

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UNA HERRAMIENTA CASE PARA LA GESTIÓN DEL ALCANCE DE PROYECTOS BASADA EN WBS

Tesis para optar por el Título de Ingeniero Informático, que presenta el bachiller:

Germán Rodolfo Arellano Guzmán

ASESOR: Ingeniero Renzo Jeremías Quintanilla

Lima, mayo del 2014

Resumen

El alcance de un proyecto se define como todo el trabajo necesario para crear el resultado final del mismo. Una inadecuada definición del alcance trae consigo errores en la planificación de recursos, tiempos y costos del proyecto. Estos errores generan retrasos, sobrecostos, productos con características incompletas e inclusive puede ocasionar la cancelación total del proyecto antes de ser completado.

La Estructura de Descomposición del Trabajo o WBS (por sus siglas en inglés: *Work Breakdown Structure*) es una de las técnicas fundamentales para la planificación y control del alcance de un proyecto. Sin embargo, el poco difundido y limitado acceso a herramientas de gestión que apliquen técnicas como WBS, ocasiona que se dejen de lado estas técnicas o que no sean utilizadas correctamente. El resultado de no aplicar técnicas estándares de gestión del alcance de proyectos (en este caso WBS) conlleva a un probable fracaso del proyecto.

El presente proyecto de fin de carrera busca analizar, diseñar y construir una herramienta CASE permita gestionar integralmente el alcance de un proyecto, partiendo de la aplicación de la técnica WBS.







A mis padres, por su apoyo incondicional y sacrificio.

ÍNDICE GENERAL

1.	Generalidades	10
1.1.	Definición del Problema	10
1.2.	Marco conceptual del problema	12
1.2.1.	Proyecto.....	12
1.2.2.	Alcance de un proyecto	13
1.2.3.	Alcance de un producto	13
1.2.4.	WBS.....	13
1.2.4.1.	Definición.....	13
1.2.4.2.	Historia	14
1.2.4.3.	Diseño de un WBS	14
1.3.	Plan del proyecto	16
1.3.1.	Metodología para la gestión del proyecto	16
1.4.	Estado del arte	22
1.5.	Descripción y sustentación de la solución	24
2.	Análisis	28
2.1.	Definición de la metodología de la solución.....	28
2.2.	Identificación de requisitos.....	31
2.3.	Identificación de casos de uso	33
2.4.	Análisis de la Solución	37
2.4.1.	Viabilidad de la herramienta	37
2.4.2.	Definición de la herramienta	37
2.4.2.1.	Diagramas de Clases de Análisis.....	38
3.	Diseño	42
3.1.	Arquitectura de la solución.....	42
3.1.1.	Definición de la arquitectura	42
3.1.2.	Diagrama de componentes.....	43
3.1.3.	Beneficios de la arquitectura.....	47
3.2.	Diseño de Interfaz Gráfica	48
4.	Construcción.....	56
4.1.	Construcción	56
4.1.1.	Lenguajes	56
4.1.2.	Entorno de programación	57
4.1.3.	Tecnologías, frameworks y patrones utilizados	58
4.1.4.	Tópicos relevantes de la construcción.....	62

4.2. Pruebas.....	64
4.2.1. Estrategia de pruebas.....	65
4.2.2. Tipos de prueba.....	66
4.2.3. Catálogo de pruebas.....	66
4.2.4. Reporte de pruebas.....	70
5. Observaciones, conclusiones y recomendaciones.....	72
5.1. Observaciones.....	72
5.2. Conclusiones.....	73
5.3. Recomendaciones y trabajos futuros.....	73
Bibliografía.....	75



ANEXOS

ANEXO A: CATÁLOGO DE REQUISITOS

ANEXO B: ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ANEXO C: DICCIONARIO DE CLASES DE ANÁLISIS

ANEXO D: CATÁLOGO DE PRUEBAS

ANEXO E: DOCUMENTO DE VISIÓN



ÍNDICE DE FIGURAS

1.1: Modelo de esquema de un WBS	15
1.2: Modelo de árbol de un WBS	16
1.3: Modelo gráfico del WBS del proyecto	20
1.4: Principales hitos del cronograma del proyecto	21
2.1: Diagrama de casos de uso para el Paquete de creación de WBS	36
2.2: Diagrama de Clases de un proyecto tipo elaborado en la herramienta	38
2.3: Diagrama de Clases del Metamodelo para un modelo WBS	40
3.1: Diagrama de paquetes de la herramienta	44
3.2: Diagrama de componentes de la herramienta	45
3.3: Pantalla principal de la interfaz de usuario	49
3.4: Pantalla de ingreso de información	50
3.5: Árbol de navegación de versiones del proyecto	51
3.6: Árbol de navegación de nodos de un modelo WBS	52
3.7: Modelo gráfico de un WBS	52
3.8: Pantalla de configuración de atributos de nodo	54
3.9: Pestaña de atributos de nodo	54
3.10: Pantalla de generación de documentación	55
4.1: Pseudocódigo para el algoritmo de copia de un modelo WBS	64
4.2: Ejemplo de reporte de pruebas	71

ÍNDICE DE TABLAS

1.1: Procesos para la dirección del proyecto	19
1.2 Resumen de las características de los productos y alternativas para crear un WBS	26
1.3: Problemática versus funcionalidades que la resuelven.	27
2.1: Resumen de entregables de la metodología de construcción del producto	31
4.1: Resumen de tecnologías utilizadas por cada capa de la arquitectura	62

INTRODUCCIÓN

La aplicación de herramientas y técnicas estándares en la gestión de proyectos se ha convertido en un factor importante para culminar la ejecución de un proyecto dentro de los límites de tiempo, costo y calidad planificados.

El alcance de un proyecto define todo el trabajo necesario para cumplir con los objetivos del mismo. Una correcta gestión del alcance es el punto de partida para la planificación del proyecto, ya que permite estimar los recursos, tiempos y costos asociados a su ejecución. Técnicas como la Estructura de Descomposición del Trabajo (o WBS por sus siglas en inglés: *Work Breakdown Structure*) permiten identificar todo el trabajo requerido para la ejecución de un proyecto, para luego descomponerlo en unidades más pequeñas a las que se les puede asignar recursos, tiempos y costos. Sin embargo, la limitada cantidad y poca difusión de herramientas que permiten aplicar técnicas como WBS (en términos de usabilidad, facilidad de uso y adopción de estándares) ocasiona que estas técnicas sean dejadas de lado por los gerentes de proyecto, configurando un escenario inadecuado para la gestión del alcance que muchas veces resulta en retrasos, sobrecostos o en la cancelación del proyecto.

Por tanto, todo esfuerzo que vaya en la dirección de brindar una herramienta usable que se base en técnicas estándar como WBS para la gestión integral del alcance de un proyecto es valioso.

El presente documento está organizado en cinco capítulos. En un primer capítulo se explica la problemática que se quiere resolver, se expone el marco teórico y se describe y sustenta la solución a presentar. En un segundo capítulo se detalla el análisis de la solución propuesta, explicando tanto la metodología de desarrollo utilizada como el análisis técnico y económico de la solución. El tercer capítulo muestra el diseño de la solución, a nivel de arquitectura de software y de diseño de interfaz gráfica de usuario. Un cuarto capítulo se enfoca en la construcción en sí de la solución a implementar, sustentando las tecnologías utilizadas y describiendo la estrategia de pruebas aplicada. Finalmente, en un quinto capítulo se incluyen observaciones, conclusiones y recomendaciones del proyecto realizado.

1. Generalidades

En este capítulo se presenta los conceptos necesarios para entender el problema que se desea resolver a través del desarrollo del presente proyecto. Asimismo se muestra la metodología seguida para la gestión del proyecto, y finalmente se presenta las alternativas de solución existentes en la actualidad.

1.1. Definición del Problema

La definición del alcance total de un proyecto (alcance de producto y alcance de proyecto) es un factor clave para el éxito en la planificación, ejecución y control del mismo. El alcance de un proyecto determina todo el trabajo a ser realizado para su planificación, ejecución y control. Una inadecuada gestión del alcance trae dificultades en la definición de actividades, tiempos, costos y recursos para completar el trabajo del proyecto, lo que a su vez tiene impacto en fases subsiguientes.

De acuerdo a la Guía de los Fundamentos de la Dirección de Proyectos [1] (o PMBOK por sus siglas en inglés: *Project Management Body of Knowledge*), una de las técnicas fundamentales para la definición y organización del alcance total de un

proyecto es la Estructura de Descomposición del Trabajo [2] (conocida también por sus siglas en inglés como WBS: *Work Breakdown Structure*).

La encuesta *Chaos Report* del Standish Group realizada en el año 2009 muestra que sólo el 32% de proyectos en el área de tecnología son entregados a tiempo, dentro del presupuesto y con todas las características requeridas. Un 44% de proyectos se concluyó fuera de tiempo, por encima del presupuesto y con funcionalidades o características incompletas. El 24% restante son proyectos que fracasaron en su totalidad, al haber sido cancelados antes de concluirse o entregados pero nunca utilizados. Requisitos incompletos, expectativas poco realistas y requisitos cambiantes son algunos de los principales factores que hacen fracasar a un proyecto y están directamente relacionados a una pobre gestión del alcance [24].

La complejidad para aplicar técnicas estándar para la gestión de proyectos (en este caso WBS) debido al limitado y poco difundido acceso a herramientas que permitan esto, ocasiona que no se apliquen adecuadamente estas técnicas o que no se exploten al máximo las posibilidades que brindan. Por ejemplo, no se encuentra utilidad a un WBS si sólo se ve como un diagrama de bloques. Por el contrario, si se contara con herramientas adecuadas para aplicar WBS, se sacaría provecho a las cualidades que tiene para verificar, monitorear y controlar el alcance y para servir como base para la elaboración del cronograma del proyecto [1].

