiqueta constraint hereda los valores definidos al inicio del modelo, la cual contiene una lista cuyos valores son Aprobar y Rechazar.

❖ Fichero **RevisionesTécnicasFormales.bpmn20.xml**:



En la siguiente figura se muestra un fragmento de la implementación realizada del proceso RTF. Se definen el nombre, el id del proceso. Se configura el flujo de trabajo para que al iniciarse el mismo se le otorguen los permisos correspondientes a cada uno de los grupos definidos.

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://www.omg.org/spec/BPM/20100804/DMODEL" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:as="http://www.w3.org/2001/XMLSchema" xmlns:activiti="http://activiti.org/bpm" xmlns:camdi="http://www.omg.org/spec/BPM/20100804/DI"
xmlns:cmpl="http://www.omg.org/spec/DD/20100524/DC" xmlns:compdi="http://www.omg.org/spec/DD/20100524/DI"
typeLanguage="http://www.w3.org/2001/XMLSchema" expressionLanguage="http://www.w3.org/1999/XPath" targetNamespace="http://www.activiti.org/bpm"/>
<process id="RevisionesTécnicasF" name="Revisiones técnicas formales." isExecutable="true">
  <extensionElements>
    <activiti:executionListener event="start" class="org.alfresco.repo.workflow.activiti.listener.ScriptExecutionListener">
      <activiti:field name="script">
        <activiti:string>&lt;import resource="classpath:alfresco/extension/script/BTF-xml.js"&gt;
var usuariosServicio="GROUP_ASesor_CALENDE";
var usuariosServicio= "GROUP_EXHIBIDO_IDEAS";
var usuariosServicio="GROUP_EXHIBIDO";
var permission = [{"Coordinador",usuariosServicio}];
var permission = [{"Coordinador",usuariosServicio}];
var permission = [{"Coordinador",usuariosServicio}];
setPermissionsNode(permission);
]]&gt;&lt;/activiti:string&gt;
</pre>
</div>
<div data-bbox="383 398 649 413" data-label="Caption">
<p>Fig. 12: Configuración del proceso RTF.</p>
</div>
<div data-bbox="119 438 650 455" data-label="Section-Header">
<h3>❖ Fichero <b>RevisionesTécnicasFormales-generic-context.xml</b>:</h3>
</div>
<div data-bbox="88 463 946 576" data-label="Text">
<p>Luego de configurar todos los modelos de contenido para los flujos de trabajo, se le indica a Alfresco en el fichero RevisionesTécnicasFormales-generic-context.xml la ubicación de los flujos de trabajo y modelos de contenido que debe cargar. Estos son ubicados dentro de etiquetas list, en una la propiedad name tiene valor workflowDefinitions y en la otra models. A continuación se muestra un fragmento donde se especifica el flujo de trabajo.</p>
</div>
<div data-bbox="258 581 766 798" data-label="Code-Block">
<pre>&lt;?xml version="1.0" encoding="UTF-8"?&gt;
&lt;!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd"&gt;
&lt;beans&gt;
  &lt;bean id="org.alfresco.repo.workflow.activiti.deployer" parent="workflowDeployer"&gt;
    &lt;!-- Register workflows --&gt;
    &lt;property name="workflowDefinitions"&gt;
      &lt;list&gt;
        &lt;props&gt;
          &lt;prop key="engineId"&gt;activiti&lt;/prop&gt;
          &lt;prop key="location"&gt;alfresco/workflows/RevisionesTécnicasFormales.bpmn20.xml&lt;/prop&gt;
          &lt;prop key="mimetype"&gt;text/xml&lt;/prop&gt;
          &lt;prop key="redeploy"&gt;true&lt;/prop&gt;
        &lt;/props&gt;
      &lt;/list&gt;
    &lt;/property&gt;
    &lt;!-- Register models --&gt;
    &lt;property name="models"&gt;
      &lt;list&gt;
        &lt;value&gt;alfresco/workflows/RevisionesTécnicasFormales-generic-model.xml&lt;/value&gt;
        &lt;value&gt;alfresco/workflows/RevisionesTécnicasFormales-model.xml&lt;/value&gt;
      &lt;/list&gt;
    &lt;/property&gt;
  &lt;/bean&gt;
</pre>
</div>
<div data-bbox="239 831 793 846" data-label="Caption">
<p>Fig. 13: Configuración del fichero RevisionesTécnicasFormales-generic-context.xml.</p>
</div>
<div data-bbox="119 871 734 888" data-label="Section-Header">
<h3>❖ Fichero <b>RevisionesTécnicasFormales-message-workflow.properties</b>:</h3>
</div>
<div data-bbox="503 965 530 980" data-label="Page-Footer">
<p>46</p>
</div>
```

En este fichero se definirán 2 etiquetas, una para que muestre el nombre del Proceso y la otra para que en el campo de la descripción de la tarea muestre el texto “Descripción del proceso”. A continuación se muestra la definición de estas etiquetas:

```
#Internacionalización del proceso RIF

workflow.set.title.RevisionesTecnicasFormales=Proceso de Revisiones Técnicas Formales

workflow.set.description=Descripci\u00f3n del proceso

workflow.set.title.emitirmri=Emitir minuta de reuni\u00f3n de inicio
workflow.set.title.revisarexproy=Revisar expediente de proyecto
```

Fig. 14: Configuración del fichero RevisionesTécnicasFormales-message-workflow.properties.

### ❖ Fichero **RevisionesTécnicasFormales-form-config.xml**

En la etiqueta field-visibility se definen los campos que serán visibles, entre ellos la lista de los adjuntos (packageItems) y el botón con las acciones para aprobar o rechazar la tarea (rtff: approveRejectOutcome). Luego en la etiqueta appearance se define la apariencia de los campos según se desee, por ejemplo, a la propiedad appearance de la etiqueta set se le asigna valor bordered-panel, para que cada campo aparezca con borde.