En todo tipo de proyecto, se debe definir a detalle los entregables y el trabajo total que se tiene que concluir para cerrar con éxito el mismo. Si bien el gerente de proyectos es el responsable de definir, acotar y estructurar el alcance, requiere apoyarse de los recursos de mando medio o inferior ya que estos conocen detalle especializado del trabajo que se tiene que realizar en un proyecto y, por lo tanto, pueden aportar en validar si la definición del alcance está correcta y/o completa. De forma similar, el patrocinador del proyecto y el cliente también intervienen en validar que el alcance de un proyecto sea el adecuado. Un modelo WBS (ya sea gráfico o en esquema) permite mostrar de forma clara y ordenada todo el trabajo a realizar en un proyecto. De esta manera se puede validar el alcance con los diferentes involucrados que intervienen en el proyecto. La falta de aplicación de técnicas como WBS en la gestión del alcance de un proyecto no permite un correcto dimensionamiento y seguimiento del trabajo total requerido [25].

Todos estos factores contribuyen a que no se lleve a cabo una adecuada gestión del alcance de un proyecto, lo cual conlleva a un probable fracaso del proyecto (retrasos, sobrecostos, cancelación del proyecto, etc.).

1.2. Marco conceptual del problema

1.2.1. Proyecto

Un proyecto es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único, sujeto a determinadas restricciones de costo y tiempo [1].

Temporal significa que un proyecto tiene un inicio y un final definido. El final se alcanza cuando se cumplen todos los objetivos del proyecto o cuando queda claro que los objetivos no podrán ser alcanzados. También se llega al final de un proyecto cuando la necesidad del proyecto ya no existe o el proyecto se ha cancelado. Si bien se cumple que los proyectos tienen un carácter temporal, esto no aplica a los productos, resultados o servicios creados por un proyecto [1].

Un proyecto crea productos entregables únicos. Los productos entregables de un proyecto pueden ser productos, servicios o resultados. Que estos productos entregables sean únicos quiere decir que el resultado de un proyecto nunca va a ser el mismo que el de otro proyecto. La presencia de elementos repetitivos entre proyectos no cambia la condición de único a los resultados de un proyecto [1].

Los trabajos que se realizan en una organización con la finalidad de lograr un conjunto de objetivos pueden ser clasificados en proyectos y operaciones. Los proyectos y las operaciones tienen ciertas características comunes. Son realizados por personas, tienen restricciones de tiempo y costo y deben ser planificados, ejecutados y controlados. Sin embargo, su principal diferencia radica en que las operaciones son repetitivas y continuas mientras que los proyectos son temporales y únicos. La finalidad de un proyecto es alcanzar sus objetivos y concluir. Por el contrario, el objetivo de una operación continua es dar respaldo al negocio en el día a día [1].

1.2.2. Alcance de un proyecto

El alcance de un proyecto es la definición del trabajo que se va a realizar en el proyecto para elaborar y entregar un producto con determinadas funciones y características especificadas. [1]

Entre las técnicas para la gestión del alcance de un proyecto se encuentran las siguientes [1]:

- Juicio de Expertos, utilizado para analizar la información disponible y conseguir aportes en base a su experiencia para aplicarlos a cualquier detalle técnico y de gestión.
- Descomposición, subdivisión del trabajo que comprenderá el proyecto en componentes cada vez más pequeños y manejables, hasta llegar al nivel de paquetes de trabajo.
- Identificación de Alternativas, para generar diferentes enfoques para la ejecución del trabajo del proyecto. Puede incluir algunas técnicas de gestión como comparación entre pares, tormenta de ideas, etc.
- Inspección, verificación de que el trabajo y/o entregables cumplen los criterios de aceptación del producto

1.2.3. Alcance de un producto

Las características y funciones que caracterizan a un producto, servicio o resultado. [1]

1.2.4. WBS

1.2.4.1. Definición

El WBS es una descomposición jerárquica, orientada al producto entregable, del trabajo que será ejecutado por el equipo del proyecto para lograr los objetivos del proyecto y crear los productos entregables requeridos. El WBS organiza y define el alcance total del proyecto y subdivide el trabajo del proyecto en porciones de trabajo más pequeñas y fáciles de manejar, donde cada nivel descendente representa una definición cada vez más detallada del trabajo del proyecto. [2]

1.2.4.2. Historia

WBS es una de las técnicas modernas de planificación de proyectos. Fue inicialmente desarrollada por el Departamento de Defensa de los Estados Unidos y está descrita en el *Military Standard (MIL-STD) 881B (25 Mar 93)* de la siguiente forma: “Una estructura desglosada de trabajo es un árbol jerárquico orientado a productos compuesto por hardware, software, servicios, datos y facilidades” [3]. En el año 1998, el *Project Management Institute (PMI)* inició un proyecto de elaboración de estándares para la creación de WBS, publicando en el año 2001 el *Practice Standard for Work Breakdown Structure* [2]. En esta publicación se describen los conceptos relacionados a WBS y se recomiendan las acciones a seguir para elaborarlo correctamente. En el 2006 se publicó la segunda edición de este documento, presentando una extensión de estos estándares. En el PMBOK, se menciona al WBS como “uno de los principales entregables para la planificación del alcance de un proyecto”, incluyendo a la creación de un WBS como “uno de los procesos de la gestión del alcance de un proyecto” [1].

1.2.4.3. Diseño de un WBS

Para obtener un buen diseño de WBS se debe tener en cuenta las siguientes recomendaciones:

- Cada elemento del WBS debe representar un único entregable, sólo debe aparecer en una posición de la estructura [4].
- Un elemento del WBS debe representar la agregación de todos sus elementos hijos, posicionados debajo de él (regla del 100%) [4].
- Cada elemento hijo debe pertenecer a un solo elemento (llamado padre) [4].
- Los entregables deben descomponerse de tal manera que se alcance un nivel en el que se muestre de qué manera serán producidos (diseñados, comprados, subcontratados, fabricados) [4].
- Los niveles inferiores de un WBS son llamados *work packages* (o paquetes de trabajo) y se caracterizan por ser unidades de trabajo a las que se les puede asignar recursos, así como estimar su costo y duración [2].
- Los entregables deben identificarse de forma clara para evitar duplicar el trabajo a realizarse [4].
- Todos los entregables definidos en el alcance de un proyecto deben encontrarse en el WBS [4].

- Cada uno de los elementos de un WBS debe estar documentado para asegurar un entendimiento completo del trabajo a realizarse. A esta documentación se le llama Diccionario de WBS [2].
- Se debe usar un código jerárquico que muestre claramente la agregación de cada uno de los elementos del WBS. Por ejemplo, un elemento del nivel más alto puede tener un código como 1, 2, ó 3 mientras que los elementos que deriven de éste pueden tener códigos como 1.1, 2.2, etc. [4]
- Para el diseño de un WBS se pueden usar dos formatos: Esquema y Árbol Jerárquico [2]. Los dos modelos se presentan en la Figura 1.1 y en la Figura 1.2.
- En el WBS no se muestran dependencias ni duraciones [2].

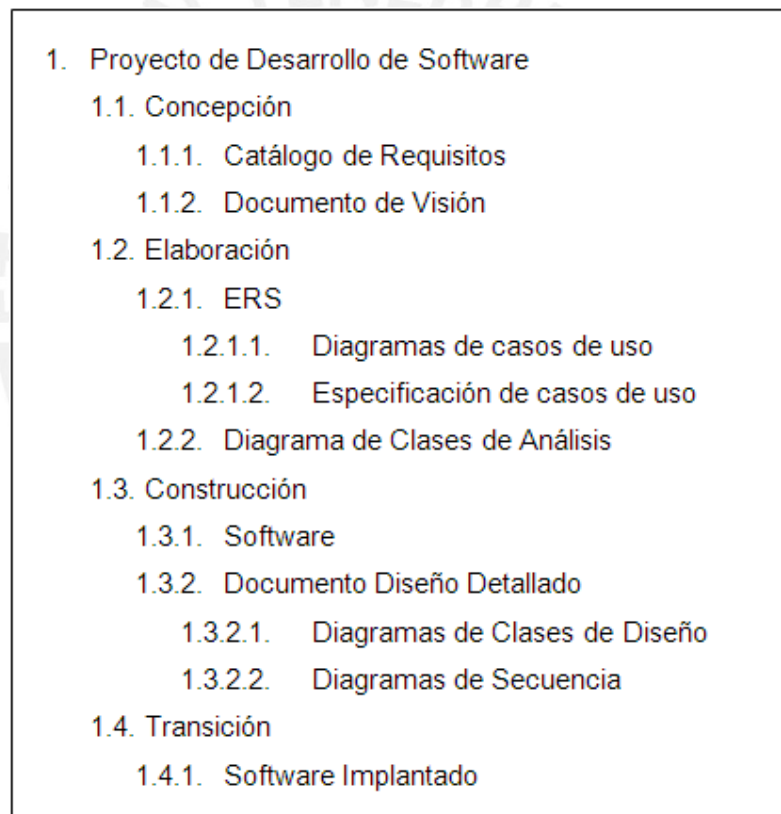


Figura 1.1: Modelo de esquema de un WBS [27]