```
<alfresco-config>
  <config evaluator="string-compare" condition="activiti$RevisionesTecnicasF">
    <forms>
      <form>
        <field-visibility>
          <show id="bpm:workflowDescription" />
          <show id="packageItems" />
          <show id="transitions" />
        </field-visibility>
        <appearance>
          <set id="" appearance="bordered-panel" label-id="workflow.set.title.emitirmri" />
          <field id="bpm:workflowDescription" label-id="workflow.set.description" />
          <field id="packageItems" label-id="workflow.set.items" mandatory="true"/>
          <field id="transitions" />
        </appearance>
      </form>
    </forms>
  </config>
```

Fig. 15: Configuración del fichero RevisionesTécnicasFormales-form-config.xml.

### ❖ Fichero **RevisionesTécnicasFormales-context.xml**:

Finalmente, se procede a configurar el fichero **RevisionesTécnicasFormales-context.xml**, en el cual se define la ubicación de las configuraciones de las vistas de cada flujo y se especifica la ubicación de los mensajes a mostrar, quedando de la siguiente forma:

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
  <bean id="org.RevisionesTécnicasFormales.process.share" class="org.springframework.extensions.config.ConfigBootstrap"
    init-method="register" depends-on="">
    <property name="configService" ref="web.config" />
    <property name="config">
      <list>
        <value>classpath:alfresco/web-extension/RevisionesTécnicasFormales-ferm-config.xml</value>
      </list>
    </property>
  </bean>
  <bean id="org.RevisionesTécnicasFormales.message.share" class="org.springframework.extensions.srfr.cdi.ResourceBundleBootstrapComponent">
    <property name="resourceBundles">
      <list>
        <value>alfresco.messages.RevisionesTécnicasFormales-message-workflow</value>
      </list>
    </property>
  </bean>
</beans>

```

Fig. 16: Configuración del fichero RevisionesTécnicasFormales-context.xml.

### 3.6 Tarea programada (Scheduled jobs).

Son una serie de trabajos programados que ayudan en el mantenimiento de un entorno de producción (Inc, 2014). En la propuesta de solución se realiza una tarea o trabajo que todos los días a las 6: 00 am ejecute un script. Mediante este script se verifica si existen unas nuevas peticiones de pruebas RTF.

```

<bean id="exscriba_rtf_schednledJobDetail" class="org.springframework.scheduling.quartz.JobDetailBean">
  <property name="jobClass">
    <value>org.alfresco.repo.jscrip.ExecuteScriptJob</value>
  </property>
  <property name="jobDataAsMap">
    <map>
      <entry key="scriptLocation">
        <bean class="org.alfresco.repo.jscrip.ClasspathScriptLocation">
          <constructor-arg>
            <value>alfresco/direccion/javascrip.js</value>
          </constructor-arg>
        </bean>
      </entry>
      <entry key="scriptService">
        <ref bean="ScriptService"/>
      </entry>
      <entry key="authenticationComponent">
        <ref bean="authenticationComponent"/>
      </entry>
    </map>
  </property>
</bean>

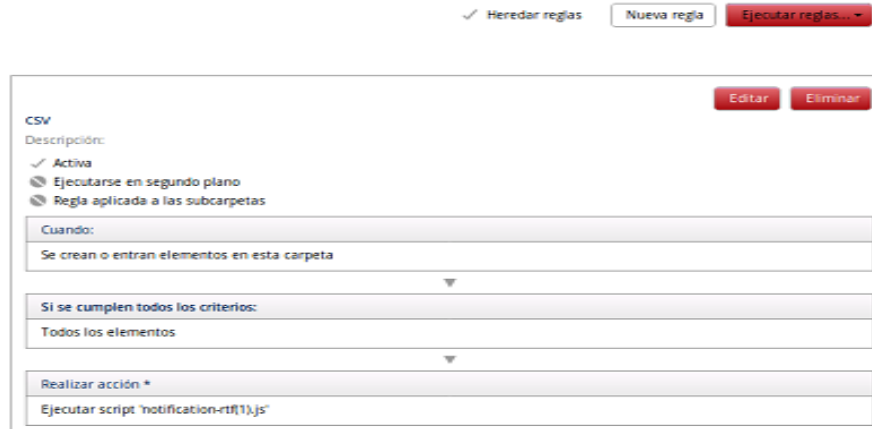
```

Fig. 17: Tarea programada.

### 3.7 Reglas de contenido.

Una regla de contenido permite definir acciones automáticas a partir de ciertos como crear o entrar elementos a una carpeta. Estas reglas son definidas de forma manual en el sistema. En la propuesta de solución la regla de contenido se utiliza para notificar a los usuarios la nueva revisión a realizar, en caso de que no se inicie el flujo de trabajo de forma automática. Esta regla de contenido ejecutará un script, el cual va a contener la implementación para realizar la notificación.

A continuación se muestra la configuración de la regla de contenido csv para el inicio del flujo de trabajo.



The image shows a configuration window for a rule. At the top, there are buttons for 'Heredar reglas' (checked), 'Nueva regla', and 'Ejecutar reglas...'. Below this, there are 'Editar' and 'Eliminar' buttons. The main area is titled 'CSV' and contains the following configuration:

- Descripción:
- Activa
- Ejecutarse en segundo plano
- Regla aplicada a las subcarpetas
- Cuando: Se crean o entran elementos en esta carpeta
- Si se cumplen todos los criterios: Todos los elementos
- Realizar acción \*: Ejecutar script "notification-rtf(1).js"

Fig. 18: Regla de contenido CSV.

### 3.8 Estándares de codificación.

Se definen estándares de codificación porque un estilo de programación homogéneo en un proyecto permite que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea mantenible (Crockford, 2011). A continuación se muestran los estándares de codificación utilizados:

#### Declaración de funciones:

No debe haber espacio entre el nombre de una función y el paréntesis izquierdo de su lista de parámetros. Solo debe haber un espacio entre el paréntesis derecho y el corchete izquierdo que inicia el cuerpo de la instrucción. El corchete derecho se alinea con la línea que contiene el inicio de la declaración de la función. A continuación se muestra un fragmento de código donde se hace uso del estándar declaración de funciones.

```
//Devuelve los usuarios que pertenecen a un grupo
function getUsersbyGroup(grupo) {
    var grupoObtenido = people.getGroup(grupo);
    var lista = people.getMembers(grupoObtenido);
    var collection = new java.util.ArrayList();
    if(lista.length==1){
        return collection.add(lista[0].properties.userName);
    }else{
        collection.add(lista[0].properties.userName);
        for(var i=1; i < lista.length; i++){
            collection.add(lista[i].properties.userName);
        }
    }
    return collection;
}
```

Fig. 19: Estándar de codificación declaración de funciones.

### Comentarios:

Los comentarios se definen comenzando con los caracteres `/*` y terminando con `*/` o `//`. A continuación se muestra un fragmento de código donde se hace uso del estándar comentario.

```
//Devuelve los usuarios que pertenecen a un grupo
function getUsersbyGroup(grupo){
    var grupoObtenido = people.getGroup(grupo);
    var list = people.getMembers(grupoObtenido);
    var collection = new java.util.ArrayList();
    if(list.length==1){
        return collection.add(list[0].properties.userName);
    }else{
        collection.add(list[0].properties.userName);
        for(var i=1; i < list.length; i++){
            collection.add(list[i].properties.userName);
        }
    }
    return collection;
}
```

Fig. 20: Estándar de codificación comentarios.

### Declaración de variables:

Todas las variables deben ser declaradas antes de ser usadas. Las variables son declaradas anteponiendo la instrucción `var` y el nombre de la variable se escribe con minúscula. Estas instrucciones deben ser las primeras en el cuerpo de la función. A continuación se muestra un fragmento de código donde se hace uso del estándar declaración de variables.

```
//Devuelve los usuarios que pertenecen a un grupo
function getUsersbyGroup(grupo){
    var grupoObtenido = people.getGroup(grupo);
    var list = people.getMembers(grupoObtenido);
    var collection = new java.util.ArrayList();
    if(list.length==1){
        return collection.add(list[0].properties.userName);
    }else{
        collection.add(list[0].properties.userName);
        for(var i=1; i < list.length; i++){
            collection.add(list[i].properties.userName);
        }
    }
    return collection;
}
```

Fig. 21: Estándar de codificación declaración de variables.

### Instrucción for:

La clase de instrucciones for debe tener la siguiente forma:

```
for (inicialización; condición; cambio de valor) {
    instrucciones}
```

La primera forma debe ser usada con arrays y con bucles con un número predeterminado de iteraciones. La segunda forma debe ser usada con objetos. A continuación se muestra un fragmento de código donde se hace uso del estándar instrucción for.

```
<userTask name="Emitir minuta de reunión de cierre" id="usertask4" activiti:formKey="rtfemibimenc" activiti:candidateGroups="GROUP_EXPERTO_LIDER">
  <extensionElements>
    - <activiti:taskListener class="org.alfresco.repo.workflow.activiti.tasklistener.ScriptTaskListener" event="create">
      - <activiti:field name="script">
        - <activiti:string>
          - <![CDATA[
            task.setVariable("bpm_comment", execution.getVariable("bpm_comment"));
          ]]>
        </activiti:string>
      </activiti:field>
    </activiti:taskListener>
    - <activiti:taskListener class="org.alfresco.repo.workflow.activiti.tasklistener.ScriptTaskListener" event="complete">
      - <activiti:field name="script">
        - <activiti:string>
          - <![CDATA[
            var space = companyHome.getChildByName("5b05/produccion/documentlibrary(RTF)");
            for(var i=0; i < bpm_package.children.length; i++){
              var estas = bpm_package.children[i].copy(space);
            }
          ]]>
        </activiti:string>
      </activiti:field>
    </activiti:taskListener>
  </extensionElements>
</userTask>
```

Fig. 22: Estándar de codificación instrucción for.

### 3.9 Diagrama de despliegue.

Un diagrama de despliegue consta de la interconexión de nodos a partir de relaciones de asociación. En este contexto, las relaciones representan enlaces físicos (normalmente bidireccionales). 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 (Sibat, 2013). El diagrama de despliegue de la propuesta de solución está compuesto por:

#### PC Cliente:

En este nodo físico es donde estarán alojados los navegadores web para acceder a las páginas clientes que interactúan con las páginas servidoras, donde se encuentra la aplicación GDA Xabal eXcriba 3.1. Esta interacción se realiza mediante el uso del protocolo HTTPs por el puerto 8080.

#### Servidor Base de Datos:

Representa el servidor donde estará el sistema gestor de bases de datos que se comunica con el Servidor Web a través del protocolo TCP/IP por el puerto 5432.

#### Servidor de aplicaciones:

Representa el lugar donde estará alojada la interfaz web del sistema GDA Xabal eXcriba 3.1 y su núcleo el ECM Alfresco, así como la capa de servicios.

A continuación se muestra el diagrama de despliegue.

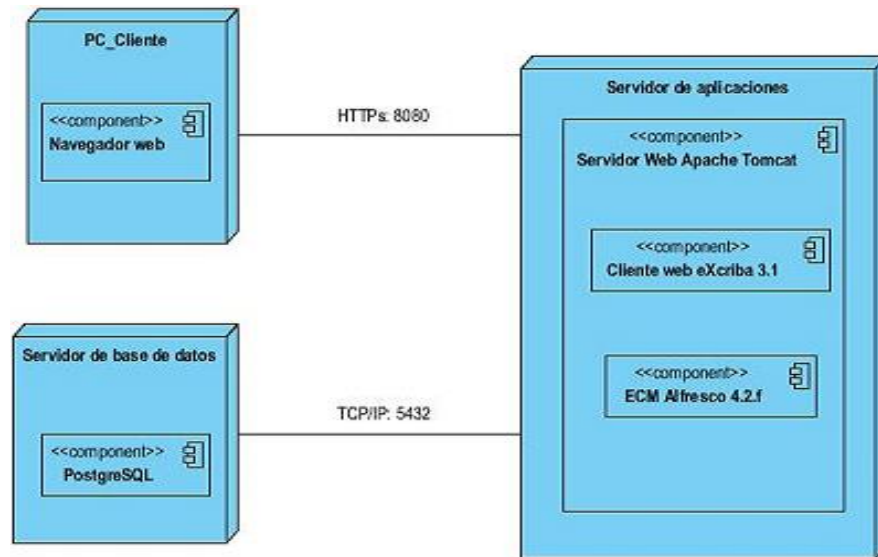


Fig. 23: Diagrama de despliegue.