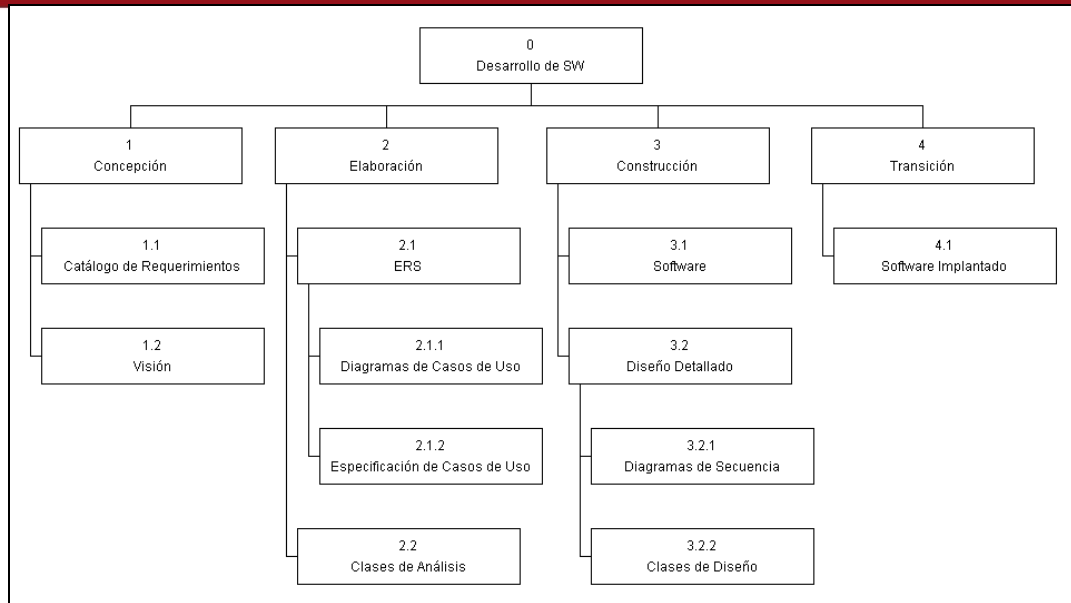


Figura 1.2: Modelo de árbol de un WBS [27]

1.3. Plan del proyecto

En esta sección se describe la metodología utilizada para la gestión del proyecto. Posteriormente se presentan los principales entregables del proyecto en un modelo WBS y los principales hitos del cronograma del proyecto en un diagrama de Gantt.

1.3.1. Metodología para la gestión del proyecto

Para la gestión del proyecto se utilizaron los procesos de gestión del alcance y del tiempo [1] propuestos en el PMBOK (*Project Management Body of Knowledge*) del *Project Management Institute*.

La gestión del alcance es crítica en un proyecto de fin de carrera ya que se tiene que definir cuidadosamente el alcance del proyecto para asegurar que no sea excesivo (con un alcance excesivo el proyecto podría tener una duración demasiado extensa e inclusive un costo elevado) y que no sea muy básico (con poca complejidad el proyecto no puede ser considerado como proyecto de fin de carrera).

La gestión del tiempo es importante también ya que el trabajo identificado en el alcance del proyecto debe ser planificado y controlado a través de actividades que

deben tener una determinada duración, teniendo en cuenta que se tienen ciertas restricciones de fechas a lo largo del proyecto (reuniones, entregas de avance, fecha límite del proyecto, etc.).

Las demás áreas del conocimiento de la dirección de proyectos (integración, costes, riesgos, calidad, recursos humanos, comunicaciones y adquisiciones) se están tomando en menor medida debido a que se trata de un proyecto pequeño realizado por una sola persona. Por ejemplo, para el caso de la gestión de riesgos se elaboró una lista con los principales riesgos del proyecto y se definió acciones de mitigación y contingencia. Si bien el uso de todas las áreas del conocimiento es relevante para la mayoría de proyectos, aplicarlas en su totalidad en este proyecto de fin de carrera implicaría aumentar demasiado la complejidad en la gestión del mismo.

Los procesos que se tomaron en cuenta, tanto para la gestión del alcance como para la gestión del tiempo se detallan a continuación:

Gestión del alcance

- *Planificación del alcance*
Proceso en el cual se detalla cómo se definirá, verificará y se controlará el alcance del proyecto y cómo se definirá y se creará el WBS. Si bien no se tendrá como entregable un documento de planificación del alcance para este proyecto, cada uno de estos puntos se describirá en el detalle de cada proceso.
- *Definición del alcance*
Proceso en el cual se definirá un enunciado del alcance del proyecto detallado que servirá como referencia para identificar el trabajo que tiene que ser realizado en el proyecto. Este alcance será definido inicialmente por la persona que realizará el proyecto pero debe ser consultado con conocedores y expertos en el área temática para asegurar que esté correctamente definido.
- *Crear WBS*
Proceso en el cual se identifican los principales entregables del proyecto y el trabajo del proyecto y se subdividen en componentes

más pequeños y fáciles de manejar. Para la creación del WBS se debe tomar en cuenta el enunciado del alcance definido para el proyecto. El WBS de este proyecto debe ser presentado en forma gráfica. En la Figura 1.3 se muestra el WBS de este proyecto.

- *Verificación del alcance*
Proceso en el cual se formaliza la aceptación de los productos entregables terminados del proyecto. Al ser un proyecto de fin de carrera, la verificación del alcance será realizada por el asesor del proyecto de tesis y el jurado encargado de evaluar el proyecto como parte de la obtención del título profesional.
- *Control del alcance*
Proceso por el cual se controlan los cambios en el alcance del proyecto. En este proyecto de fin de carrera los cambios en el alcance deben ser consultados con el asesor del proyecto de tesis y, una vez que se haya decidido realizar el cambio, éste debe ser actualizado en el enunciado del alcance y en el WBS.

Gestión del tiempo

- *Definición de las actividades*
Se detalla cómo se identifican las actividades necesarias para producir los productos entregables del proyecto. Para definir las actividades se toman como base los paquetes de trabajo definidos en el WBS y por cada paquete de trabajo se identifican las actividades necesarias para desarrollar cada paquete.
- *Establecimiento de la secuencia de las actividades*
Se identifican y se documentan las dependencias entre las actividades del cronograma. Para cada actividad se identifican sus predecesoras, en caso las tuviere.
- *Estimación de la duración de las actividades*
Proceso en el que se determina el esfuerzo (en horas hombre) que será necesario para completar cada actividad del cronograma. Para este proyecto, los períodos laborables serán de 4 horas de lunes a

domingo. Las estimaciones se harán en base a experiencias pasadas de desarrollo.

- Desarrollo del cronograma*

Se toman las actividades identificadas, su secuencia y su duración para la elaboración del cronograma del proyecto. El cronograma del proyecto se mostrará en un diagrama de Gantt. Los principales hitos del cronograma del proyecto se muestran en la Figura 1.4.
- Control del cronograma*

Proceso en el cuál se controlan los cambios en el cronograma del proyecto. El cronograma debe ser actualizado en caso se presente algún retraso en el proyecto o se requiera ajustar la duración de alguna actividad. También se debe ir actualizando el avance de cada actividad.

En la Tabla 1.1 se muestran los procesos de la dirección del proyecto, de acuerdo al PMBOK, agrupados por cada grupo de procesos.

Procesos de un Área de Conocimiento	Grupos de Procesos de Dirección de Proyectos				
	Grupo de Procesos de Iniciación	Grupo de Procesos de Planificación	Grupo de Procesos de Ejecución	Grupo de Procesos de Seguimiento y Control	Grupo de Procesos de Cierre
Gestión del Alcance del Proyecto		<ul style="list-style-type: none"> - Planificación del Alcance - Definición del Alcance - Crear WBS 		<ul style="list-style-type: none"> - Verificación del Alcance - Control del Alcance 	
Gestión del Tiempo del Proyecto		<ul style="list-style-type: none"> - Definición de las Actividades - Establecimiento de la Secuencia de las Actividades - Estimación de la Duración de las Actividades - Desarrollo del Cronograma 		<ul style="list-style-type: none"> - Control del Cronograma 	

Tabla 1.1: Procesos para la dirección del proyecto [27]