### 3.10 Pruebas de software.

Las pruebas del software son un elemento fundamental para la garantía de calidad del sistema. Estas pruebas representan una revisión final de las especificaciones del diseño y de la codificación, es decir, las pruebas verifican que el software funcione como se diseñó y que los requerimientos son satisfechos, además de brindar soporte para encontrar y documentar defectos del sistema (Pressman, 2010). A continuación se muestran los casos de pruebas realizados.

La metodología AUP-UCI define 3 pruebas, las cuales son:

**Pruebas internas:** mediante estas pruebas se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones a ser liberadas. Se deben desarrollar artefactos de pruebas como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de pruebas ejecutables para automatizar las pruebas (Sánchez, 2014).

**Pruebas de liberación:** pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación (Sánchez, 2014).

**Pruebas de aceptación:** es la prueba final antes del despliegue del sistema. Su objetivo es verificar que el software está listo y que puede ser usado por usuarios finales para ejecutar aquellas funciones y tareas para las cuales el software fue construido (Sánchez, 2014).

De las pruebas anteriormente mencionadas se realizaron las pruebas internas y las pruebas de aceptación. Las pruebas internas se realizaron mediante la prueba de caja de negra y la prueba de caja blanca.

### Pruebas de caja negra.

Las pruebas de caja negra también denominadas pruebas de comportamiento se centran en los requisitos funcionales del software. El objetivo principal de las pruebas de caja negra es verificar que se cumplan los requisitos funcionales planteados (Pressman, 2010). Estas pruebas intentan encontrar errores en las categorías siguientes:

- ❖ Funciones incorrectas o faltantes.
- ❖ Errores de interfaz.
- ❖ Errores en las estructuras de datos o en el acceso a base de datos externas.
- ❖ Errores de comportamiento o rendimiento.
- ❖ Errores de inicialización y terminación.

Para desarrollar las pruebas de caja negra existen varias técnicas, entre ellas se encuentran (Pressman, 2010):

- ❖ Técnica de la partición de equivalencia: divide el dominio de entrada de un programa en clases de datos de los que pueden derivarse casos de prueba. Además se basa en una evaluación de las clases de equivalencia para una condición de entrada.
- ❖ 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.
- ❖ Prueba basada en grafos: la prueba de software comienza con la creación de un gráfico de objetos importantes y sus relaciones y luego diseña una serie de pruebas que cubrirán el gráfico de modo que cada objeto y relación se revise y se descubran errores.

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 (Pressman, 2010). Por lo anteriormente expuesto se utiliza la técnica de partición de equivalencia para la realización de los casos de prueba.

A continuación, se describe el caso de prueba basado en el primer requisito funcional utilizando la técnica de partición equivalente, donde se exponen el nombre del requisito, la descripción del caso de prueba, las



variables, la respuesta del sistema y el flujo central. Los casos de pruebas restantes se pueden observar en el anexo # 5.

Clases válidas:

- ❖ Se ha iniciado una revisión RTF.
- ❖ Documentos a revisar.

Clase inválida:

- ❖ Campo vacío

Tabla. 8: Caso de prueba # 1.

**SC Iniciar flujo de trabajo RTF.**

Escenario	Descripción	Comentario	Elementos adjuntos	Respuesta del sistema	Flujo central
EC 1.1 Iniciar flujo de trabajo RTF.	EL experto líder inicia el flujo de trabajo RTF.	V	V	El sistema muestra el flujo de trabajo RTF creado.	1- El experto líder (usuario) se autentica en el sistema. 2- Selecciona el panel de tareas. 3- Selecciona la opción flujos de trabajo que he iniciado. 4- Selecciona la opción Iniciar flujo de trabajo. 5- Selecciona el flujo de trabajo Revisiones Técnicas Formales. 6- Llena los campos mostrados. 7- Selecciona el botón Iniciar flujo de trabajo.
		Se ha iniciado una revisión RTF.	I vacío		

**Resultados de la prueba:**

Tabla. 9: Resultados de las pruebas.

No. Iteración	Defectos encontrados	Cantidad de no conformidades	No conformidades resueltas
1	Errores de interfaz y ortográficos	6	6
2	Errores ortográficos	2	2
3	-	-	-

Las pruebas realizadas, como se puede apreciar en la tabla anterior, se llevaron a cabo en tres iteraciones.

- ❖ 1ra iteración: se encontraron 3 errores de interfaz (no significativas), 1 error de validación (significativa) y 2 errores ortográficos (no significativas), las cuales fueron resueltas inmediatamente.

- ❖ 2da iteración: se encontraron 2 errores ortográficos (no significativas), los cuales fueron corregidos.
- ❖ 3ra iteración: no se encontraron no conformidades.

Se puede concluir que las pruebas de funcionalidad a la aplicación fueron realizadas satisfactoriamente.

### Prueba de caja blanca.

Las pruebas de caja blanca, en ocasiones llamada prueba de caja de vidrio. Es una filosofía de diseño de casos de prueba que usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba. Al usar los métodos de prueba caja blanca, puede derivar casos de pruebas que:

- ❖ Garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez.
- ❖ Se revisen todas las decisiones lógicas en sus lados verdadero y falso.
- ❖ Ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas.
- ❖ Revisen estructuras de datos internas para garantizar su validez.

La técnica para obtener casos de prueba de caja blanca a utilizar es la prueba del camino básico. Este método permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez.

Los pasos del diseño de pruebas mediante el camino básico son:

1. Obtener el grafo de flujo, a partir del diseño o del código del módulo.
2. Obtener la complejidad ciclomática del grafo de flujo.
3. Definir el conjunto básico de caminos independientes.
4. Determinar los casos de prueba que permitan la ejecución de cada uno de los caminos anteriores.

La **complejidad ciclomática** es una medida del software que aporta una valoración cuantitativa de la complejidad lógica de un programa.

Formas de cálculo:

- ❖ El número de regiones del grafo es igual a la complejidad ciclomática.
- ❖  $V(G) = A - N + 2$ , donde A es el número de aristas y N es el número de nodos contenidos en el grafo.
- ❖  $V(G) = P + 1$ , donde P es el número de nodos predicados contenidos en el grafo.

Para la realización de la prueba de camino básico se escogió el siguiente método:

```
//Devuelve los usuarios que pertenecen a un grupo
function getUsersbyGroup(grupo) {
    var grupoObtenido = people.getGroup(grupo);
    var lista = people.getMembers(grupoObtenido);
    var collection = new java.util.ArrayList();
    if(lista.length==1){
        return collection.add(lista[0].properties.userName);
    }else{
        collection.add(lista[0].properties.userName);
        for(var i=1; i < lista.length; i++){
            collection.add(lista[i].properties.userName);
        }
    }
    return collection;
}
```

Fig. 24: Método para la prueba de caja blanca.

Paso 1: Obtener el grafo de flujo, a partir del diseño o del código del módulo.

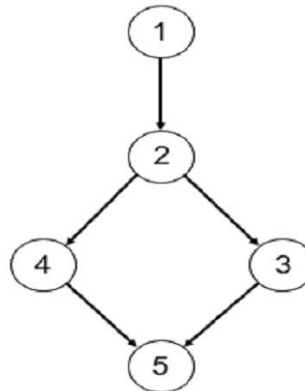


Fig. 25: Grafo de flujo del método.

Paso 2: Obtener la complejidad ciclomática del grafo de flujo.

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

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

$$V(G) = 2$$

Paso 3: Definir el conjunto básico de caminos independientes.

Camino 1: 1, 2, 3, 5

Camino 2: 1, 2, 4, 5

Paso 4: Determinar los casos de prueba que permitan la ejecución de cada uno de los caminos anteriores.

Tabla. 10: Caso de prueba camino básico 1.

<b>Caso de prueba para el camino básico 1</b>	
<b>Descripción:</b> Devolver el único usuario que contiene la lista de miembros.	
<b>Condición de ejecución:</b> Si la lista de miembros contiene un solo usuario (if (lista.length == 1)).	
<b>Datos de entrada</b>	Lista de miembros
<b>Tipos de datos esperados</b>	Lista
<b>Evaluación del caso de prueba:</b> Satisfactoria	

Tabla. 11: Caso de prueba camino básico 2.

<b>Caso de prueba para el camino básico 2</b>	
<b>Descripción:</b> Devolver la lista de los usuarios que están en la lista de miembros.	
<b>Condición de ejecución:</b> Si la lista de miembros contiene o no varios usuarios.	
<b>Datos de entrada</b>	Lista de miembros
<b>Tipo de datos esperados</b>	Lista
<b>Evaluación del caso de prueba:</b> Satisfactoria	

### 3.11 Conclusiones parciales.

- ❖ Las distintas tipologías documentales empleadas permitieron identificar los tipos de contenido que reconocerá la personalización del GDA Xabal eXcriba 3.1.
- ❖ El flujo de trabajo y los ficheros que se desarrollaron posibilitaron la personalización del GDA eXcriba 3.1 para el proceso RTF.
- ❖ La creación de la regla de contenido permitió realizar distintas acciones al conjunto de carpetas que se encuentran en la personalización.
- ❖ Las pruebas realizadas permitieron garantizar la calidad de la propuesta de solución y que los requisitos de la personalización se cumplieran.

## **Conclusiones generales**

A partir de la investigación realizada y los resultados obtenidos, se puede concluir lo siguiente:

- ❖ El estudio realizado sobre los conceptos asociados al dominio del problema facilitó sentar las bases teóricas de la investigación.
- ❖ El estudio de las herramientas estudiadas demostró que no cubren las necesidades de la dirección de calidad para el proceso rtf.
- ❖ La selección de las herramientas y tecnologías posibilitó realizar la personalización del GDA Xabal eXcriba 3.1 a las necesidades de la Dirección de Calidad de Software de la UCI.
- ❖ El modelado del proceso RTF de la Dirección de Calidad de Software de la UCI proporcionó entender su funcionamiento y especificar correctamente el flujo documental para su informatización.
- ❖ La personalización del GDA Xabal eXcriba 3.1 permitió informatizar el flujo de trabajo documental RTF de la Dirección de Calidad de Software de la UCI.

## Referencias bibliográficas

- Potts, Jeff. 2018.** Working With Custom Content Types in Alfresco : s.n., April 2018.
- . **2018.** Introduction to the Web Script Framework. April 2018.
- Software , Inc Alfresco. 2013.** Alfresco Documentation. Scheduled Jobs. [Online] 2013. [Cited: junio 3, 2018.] <https://docs.alfresco.com/5.1/references/dev-extension-points-scheduled-jobs.html>.
- (ISO), International Organization for Standarization. 2015.** Nueva ISO 9001: 2015. [Online] noviembre 24, 2015. [Cited: mayo 15, 2018.] <https://www.nueva-iso-9001-2015.com/2014/11/iso-9001-entendiendo-enfoque-basado-procesos/>.
- Activiti.org. 2017.** Activiti User Guide. mayo 25, 2017.
- Alegsa, Leandro. 2014.** Diccionario de Informática y Tecnología. [Online] 2014. [Cited: enero 8, 2018.] <http://www.alegsa.com.ar/Dic/javascript.php..>
- Alvarado, Yoandy Lazo. 2016.** Proceso de aseguramiento de la calidad para un modelo de la calidad en Cuba. Especial Informática 2016, La Habana : s.n., 2016, Revista Cubana de Ciencias Informáticas, Vol. 10.
- ANDALUCÍA, JUNTA DE. 2013.** Marco de desarrollo de la Junta de Andalucía. [Online] marzo 1, 2013. [Cited: noviembre 10, 2017.] <http://www.juntadeandalucia.es/servicios/madeja/>.
- Antelo Collado, MSc. Aurelio. 2014.** Implementación de procesos modelados con BPMN 2.0 en Activiti Designer. 2014.
- BizAgi. 2014.** Workflow\_Patterns\_using\_BizAgi\_Process\_Modeler. 2014.
- Calleja, Alberto. 2010.** BPMS ACTIVITI. Aplicaciones Webs basadas en Servicios. 2010.
- CORPORATION, I.B.M. 2012..** Iniciándose en la plataforma Eclipse. [Online] 2012. [Cited: enero 11, 2018.] [https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html. .](https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html.)
- Correa, Yenía Silva. 2017.** Buenas prácticas en Universidad de las Ciencias Informáticas. Granma. octubre 17, 2017.
- Crockford, Douglas. 2011.** [Online] julio 30, 2011. [Cited: mayo 17, 2018.] <https://practicasdeprogramacion.wordpress.com/2011/07/30/convenciones-de-codificacion-para-javascript/>.
- Crosby, Philip.B. 1996.** La calidad no cuesta: el arte de cerciorarse de la calidad. s.l. : Compañía Editorial Continental,, 1996.
- Cruz, Germán. 2014.** Gestionando Procesos de Negocio usando BPMN 2.0. [Online] 2014. [Cited: enero 8, 2018.] <https://sg.com.mx/revista/46/gestionando-procesos-negocio-usando-bpmn-20..>
- Delgado Dapena, Martha D and Reloba Varela, Sigrid. 2005.** 2, s.l. : MGRSOFT: UN MODELO PARA GESTIONAR LAS REVISIONES EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS. La Habana, 2005, Red de Revistas Científicas de América Latina, el Caribe, España y Portugal, Vol. XXVI, pp. 53-60.