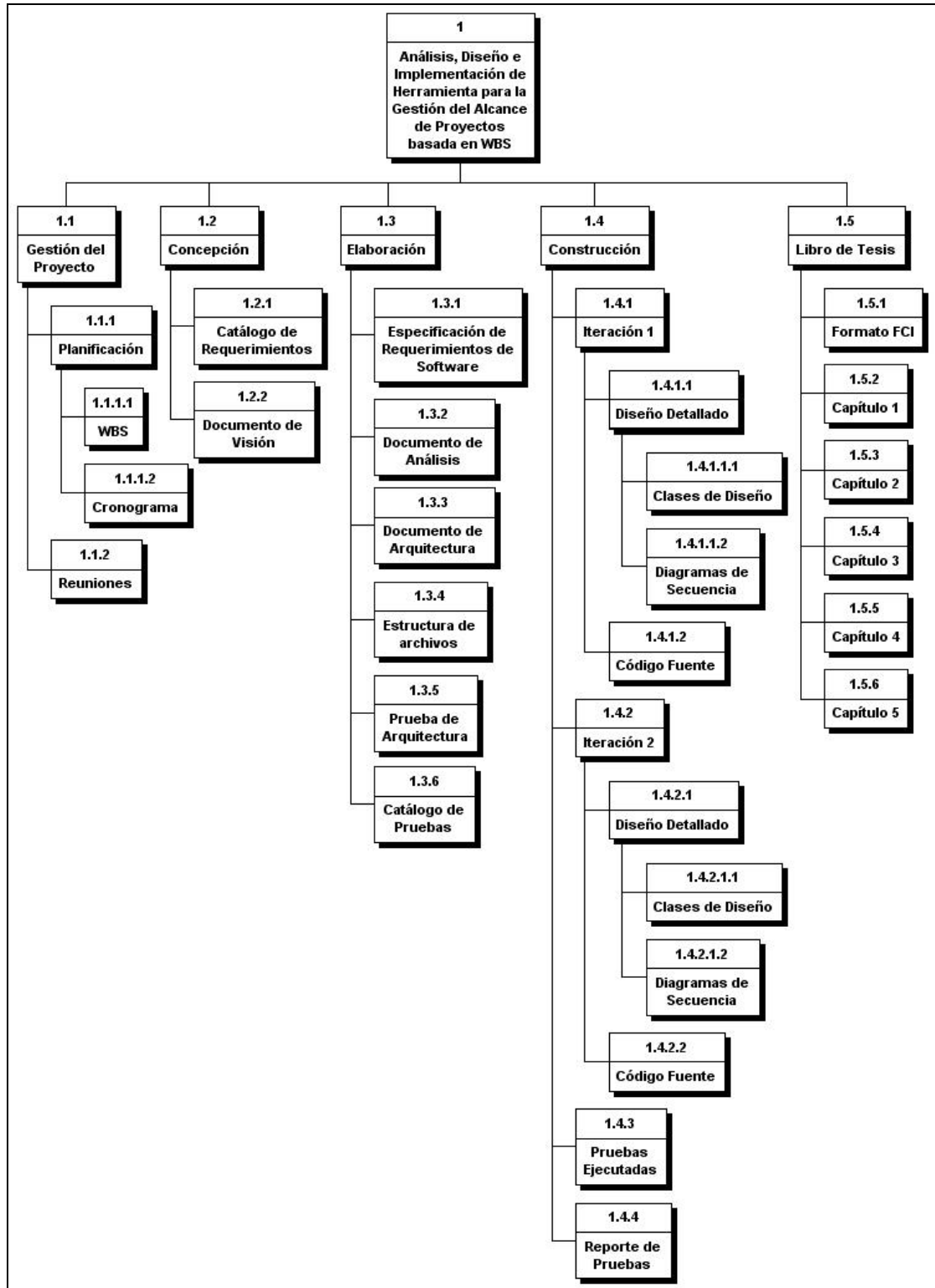


Figura 1.3: Modelo gráfico del WBS del proyecto [27]

1.4. Estado del arte

Un WBS simple puede ser elaborado a mano en una hoja o mediante una lista numerada en un procesador de texto. Sin embargo, actualmente existen herramientas de software que permiten definir el alcance de un proyecto a través de un WBS. Una de las más usadas es el *WBS Chart Pro* [5], una herramienta de planificación de proyectos que permite crear y mostrar proyectos partiendo de un modelo WBS que puede ser elaborado gráficamente. Cabe resaltar también que existe una extensión para *Microsoft Visio 2010* llamada *Microsoft Visio 2010 Add-in for WBS Modeler* [6] que permite visualizar y editar proyectos creados desde *Microsoft Project* o crear un nuevo proyecto a partir del modelo WBS. Ambas herramientas no son gratuitas, una versión de prueba del *WBS Chart Pro* puede ser descargada desde su página web pero restringe la información y la cantidad de nodos que puede tener el diagrama y el *Microsoft Visio 2010 WBS Modeler* requiere que se tenga instalado el *Microsoft Visio 2010*.

Revisando estos productos, se identificaron las siguientes características comunes:

- **Modelador gráfico del WBS**
Permite crear un modelo WBS gráficamente, insertando nodos hijos o padres para un nodo dependiendo de la dirección en la que se inserte. Además permiten editar ciertos atributos de los nodos desde el gráfico.
- **Capacidad ilimitada de nodos**
No se tienen restricciones para la cantidad de nodos que puede tener un WBS.
- **Almacenar especificación de cada nodo**
Se permite especificar cada nodo con atributos fijos que pueden ser ingresados por el usuario. También se permite configurar nuevos atributos para cada nodo.
- **Resumir y totalizar la información de cada nodo con la de sus hijos**
Cierta información numérica de los nodos (por ejemplo costo, trabajo, duración) se totaliza en los nodos padres usando la información de cada uno de sus hijos.

- **Exportar gráficamente el modelo WBS a formatos de imagen**
El gráfico del WBS puede ser exportado a diferentes formatos de imágenes.
- **Exportar el modelo WBS a HTML**
El gráfico del WBS puede ser exportado a HTML.
- **Personalización de la impresión del WBS**
La impresión del WBS se puede personalizar, permitiendo elegir qué información se imprime y en qué posiciones.
- **Personalización gráfica de los nodos**
Se puede personalizar el tamaño, colores e incluso la información que muestra cada nodo.
- **Asociar un archivo a cada nodo**
Se permite asociar un archivo adjunto para cada nodo e incluso se permite asociar un archivo para el proyecto.
- **Acceso al WBS para múltiples usuarios**
A través de la exportación en HTML del WBS, se permite que el modelo pueda ser consultado por múltiples usuarios.
- **Integración con Microsoft Project**
Al ser el Microsoft Project una de las principales herramientas para desarrollar cronogramas del proyecto, ambos productos permiten integrar los modelos WBS que generan con un cronograma de proyecto elaborado en Microsoft Project.

A pesar de todas las características que brindan estas herramientas, no se están tomando en cuenta funcionalidades críticas que una herramienta de modelado de WBS debería tener como permitir crear y visualizar el modelo WBS tanto en forma gráfica (cuya representación se encuentra en la Figura 1.2) como en un listado de viñetas (cuya representación se encuentra en la Figura 1.1), generar el diccionario del WBS con la especificación de cada nodo, trabajar con plantillas de WBS pre-elaboradas, entre otras. Cabe resaltar también que estas herramientas sólo se centran en la creación del WBS y no ofrecen la posibilidad de gestionar el alcance de un proyecto con una mayor facilidad.

Otro factor importante para discernir entre utilizar una herramienta u otra es el costo asociado a su adquisición y/o utilización. Para el caso del *WBS Chart Pro* debe adquirirse una licencia cuyo costo es de 199 USD. El *Microsoft Visio 2010 Add-in for WBS Modeler* requiere que se tenga instalado el *Microsoft Visio 2010*, cuyo precio varía entre los 250 USD y los 999 USD dependiendo de la versión que se utilice.

El resumen de las principales características de las herramientas y alternativas que han sido descritas en esta sección se muestra en la Tabla 1.2.

1.5. Descripción y sustentación de la solución

Dada la importancia de una adecuada gestión del alcance usando técnicas generalmente reconocidas como buenas prácticas y la limitada cantidad de herramientas que permitan esto (en términos de usabilidad, facilidad de uso y adopción de estándares), el proyecto propuesto en esta tesis es la realización del análisis, diseño e implementación de una herramienta para la gestión del alcance de proyectos basada en la técnica de WBS.

Las funcionalidades a implementar se pueden clasificar en 4 grupos: Creación del WBS, Verificación del Alcance, Control Integrado de Cambios y Generación de Documentación. La descripción de las funcionalidades cubiertas en cada uno de estos grupos se detalla a continuación:

Creación del WBS

La herramienta a desarrollar permitirá definir y organizar las fases, entregables y paquetes de trabajo de un proyecto usando un modelo WBS de n-niveles que puede ser visualizado gráficamente o siguiendo una estructura jerárquica. Cada nodo resultante podrá ser especificado mediante atributos fijos y atributos configurables. Se podrán especificar los atributos configurables por cada tipo de nodo (nodos genérico, entregable y paquete de trabajo).

La información del proyecto y de los modelos WBS podrá hacerse persistente, para su posterior consulta y/o modificación, grabándola preferentemente en un archivo u otra estructura portable.

Verificación del Alcance

En los nodos que representen entregables del proyecto se incluirán atributos que puedan ser actualizados tales como estado, porcentaje de avance, entre otros. Usando la información ingresada en estos atributos se mostrarán reportes de los entregables por estado y vistas con la situación actual de los entregables.

Control Integrado de Cambios

El control de cambios en el alcance del proyecto se logrará con el versionamiento de los modelos WBS. Se permitirá el registro de solicitudes de cambio del alcance y cada nueva versión estará asociada a una solicitud aprobada.

Generación de Documentación

Una vez elaborado el modelo WBS, se dará al usuario la posibilidad de generar los siguientes documentos:

- Diccionario del WBS, el cual incluye la estructura completa del WBS y la especificación de cada nodo. El diccionario puede ser visualizado o exportado en diferentes formatos.
- Modelo del WBS, en forma gráfica o en una estructura jerárquica.
- Plan de proyecto en Microsoft Project (mpp) con la información necesaria para la elaboración del cronograma, explotando así la posibilidad de mapear el modelo WBS con el cronograma del proyecto.

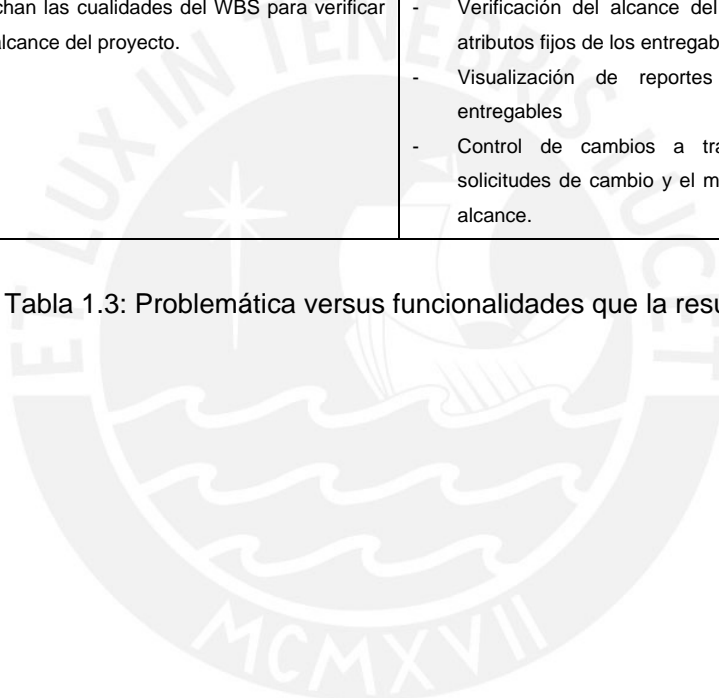
La solución descrita resuelve la problemática expuesta en la subsección 1.1 ya que da soporte a la gestión del alcance del proyecto, siguiendo las recomendaciones y buenas prácticas sugeridas en el PMBOK. En la Tabla 1.3 se muestran las funcionalidades del producto a desarrollar versus la problemática que resuelven.