- Deming, William Edwards. 1989.** Calidad, productividad y competitividad. s.l. : Díaz de Santos.S.A, 1989.
- Escalona, M. J. 2008.** 2, Revisiones de código en el contexto del aseguramiento.Universidad de Sevilla, Madrid, España : s.n., septiembre 2008, Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 4, pp. 46-57.
- Europea de excelencia, Escuela . 2015.** Nuevas normas ISO 9001:2015. [Online] 2015. [Cited: abril 10, 2018.] <https://www.nueva-iso-9001-2015.com/2016/05/como-es-un-mapa-procesos-basado-norma-iso-9001-2015/>.
- Figuroa, María Antonieta Abud. 2000.** Calidad en la Industria del Software. La Norma ISO-9126. 2000.
- Garzás, Javier. 2014.** [Online] agosto 28, 2014. [Cited: marzo 22, 2018.] <http://www.javiergarzas.com/2014/08/tipos-patrones-software.html>.
- Gómez, Dra María del Carmen. 2011.** Notas del curso: Análisis de requerimientos. Universidad Autónoma Metropolitana. Unidad Cuajimalpa. 2011.
- 2013.** Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK). 5ta. 2013.
- Hernández, Lizbeth A. 2012.** Herramienta para la Automatización de Revisiones Técnicas Formales como apoyo al desarrollo Orientado a Objetos. RETO : s.n., 2012.
- IBM Rational RequisitePro. 2010.** Rational RequisitePro. [Online] 2010. [Cited: febrero 28, 2018.] [http://public.dhe.ibm.com/software/dw/ssa/WR366309DM\\_06\\_Rational\\_RequisitePro-Datasheet.pdf](http://public.dhe.ibm.com/software/dw/ssa/WR366309DM_06_Rational_RequisitePro-Datasheet.pdf).
- Inc, Alfresco. 2014.** Alfresco Documentation. [Online] 2014. [Cited: mayo 15, 2018.] <https://docs.alfresco.com/5.0/concepts/scheduled-jobs.html>.
- Informáticas, Universidad de las Ciencias. 2017.** Universidad de las Ciencias Informáticas. [Online] 2017. [Cited: diciembre 3, 2017.] <http://mejoras.prod.uci.cu>.
- Informáticas., Universidad de las Ciencias. 2017.** Universidad de las Ciencias Informáticas. [Online] 2017. [Cited: diciembre 3, 2017.] [www.uci.cu](http://www.uci.cu).
- JACOBSON, , Ivar. 2004.** El Proceso Unificado de Desarrollo de Software. 2004.
- Larman, Craig. 2003.** UML y patrones. 2da edición. 2003.
- Ledesma , Pablo. 2013.** PDCAHOME. [Online] 2013. [Cited: marzo 4, 2018.] <https://www.pdcahome.com/check-list/>.
- López, María Sierra y Patricia. 2017.** Herramienta CASE Visual Paradigm. Univ. Cantabria – Fac. de Ciencias : s.n., 2017.
- Marín, Díaz, Aymara. 2015.** Manual de Funcionamiento Interno. Dirección de Calidad. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015.
- Martínez, Prof. Ivette Carolina. 2010.** Clase 6: Modelo conceptual/Modelo de dominio. Universidad Simón Bolívar. Ingeniería de software. 2010.
- Mestras, Juan Pavón. 2013.** JavaScript. Aplicaciones Web/Sistemas Web. 2013.

**Microsoft. 2018.** Visual Studio Code. 2018.

**Moses, Juran, Joseph. 1993.** Juran y el liderazgo para la calidad. Un manual para ejecutivos. s.l. : Díaz de Santos, S.A., 1993.

**Pérez, Javier Eguíluz. 2015.** Introducción a JavaScript. 2015.

**Potts, Jeff. 2012.** Alfresco Developer Series. Advanced Workflows. 2012.

**Pressman, Roger S. 2010.** Ingeniería del software. Un enfoque práctico. [ed.] Pablo Roig Vázquez. [trans.] Victor Campos Olgún. 10ma. 2010.

Proceso de aseguramiento de la calidad para un modelo de la calidad en Cuba. **Lazo Alvarado, Yoandy and otros, y. 2016.** Especial Informática 2016, La Habana : Ediciones Futuro, mayo 5, 2016, Vol. 10, pp. 124-137.

**Project Management Institute, Inc. 2013.** Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK). Quinta. 2013.

**RAE. 2017.** Diccionario de la lengua española. [Online] 2017. [Cited: abril 11, 2018.] <http://dle.rae.es/?id=ZpHCefb>.

—. **2014.** Diccionario de la Lengua Española. [Online] 2014. [Cited: marzo 4, 2018.] <http://dle.rae.es/?id=SjfQTpg>.

**Revelles, Jesús Salinas. 2014.** Alfresco 4.2 para desarrolladores. El núcleo de alfresco. 2014.

—. **2014.** Alfresco ECM 4.2. Tipos de contenidos personalizados. 2014.

**Sánchez, Tamara Rodríguez. 2014.** Metodología de desarrollo para la Actividad productiva UCI. La Habana : s.n., 2014.

**Scientec., Software científico y técnico. 2016.** Software científico y técnico Scientec. [Online] octubre 17, 2016. [Cited: junio 12, 2018.] <https://www.scientec.com.mx/axure-rp/>.

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

**Sibat, Martinez Yanet. 2013.** Solución para la ejecución de flujos de trabajo de aprobación. 2013.

**Sommerville, Ian. 2011.** SOFTWARE ENGINEERING 9. Ninth Edition. 2011.

**Valencia, Universidad de.** Universidad de Valencia. [Online] [Cited: marzo 21, 2018.] <https://www.uv.es/uvweb/unidad-calidad/es/calidad-servicios/evaluacion-servicios/mapa-procesos-1285885175420.html>.

**Villate., Jaime E. 2001.** Introducción al XML. Universidad de Oporto. Madrid, España : s.n., 2001.

**Vonka, Jan . 2014.** Alfresco Core Repository. october 2014.



## **Bibliografía**

- Potts, Jeff. 2018.** Working With Custom Content Types in Alfresco : s.n., April 2018.
- . 2018. Introduction to the Web Script Framework. April 2018.
- Software , Inc Alfresco. 2013.** Alfresco Documentation. Scheduled Jobs. [Online] 2013. [Cited: junio 3, 2018.] <https://docs.alfresco.com/5.1/references/dev-extension-points-scheduled-jobs.html>.
- (ISO), International Organization for Standarization. 2015.** Nueva ISO 9001: 2015. [Online] noviembre 24, 2015. [Cited: mayo 15, 2018.] <https://www.nueva-iso-9001-2015.com/2014/11/iso-9001-entendiendo-enfoque-basado-procesos/>.
- Activiti.org. 2017.** Activiti User Guide. mayo 25, 2017.
- Alegsa, Leandro. 2014.** Diccionario de Informática y Tecnología. [Online] 2014. [Cited: enero 8, 2018.] <http://www.alegsa.com.ar/Dic/javascript.php..>
- Alvarado, Yoandy Lazo. 2016.** Proceso de aseguramiento de la calidad para un modelo de la calidad en Cuba. Especial Informática 2016, La Habana : s.n., 2016, Revista Cubana de Ciencias Informáticas, Vol. 10.
- ANDALUCÍA, JUNTA DE. 2013.** Marco de desarrollo de la Junta de Andalucía. [Online] marzo 1, 2013. [Cited: noviembre 10, 2017.] <http://www.juntadeandalucia.es/servicios/madeja/>.
- Antelo Collado, MSc. Aurelio. 2014.** Implementación de procesos modelados con BPMN 2.0 en Activiti Designer. 2014.
- BizAgi. 2014.** Workflow\_Patterns\_using\_BizAgi\_Process\_Modeler. 2014.
- Calleja, Alberto. 2010.** BPMS ACTIVITI. Aplicaciones Webs basadas en Servicios. 2010.
- CORPORATION, I.B.M. 2012..** Iniciándose en la plataforma Eclipse. [Online] 2012. [Cited: enero 11, 2018.] <https://www.ibm.com/developerworks/ssa/library/os-ecov/index.html.>
- Correa, Yenía Silva. 2017.** Buenas prácticas en Universidad de las Ciencias Informáticas. Granma. octubre 17, 2017.
- Crockford, Douglas. 2011.** [Online] julio 30, 2011. [Cited: mayo 17, 2018.] <https://practicasdeprogramacion.wordpress.com/2011/07/30/convenciones-de-codificacion-para-javascript/>.
- Crosby, Philip.B. 1996.** La calidad no cuesta: el arte de cerciorarse de la calidad. s.l. : Compañía Editorial Continental,, 1996.
- Cruz, Germán. 2014.** Gestionando Procesos de Negocio usando BPMN 2.0. [Online] 2014. [Cited: enero 8, 2018.] <https://sg.com.mx/revista/46/gestionando-procesos-negocio-usando-bpmn-20..>
- Delgado Dapena, Martha D and Reloba Varela, Sigrid. 2005.** 2, s.l. : MGRSOFT: UN MODELO PARA GESTIONAR LAS REVISIONES EN LAS PEQUEÑAS Y MEDIANAS EMPRESAS. La Habana, 2005, Red de Revistas Científicas de América Latina, el Caribe, España y Portugal, Vol. XXVI, pp. 53-60.

- Deming, William Edwards. 1989.** Calidad, productividad y competitividad. s.l. : Díaz de Santos.S.A, 1989.
- Escalona, M. J. 2008.** 2, Revisiones de código en el contexto del aseguramiento.Universidad de Sevilla, Madrid, España : s.n., septiembre 2008, Revista Española de Innovación, Calidad e Ingeniería del Software, Vol. 4, pp. 46-57.
- Europea de excelencia, Escuela . 2015.** Nuevas normas ISO 9001:2015. [Online] 2015. [Cited: abril 10, 2018.] <https://www.nueva-iso-9001-2015.com/2016/05/como-es-un-mapa-procesos-basado-norma-iso-9001-2015/>.
- Figuroa, María Antonieta Abud. 2000.** Calidad en la Industria del Software. La Norma ISO-9126. 2000.
- Garzás, Javier. 2014.** [Online] agosto 28, 2014. [Cited: marzo 22, 2018.] <http://www.javiergarzas.com/2014/08/tipos-patrones-software.html>.
- Gómez, Dra María del Carmen. 2011.** Notas del curso: Análisis de requerimientos. Universidad Autónoma Metropolitana. Unidad Cuajimalpa. 2011.
- 2013.** Guía de los fundamentos para la dirección de proyectos (Guía del PMBOK). 5ta. 2013.
- Hernández, Lizbeth A. 2012.** Herramienta para la Automatización de Revisiones Técnicas Formales como apoyo al desarrollo Orientado a Objetos. RETO : s.n., 2012.
- IBM Rational RequisitePro. 2010.** Rational RequisitePro. [Online] 2010. [Cited: febrero 28, 2018.] [http://public.dhe.ibm.com/software/dw/ssa/WR366309DM\\_06\\_Rational\\_RequisitePro-Datasheet.pdf](http://public.dhe.ibm.com/software/dw/ssa/WR366309DM_06_Rational_RequisitePro-Datasheet.pdf).
- Inc, Alfresco. 2014.** Alfresco Documentation. [Online] 2014. [Cited: mayo 15, 2018.] <https://docs.alfresco.com/5.0/concepts/scheduled-jobs.html>.
- Informáticas, Universidad de las Ciencias. 2017.** Universidad de las Ciencias Informáticas. [Online] 2017. [Cited: diciembre 3, 2017.] <http://mejoras.prod.uci.cu>.
- Informáticas., Universidad de las Ciencias. 2017.** Universidad de las Ciencias Informáticas. [Online] 2017. [Cited: diciembre 3, 2017.] [www.uci.cu](http://www.uci.cu).
- JACOBSON, , Ivar. 2004.** El Proceso Unificado de Desarrollo de Software. 2004.
- Larman, Craig. 2003.** UML y patrones. 2da edición. 2003.
- Ledesma , Pablo. 2013.** PDCAHOME. [Online] 2013. [Cited: marzo 4, 2018.] <https://www.pdcahome.com/check-list/>.
- López, María Sierra y Patricia. 2017.** Herramienta CASE Visual Paradigm. Univ. Cantabria – Fac. de Ciencias : s.n., 2017.
- Marín, Díaz, Aymara. 2015.** Manual de Funcionamiento Interno. Dirección de Calidad. Universidad de las Ciencias Informáticas. La Habana : s.n., 2015.
- Martínez, Prof. Ivette Carolina. 2010.** Clase 6: Modelo conceptual/Modelo de dominio. Universidad Simón Bolívar. Ingeniería de software. 2010.
- Mestras, Juan Pavón. 2013.** JavaScript. Aplicaciones Web/Sistemas Web. 2013.