Característica \ Alternativa	WBS Chart Pro	Visio 2010 Add-in for WBS Modeler	Procesador de textos	Bosquejo a mano
Modelador gráfico del WBS	Sí	Sí	No	No
Capacidad ilimitada de nodos	Sí	Sí	Sí	Sí
Almacenar especificación de cada nodo	Sí (los atributos configurables son finitos)	Sí	No	No
Resumir y totalizar la información de cada nodo con la de sus hijos	Sí	Sí	No	No
Exportación gráfica del WBS a imágenes	Sí	Sí	No	No
Exportación gráfica del WBS a HTML	Sí (muy completo, permite usar plantillas HTML)	No	No	No
Personalización de la impresión del WBS	Sí	Sí, pero limitado	No	No
Personalización gráfica de cada nodo	Sí (permite cambiar colores, tamaños e información visible para cada nodo)	Sí (permite cambiar los colores y tamaños de cada nodo)	No	No
Asociar un archivo a cada nodo	Sí	No	No	No
Acceso al WBS a múltiples usuarios	Sí (a través de su publicación en HTML)	No	No	No
Integración con Microsoft Project	Sí, pero limitado	Sí	No	No
Costo	199 USD	250 – 999 USD	No	No

Tabla 1.2: Resumen de las características de los productos y alternativas para crear un WBS [27]

Problemática	Funcionalidades que la resuelven
Complejidad para el uso de técnicas y herramientas para la gestión del alcance del proyecto.	- Creación de modelo WBS
WBS visto sólo como un diagrama de bloques.	- Especificación de nodos a través de atributos fijos y atributos configurables
Dificultad para elaborar la documentación a partir del alcance.	- Visualización y exportación del diccionario del WBS a diferentes formatos. - Generar cronograma del proyecto en formato de Microsoft Project (mpp)
Acceso de lectura al WBS a los involucrados del proyecto	- Exportación del modelo WBS en formatos de uso común. - Guardar proyectos para su posterior consulta y/o modificación
No se aprovechan las cualidades del WBS para verificar y controlar el alcance del proyecto.	- Verificación del alcance del proyecto a través de atributos fijos de los entregables - Visualización de reportes de estado de los entregables - Control de cambios a través del registro de solicitudes de cambio y el manejo de versiones del alcance.

Tabla 1.3: Problemática versus funcionalidades que la resuelven.



2. Análisis

En este capítulo se presenta la metodología para la construcción de la solución descrita en el capítulo anterior. Luego se detallan los principales requisitos del producto, y finalmente se evaluará la viabilidad de la construcción de la solución y se planteará el análisis base para la implementación del producto.

2.1. Definición de la metodología de la solución

Al tratarse de un proyecto de desarrollo de software, se usará una adaptación de RUP (*Rational Unified Process*) [21], que ya presenta fases y entregables específicos para el análisis, diseño y construcción de un producto software. RUP, al estar orientado a casos de uso, permite definir a través de diagramas de casos de uso las principales funcionalidades que debe contemplar la herramienta a desarrollar. La definición de la arquitectura de software de la solución se ve simplificada también debido a que RUP permite modelar una arquitectura basada en componentes. Estas ventajas, acompañadas del carácter gráfico que tiene la mayoría de entregables propuestos por la metodología, son las que orientaron la definición de la metodología de la solución hacia RUP.

Las fases y entregables, además del enfoque que se utilizará en este proyecto, se muestran a continuación [7]:

Concepción

Se definen los objetivos del proyecto así como el alcance necesario para resolver la problemática expuesta en el capítulo anterior. También se establecen las principales funcionalidades que tendrá el sistema. Los entregables que se presentaron en esta fase son:

- Documento de Visión: Se describe la problemática que será resuelta con este proyecto y se presenta un resumen del alcance y de las principales funcionalidades de la herramienta.
- Catálogo de Requisitos: Se muestra el listado de las principales funcionalidades de la herramienta, mostrando atributos como prioridad, dificultad y exigencia. Este documento no es un producto de trabajo de RUP, sin embargo es importante porque brinda una línea base para los requisitos que tendrá la herramienta.

Elaboración

En la fase de Elaboración se realiza el análisis para cumplir con los requisitos de la herramienta y se define la arquitectura a seguir en el desarrollo del software. Al completar esta fase se identifican y especifican todos los casos de uso. También se define la arquitectura que seguirá la aplicación y se prueba su factibilidad con el desarrollo de una prueba de arquitectura. El catálogo de pruebas también se realiza en esta fase. Los entregables que se elaboraron en esta fase son:

- Especificación de Requisitos de Software: Se describen todos los requisitos funcionales de la herramienta, agrupados en casos de uso. Cada caso de uso será especificado. También se presentan los requisitos no funcionales de la herramienta.
- Documento de Arquitectura de Software: Se muestra una visión general de la arquitectura de la herramienta y se describen las principales decisiones de arquitectura tomadas para el desarrollo del proyecto.
- Documento de Análisis: Se presentan modelos de análisis como diagramas de clases de análisis y diagramas de estado que solucionan la problemática descrita en el capítulo anterior. Adicionalmente se presenta el diccionario de clases y atributos para el modelo de clases propuesto.
- Prueba de Arquitectura: Se presenta una versión funcional de la herramienta con funcionalidades básicas que aseguren que la arquitectura planteada puede ser construida.

- Catálogo de Pruebas: Se definen las pruebas que tienen que realizarse para asegurar la correcta implementación de cada caso de uso.

Construcción

Esta fase se enfoca en la implementación de los casos de uso del sistema, así como otros aspectos de la aplicación que tienen que ser diseñados. Esta fase se realiza en forma incremental, con 2 iteraciones, obteniendo en la última una versión totalmente funcional de la herramienta. La primera iteración consistió en las funcionalidades básicas de la herramienta, mientras que en la segunda se desarrollaron las funcionalidades extra o deseables. Los entregables que se presentaron en cada iteración de esta fase son:

- Diseño Detallado: Se presenta el diagrama de clases de diseño de la solución y los diagramas de secuencia de los casos de uso implementados. En su conjunto, estos diagramas deben representar una abstracción del código fuente de la herramienta a desarrollar.
- Código Fuente: Se presenta el código fuente de una versión funcional de cada caso de uso de la iteración. Sobre esta versión se ejecutarán pruebas y de acuerdo al resultado de las mismas se realizarán las correcciones y ajustes necesarios antes de pasar a la siguiente iteración.

Transición

Para este proyecto no se tomará en cuenta la fase de Transición, debido a que no se requiere un pase a producción del software construido, al tratarse de un proyecto de fin de carrera que no involucra un uso comercial inmediato.

El resumen de los entregables por cada fase del proyecto se muestra en la Tabla 2.1.

Fase	Entregables
Concepción	<ul style="list-style-type: none"> - Catálogo de requisitos - Documento de Visión
Elaboración	<ul style="list-style-type: none"> - Especificación de Requisitos de Software - Documento de Arquitectura de Software - Documento de Análisis - Prueba de Arquitectura - Catálogo de Pruebas
Construcción	<ul style="list-style-type: none"> - Diseño Detallado - Código Fuente

Tabla 2.1: Resumen de entregables de la metodología de construcción del producto

2.2. Identificación de requisitos

En esta sección se presentan los requisitos de la herramienta. Estos requisitos se basan en las funcionalidades encontradas en productos similares revisados, unificándolas y agregando la orientación a las técnicas y estándares generalmente reconocidos como buenas prácticas del PMBOK para gestión del alcance de proyectos. El detalle de estos requisitos se encuentra en el Anexo A: Catálogo de Requisitos.

- **Administración de Proyectos**

Se permite crear proyectos, cuyo alcance puede ser gestionado desde la herramienta. Estos proyectos pueden ser almacenados en archivos XML, ya que se trata de un formato portable y legible que permite construir una estructura jerárquica personalizada, adecuada para la necesidad de almacenamiento de los modelos WBS. Los proyectos almacenados en un archivo pueden ser cargados en la herramienta en cualquier momento.

- **Creación de Modelos WBS**

La herramienta permite crear modelos WBS, a partir de la creación de nodos genéricos, entregables y paquetes de trabajo. Los nodos del modelo pueden ser visualizados a través de una estructura de árbol o gráficamente.

- **Especificación de Nodos del WBS**

La herramienta permite especificar los nodos del modelo WBS a través de atributos fijos y atributos configurables por cada tipo de nodo. Además se permite añadir archivos adjuntos a cada nodo.

- **Verificación del Alcance**

La verificación del alcance del proyecto se logra a través de los atributos fijos de los entregables estado y porcentaje de avance. La información de estos atributos puede ser consultada a través de reportes que muestran el estado de cada entregable.

- **Control del Alcance**

La herramienta permite controlar el alcance del proyecto a través de las solicitudes de cambio y las versiones del proyecto. En una versión del proyecto se pueden registrar múltiples solicitudes de cambio y al aprobar alguna de estas solicitudes se genera automáticamente una nueva versión con los cambios realizados. Las solicitudes de cambio tienen su propio modelo WBS, el cual se puede modificar para reflejar los cambios que se requieran.