- Microsoft. 2018.** Visual Studio Code. 2018.
- Moses, Juran, Joseph. 1993.** Juran y el liderazgo para la calidad. Un manual para ejecutivos. s.l. : Díaz de Santos, S.A., 1993.
- Pérez, Javier Eguíluz. 2015.** Introducción a JavaScript. 2015.
- Potts, Jeff. 2012.** Alfresco Developer Series. Advanced Workflows. 2012.
- Pressman, Roger S. 2010.** Ingeniería del software. Un enfoque práctico. [ed.] Pablo Roig Vázquez. [trans.] Victor Campos Olgún. 10ma. 2010.
- Project Management Institute, Inc. 2013.** Guía de los Fundamentos para la Dirección de Proyectos (Guía del PMBOK). Quinta. 2013.
- RAE. 2017.** Diccionario de la lengua española. [Online] 2017. [Cited: abril 11, 2018.] <http://dle.rae.es/?id=ZpHCefb>.
- . **2014.** Diccionario de la Lengua Española. [Online] 2014. [Cited: marzo 4, 2018.] <http://dle.rae.es/?id=SjfQTpg>.
- Revelles, Jesús Salinas. 2014.** Alfresco 4.2 para desarrolladores. El núcleo de alfresco. 2014.
- . **2014.** Alfresco ECM 4.2. Tipos de contenidos personalizados. 2014.
- Sánchez, Tamara Rodríguez. 2014.** Metodología de desarrollo para la Actividad productiva UCI. La Habana : s.n., 2014.
- Scientec., Software científico y técnico. 2016.** Software científico y técnico Scientec. [Online] octubre 17, 2016. [Cited: junio 12, 2018.] <https://www.scientec.com.mx/axure-rp/>.
- Shariff, Munwar. 2013.** Alfresco 4 Enterprise Content Management Implementation. 2013.
- Sibat, Martinez Yanet. 2013.** Solución para la ejecución de flujos de trabajo de aprobación. 2013.
- Sommerville, Ian. 2011.** SOFTWARE ENGINEERING 9. Ninth Edition. 2011.
- Valencia, Universidad de.** Universidad de Valencia. [Online] [Cited: marzo 21, 2018.] <https://www.uv.es/uvweb/unidad-calidad/es/calidad-servicios/evaluacion-servicios/mapa-procesos-1285885175420.html>.
- Villate., Jaime E. 2001.** Introducción al XML. Universidad de Oporto. Madrid, España : s.n., 2001.
- Vonka, Jan . 2014.** Alfresco Core Repository. october 2014.

## **Anexos**

### **Anexo 1: Entrevista.**

1. Nombre(s) y Apellidos del entrevistado. Ismaila López Sotolongo profesor.
2. ¿Cómo se realiza el proceso Revisiones Técnicas Formales actualmente en la universidad?(paso a paso)
3. ¿Qué actores intervienen en cada paso del proceso RTF?
4. ¿Cuáles son los problemas fundamentales que impiden el buen funcionamiento del proceso?
5. ¿Qué documentos se generan en el transcurso del proceso?
6. ¿Todas las personas tienen acceso a estos documentos?
7. ¿Quiénes tienen permisos para acceder a estos documentos?
8. ¿Cuáles son los permisos definidos?
9. ¿Cuáles son los roles definidos?
10. ¿Cómo reciben y envían la documentación?
11. ¿Cuál es el expediente de proyecto que actualmente se utiliza en los proyectos?
12. ¿Cuáles son las listas de chequeo que se realizan en el proceso RTF?
13. ¿Qué documentos del expediente de proyecto se tienen en cuenta para la realización de las listas de chequeo?

### **Anexo # 6: Acta de aceptación.**



## 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: **Personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software**, se hace entrega del producto que se relaciona a continuación:

- Personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software.

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

Entrega Tesista	Recibe Centro CIGED
<b>Nombre y apellidos:</b> Anamelys Lugo Zulueta	<b>Nombre y apellidos:</b> <i>Azedie Astelo Collazo</i>
<b>Cargo:</b> <i>Desarrollador</i>	<b>Cargo:</b> <i>Director</i>

Fecha: 13/06/2018





## 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: **Personalización del Gestor de Documentos Administrativos Xabal eXcriba 3.1 para el proceso Revisiones Técnicas Formales de la Dirección de Calidad de Software**, se hace entrega del producto que se relaciona a continuación:

- Especificación de requisitos del software.

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

Entrega Tesista	Recibe Centro CIGED
Nombre y apellidos: Anamelys Lugo Zulueta	Nombre y apellidos: <i>Aurelio Arturo Rojas</i>
Cargo: <i>Desarrollador</i>	Cargo: <i>Director</i>

Fecha: 13/06/2018