- **Generación de Documentación del Proyecto**

La herramienta permite visualizar y exportar el WBS y documentos asociados al modelo tales como el diccionario del WBS, el cronograma del proyecto, el diseño de esquema del WBS y el gráfico del WBS.

Los requisitos descritos en esta sección solucionan la problemática de brindar una herramienta completa, que adopte estándares reconocidos para la gestión del alcance de un proyecto, dando funcionalidades para soportar tres de los cinco procesos de Gestión del Alcance propuestos en el PMBOK (Creación del WBS, Verificación del Alcance y Control del alcance).

Con los diferentes atributos fijos y configurables que puede almacenar cada nodo del WBS se habilita la posibilidad de que se explote al máximo las cualidades que brinda el modelo, permitiendo el monitoreo del estado de los entregables del proyecto, la generación del diccionario del WBS y la generación de la estructura del cronograma del proyecto. De esta manera también se simplifica la complejidad de elaborar la documentación referente al alcance desde cero.

La exportación gráfica y en esquema del modelo WBS facilita la publicación y distribución del trabajo a realizar en el proyecto, lo que permite que los recursos e interesados en el proyecto validen si la definición del alcance es correcta.

El registro de solicitudes de cambio y las nuevas versiones del modelo WBS que se generan al aprobar cada solicitud, permiten tener un registro de cada versión del WBS y de los cambios significativos que se pueden realizar al alcance de un proyecto entre una versión y otra. Además se da la posibilidad de tener un bosquejo de lo que sería la nueva versión del modelo WBS antes de aprobar la solicitud, lo que permitiría mostrar previamente el cambio a los interesados, sin modificar el modelo actual.

2.3. Identificación de casos de uso

En esta sección se presentan los casos de uso identificados a partir de los requisitos de la herramienta. Estos casos de uso han sido agrupados en cinco paquetes, cada uno de estos relacionado a un tipo de funcionalidad. Al tratarse de una herramienta para la gestión del alcance del proyecto, el único actor identificado para la utilización de las diferentes funcionalidades es el Jefe de Proyectos.

Paquete de Administración

Este paquete contiene los casos de uso que corresponden a la administración de proyectos de la herramienta.

Los casos de uso contenidos en este paquete son:

- Mantener Proyectos
- Cargar Proyecto
- Almacenar Proyecto.

El caso de uso Mantener Proyectos permite la creación de un proyecto en la herramienta, así como la posterior edición de los datos específicos del proyecto. El caso de uso Almacenar Proyecto permite guardar de forma persistente (en un archivo XML) un proyecto modelado en la herramienta. El caso de uso Cargar Proyecto permite abrir un proyecto ya existente (guardado en un XML con el caso de uso Almacenar Proyecto), a fin de visualizarlo y/o continuar con su edición.

Paquete de creación de WBS

Este paquete contiene los casos de uso para la creación y especificación de los nodos de un modelo WBS.

Los casos de uso contenidos en este paquete son:

- Mantener Nodos Genéricos
- Mantener Entregables
- Mantener Paquetes de Trabajo
- Explorar Nodos del WBS.
- Visualizar Modelo WBS
- Mantener Atributos de Nodo

Los casos de uso Mantener Nodos Genéricos, Mantener Entregables y Mantener Paquetes de Trabajo permiten la creación, edición y eliminación de los diferentes tipos de elementos de un modelo WBS. El caso de uso Visualizar Modelo WBS permite El caso de uso Explorar Modelo WBS permite navegar jerárquicamente entre los elementos del modelo. El caso de uso Visualizar Modelo WBS muestra gráficamente un modelo, respetando la jerarquía entre elemento. Finalmente, el caso de uso Mantener Atributos de Nodo permite configurar los atributos que se muestran para los tipos de elementos del WBS (nodo genérico, entregable y paquete de trabajo).

Paquete de Verificación del Alcance

Este paquete contiene los casos de uso para la verificación del alcance del proyecto.

Los casos de uso contenidos en este paquete son:

- Actualizar Estado de Entregable
- Visualizar Vista de Estado de Entregables
- Visualizar Reporte de Estado de Entregables

El caso de uso Actualizar Estado de Entregable permite editar los campos de estado de un entregable (porcentaje de avance y estado), a fin de reflejar el progreso en la elaboración del mismo. Los casos de uso Visualizar Vista de Estado de Entregables y Reporte de Estado de Entregables muestran información consolidada del estado de los entregables del proyecto.

Paquete de Control del Alcance

Este paquete contiene los casos de uso para el control de las versiones del proyecto.

Los casos de uso contenidos en este paquete son:

- Mantener Solicitudes de Cambio
- Actualizar Estado de Solicitud de Cambio
- Explorar Versiones del Proyecto
- Visualizar Solicitud de Cambio

El caso de uso Mantener Solicitud de Cambio permite la creación, edición y eliminación de las solicitudes de cambio de alcance del proyecto. El caso de uso Actualizar Estado de Solicitud de Cambio permite aceptar o rechazar una solicitud de cambio y crea una nueva versión del modelo WBS en caso sea aceptada. El caso de uso Explorar Versiones del Proyecto permite navegar entre las versiones del modelo WBS. Finalmente, el caso de uso Visualizar Solicitud de Cambio permite mostrar el detalle de las solicitudes de cambio de alcance del proyecto.

Paquete de Generación de Documentación

Este paquete contiene los casos de uso para la generación de documentos de la herramienta

Los casos de uso contenidos en este paquete son:

- Exportar Gráfico del WBS
- Visualizar Diseño de Esquema del WBS
- Exportar Diseño de Esquema del WBS
- Visualizar Diccionario de WBS
- Exportar Diccionario de WBS
- Generar Cronograma del Proyecto

El caso de uso Exportar Gráfico del WBS permite guardar un modelo gráfico de WBS en un archivo de imagen de uso común (PNG, JPG y BMP). Los casos de uso Visualizar Diseño de Esquema del WBS y Exportar Diseño de Esquema del WBS permiten visualizar y exportar un modelo WBS en una estructura jerárquica. Los casos de uso Visualizar Diccionario de WBS y Exportar Diccionario de WBS permiten mostrar y exportar la información detallada de cada uno de los nodos del modelo. Finalmente,

el caso de uso Generar Cronograma del Proyecto permite la creación del cronograma de proyecto en formato de Microsoft Project, utilizando la información especificada para cada uno de los nodos.

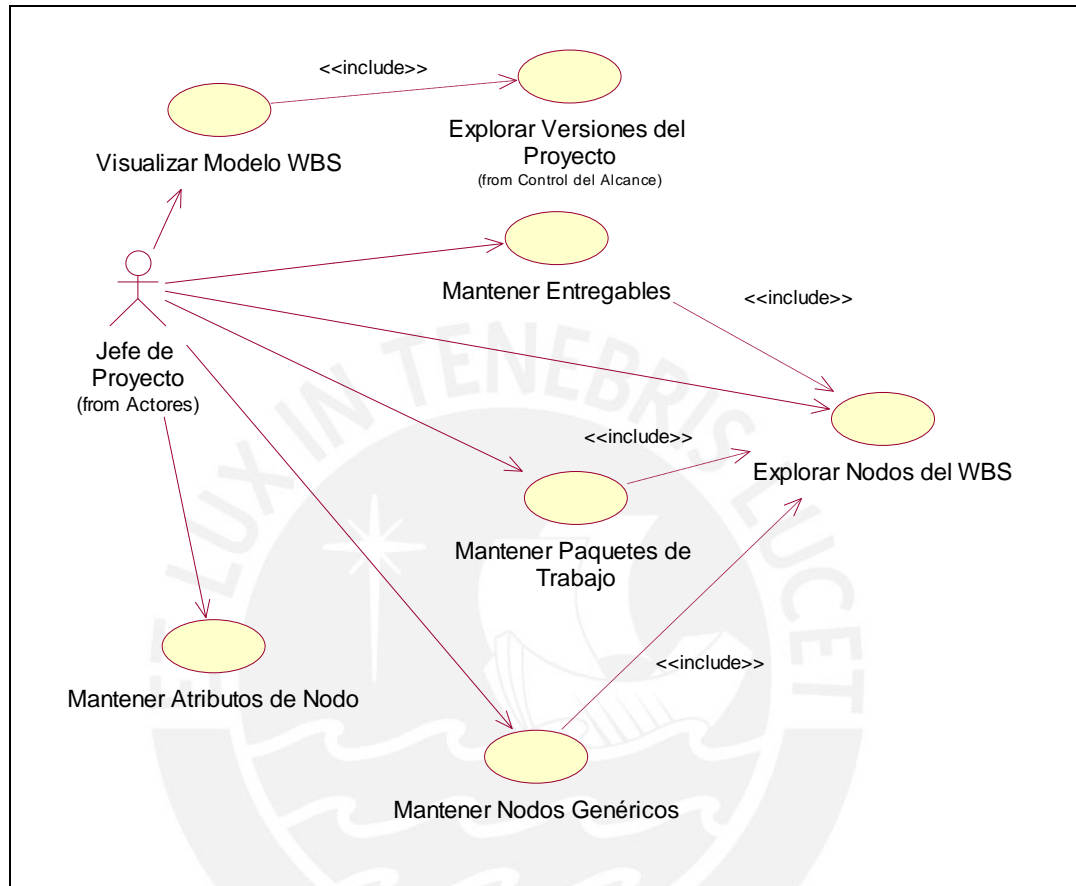


Figura 2.1: Diagrama de casos de uso para el Paquete de creación de WBS

En la Figura 2.1 se muestra el diagrama de casos de uso para el paquete de creación de WBS, al tratarse de la funcionalidad más importante de la herramienta.

La especificación detallada de los casos de uso y los diagramas de cada paquete se encuentran en el Anexo B: Especificación de Requisitos de Software.

2.4. Análisis de la Solución

2.4.1. Viabilidad de la herramienta

En esta sección se evaluará la viabilidad de la herramienta a implementar, realizando un análisis técnico y económico de la solución propuesta.

- **Análisis técnico**

La viabilidad técnica de la solución se basa en la posibilidad de implementar la herramienta que se plantea en este proyecto con las tecnologías existentes. La gran cantidad de herramientas CASE y de gestión de proyectos que han sido implementadas indican que es factible desarrollar herramientas de este tipo. Existen empresas especializadas en el desarrollo de herramientas CASE (IBM, Microsoft) y se han realizado implementaciones de este tipo para proyectos de fin de carrera en nuestro país como por ejemplo las herramientas JUML [22], EPCU Case [20] o JSPEM [23], desarrolladas como proyectos de tesis en la Pontificia Universidad Católica del Perú, incluso siendo esta última una herramienta enfocada a la gestión de proyectos de software, al igual que este proyecto de fin de carrera.

- **Análisis económico**

Para el análisis, diseño e implementación de esta solución no se requieren equipos ni licencias costosas, ya que se usarán herramientas libres, no propietarias para la implementación del software y la elaboración de diagramas y documentos. Cabe resaltar también que estas herramientas tienen requisitos de sistema accesibles, por lo que el hardware requerido no representa mayor inconveniente.

Por lo tanto, se concluye que la implementación de la herramienta es factible al no tener dificultades técnicas o económicas.

2.4.2. Definición de la herramienta

En esta sección se presentará el análisis base para la implementación de la herramienta, tomando como referencia los requisitos identificados en la subsección 2.2 de este capítulo.

2.4.2.1. Diagramas de Clases de Análisis

El diagrama de clases es el diagrama más importante en la etapa de análisis. En la Figura 2.2 se presenta el diagrama de clases de análisis de la herramienta. El diccionario de clases de análisis se encuentra el anexo C: Diccionario de Clases de Análisis. A continuación se presenta una descripción de las clases del diagrama:

Proyectos

La clase Proyecto representa a un proyecto creado en la herramienta. A cada proyecto se le puede adjuntar varios archivos, relación que se muestra en la agregación de la clase Proyecto con la clase ArchivoAdjunto.

Estructura de Nodos

La clase ModeloWBS tiene una composición de clases Nodo. Una clase Nodo puede representar a un nodo genérico, un entregable o un paquete de trabajo. El detalle de la estructura de la clase Nodo se explicará más adelante. Además, cada nodo puede tener atributos configurables, relación que se muestra en la agregación de la clase Nodo con la clase Atributo. A un nodo se le pueden adjuntar varios archivos, relación que se muestra en la agregación de la clase Nodo con la clase ArchivoAdjunto.

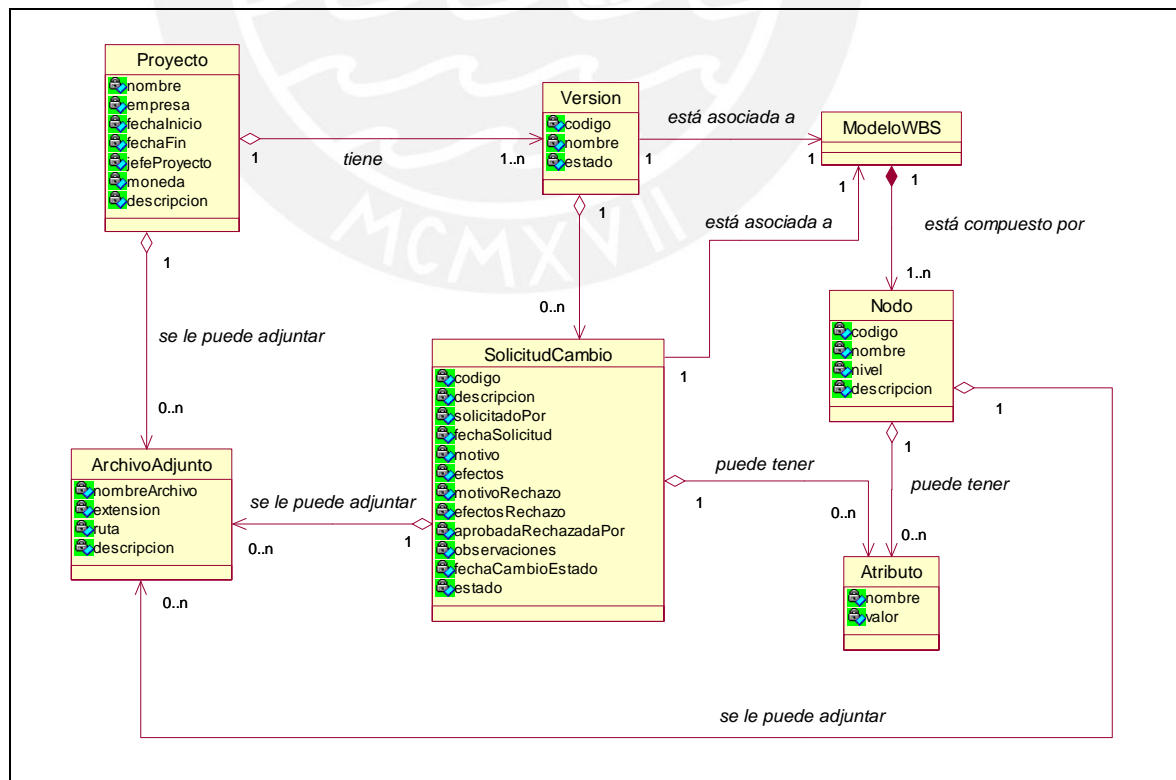


Figura 2.2: Diagrama de Clases de un proyecto tipo elaborado en la herramienta [27]

Solicitudes de Cambio

Las solicitudes de cambio del proyecto son representadas por la clase SolicitudCambio. La posibilidad de adjuntar archivos a cada solicitud de cambio se refleja en la agregación entre las clases SolicitudCambio y ArchivoAdjunto. Las solicitudes de cambio, al tener como consecuencia un cambio de alcance del proyecto y por lo tanto un cambio en el modelo WBS, están asociadas a un modelo WBS. La relación se puede apreciar entre las clases SolicitudCambio y ModeloWBS. Además, cada solicitud de cambio puede tener atributos configurables, relación que se muestra en la agregación de la clase SolicitudCambio con la clase Atributo.

Versiones

Las múltiples versiones del alcance de cada proyecto se representan en la relación de agregación de la clase Proyecto con la clase Version. Cada versión puede tener una o muchas solicitudes de cambio, relación representada en la agregación de las clases Version y SolicitudCambio. Cabe resaltar que para cada versión sólo puede existir una solicitud de cambio aprobada ya que cada aprobación de una solicitud de cambio genera automáticamente un cambio de versión. Una versión del alcance y tiene asociado un modelo WBS.

Para simplificar la representación de un modelo WBS se ha realizado un análisis para obtener un metamodelo que posteriormente servirá como base para obtener la estructura de datos que representará a los nodos de los proyectos creados en la herramienta. En la Figura 2.3 se muestra el diagrama de clases de análisis del metamodelo.

El elemento principal del metamodelo es la clase Nodo. La clase Nodo vendría a ser la representación de un nodo del modelo WBS. La clase Nodo es una clase abstracta. Es por este motivo que la clase Nodo tiene como clases hijas a NodoGenerico, Entregable y PaqueteTrabajo (Hereda de Entregable, que es también un Nodo). Cada Nodo (sea un nodo genérico, un entregable o un paquete de trabajo) tiene atributos fijos (detallados como miembros de la clase) y configurables. Los atributos configurables son representados por la clase Atributo. Como sólo existe un tipo de relación entre los nodos del modelo que vendría a ser la relación de jerarquía, es decir la relación Padre-Hijo entre nodos, no se identificó una clase que refleje esta relación. Sin embargo, sí se puede observar la relación de jerarquía mediante la composición que tiene la clase Nodo con sí misma. Un nodo puede tener muchos hijos, que vienen

a ser representados como una composición de Nodos. De esta relación también se puede concluir que un nodo sólo puede tener un padre (o puede no tenerlo para el caso del nodo raíz).

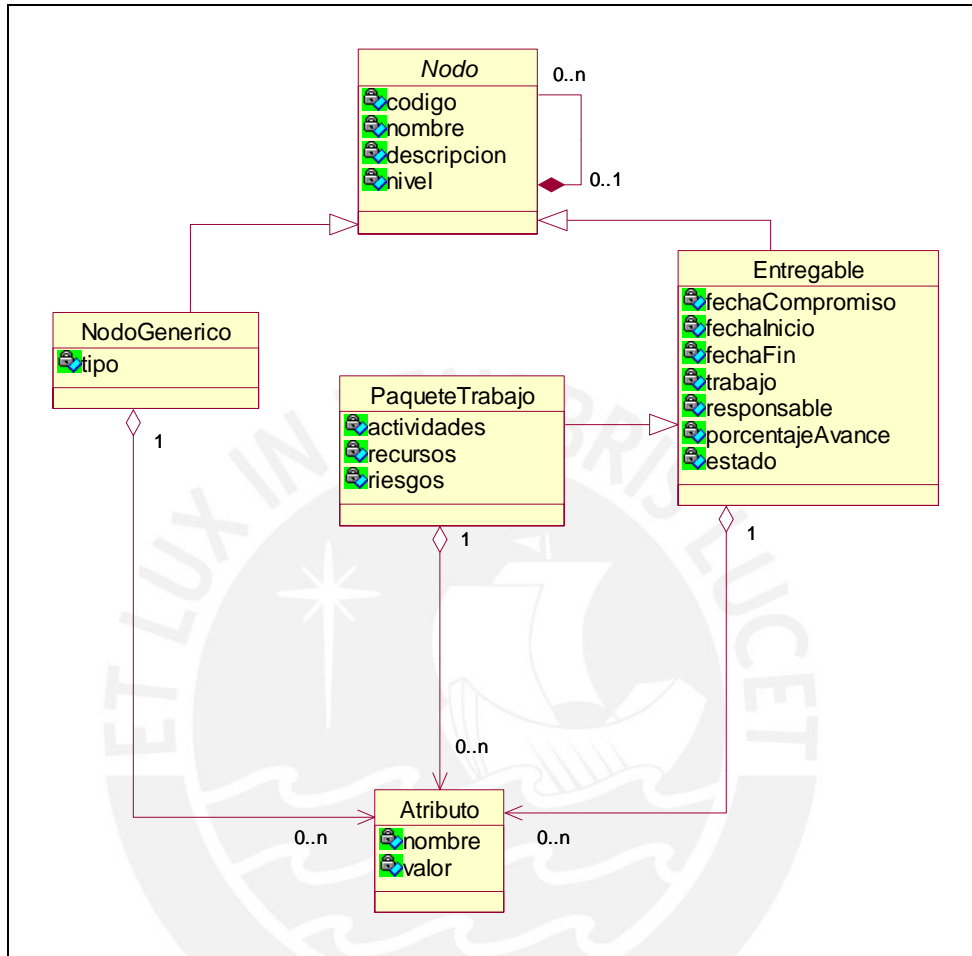


Figura 2.3: Diagrama de Clases del Metamodelo para un modelo WBS [27]

Para aclarar el uso del metamodelo planteado se puede desarrollar un ejemplo. Un proyecto de RUP tiene fases definidas para el desarrollo de un software (Concepción, Elaboración, Construcción y Transición). Estas fases pueden ser representadas por la clase *NodoGenerico* del metamodelo, ya que las fases representan subdivisiones del alcance del proyecto. A su vez, cada una de estas fases tiene diferentes entregables. Estos entregables pueden ser representados como clases *Entregable* o clases *PaqueteTrabajo* (si se encuentran en el último nivel).

Entonces un proyecto RUP puede ser representado por las clases del metamodelo de la siguiente manera:

1. Proyecto RUP (Clase NodoGenerico)
 - 1.1. Concepción (Clase NodoGenerico)
 - 1.1.1. Catálogo de Requisitos (Clase PaqueteTrabajo)
 - 1.1.2. Documento de Visión (Clase PaqueteTrabajo)
 - 1.2. Elaboración (Clase NodoGenerico)
 - 1.2.1. ERS (Clase Entregable)
 - 1.2.1.1. Diagramas de casos de uso (Clase PaqueteTrabajo)
 - 1.2.1.2. Especificación de casos de uso (Clase PaqueteTrabajo)
 - 1.2.2. Diagrama de Clases de Análisis (Clase PaqueteTrabajo)
 - 1.3. Construcción (Clase NodoGenerico)
 - 1.3.1. Software (Clase PaqueteTrabajo)
 - 1.3.2. Documento Diseño Detallado (Clase Entregable)
 - 1.3.2.1. Diagramas de Clases de Diseño (Clase PaqueteTrabajo)
 - 1.3.2.2. Diagramas de Secuencia (Clase PaqueteTrabajo)
 - 1.4. Transición
 - 1.4.1. Software Implantado (Clase Paquete Trabajo)

El nodo 1 (Proyecto RUP) es representado por una clase `NodoGenerico` sin padre, que tiene una composición de clases `NodoGenerico` que representan a las fases del proyecto. El nodo 1.1 (Concepción) es representado por una clase `NodoGenerico` que a su vez tiene en su composición a los nodos 1.1.1 (Catálogo de Requisitos representado por la Clase `PaqueteTrabajo`) y 1.1.2 (Documento de Visión representado por la Clase `PaqueteTrabajo`). De esta manera se puede seguir representando el proyecto usando como base sólo las clases del metamodelo.

3. Diseño

En este capítulo se presenta la arquitectura de la solución y luego se describe el diseño de interfaz gráfica que tiene el producto desarrollado.

3.1. Arquitectura de la solución

En esta sección se presentan las decisiones tomadas en cuanto a la arquitectura de la herramienta construida y su debida justificación.

3.1.1. Definición de la arquitectura

La herramienta implementada es *standalone* y, a diferencia de otras que necesitan un mayor despliegue, se ejecuta localmente sobre una sola computadora. Las razones que respaldan esta decisión son las siguientes:

- No es necesario restringir la ejecución de la herramienta a una conexión de red o a la utilización de un navegador de internet.
- Una herramienta *standalone* puede utilizar directamente los recursos del sistema operativo que la contiene (capacidades de interfaz gráfica y acceso a sistema de archivos).

- La gestión del alcance del proyecto es realizada por el jefe de proyecto, rol que recae en una sola persona. De ser necesario compartir la información del proyecto esto puede lograrse a través de los archivo XML que genera la herramienta.
- Otras herramientas CASE y de gestión de proyectos también son *standalone* y persisten su información en archivos locales de la computadora en la que se ejecutan.

En cuanto a la arquitectura de software a seguir para la construcción del producto, se usará una arquitectura basada en el patrón MVC [8] (Modelo-Vista-Controlador o en inglés *Model-View -Controller*). Los usuarios de la herramienta requieren construir y visualizar de forma gráfica el modelo WBS. Entonces, por un lado se tendrá la interfaz gráfica de usuario (Vista) y por otro lado los datos que componen el WBS (Modelo). El controlador servirá como una división lógica entre la vista y el modelo. Se encargará de recibir las entradas y acciones que el usuario ingresa (desde la interfaz gráfica) y de acuerdo a éstas actualizará convenientemente el modelo y la vista. Las capas de la herramienta y la función de cada capa se muestran a continuación:

- **Vista:** Se encarga de mostrar gráficamente al usuario el modelo WBS y las opciones que tiene para su edición. También se encarga de almacenar las características gráficas (posición, tamaño, etc.) de cada elemento del WBS.
- **Modelo:** Se encarga de almacenar lógicamente los datos de un proyecto y del modelo WBS. El modelo sólo puede ser modificado a través del controlador.
- **Controlador:** Es la capa lógica que separa a la vista y el modelo. Toda interacción que el usuario tenga con el modelo se debe hacer a través de la vista (interfaz gráfica de usuario) que a su vez invoca al controlador. El controlador actualizará la vista y el modelo como respuesta a la interacción del usuario. [9]

3.1.2. Diagrama de componentes

Las agrupaciones lógicas de componentes de la herramienta y sus principales componentes se pueden observar en los diagramas de la Figura 3.1 y la Figura 3.2.

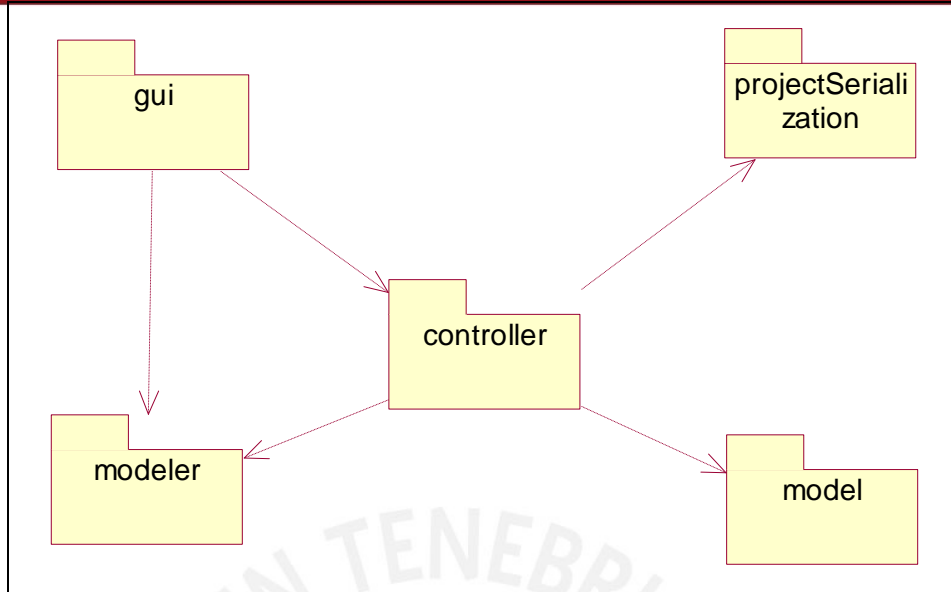


Figura 3.1: Diagrama de paquetes de la herramienta

Paquete: controller	
Descripción	
Paquete que representa a la separación lógica entre la vista y el modelo. Es invocado por el paquete gui y contiene clases que invocan al paquete model. Permite la administración de los proyectos y los modelos WBS de la herramienta.	
Componente	Descripción
controller	Componente encargado de actualizar las clases del modelo.

Paquete: gui	
Descripción	
Paquete que contiene las clases de la interfaz gráfica de usuario. Invoca a las clases del paquete controller para actualizar el modelo. También se encarga de invocar al componente report Generator, que representa a un visualizador gráfico de reportes, y al componente 2D graphic library, que permite la visualización gráfica del modelo WBS.	
Componente	Descripción
generic GUI	Componente que se encarga de la visualización en pantalla de la interfaz gráfica de usuario.

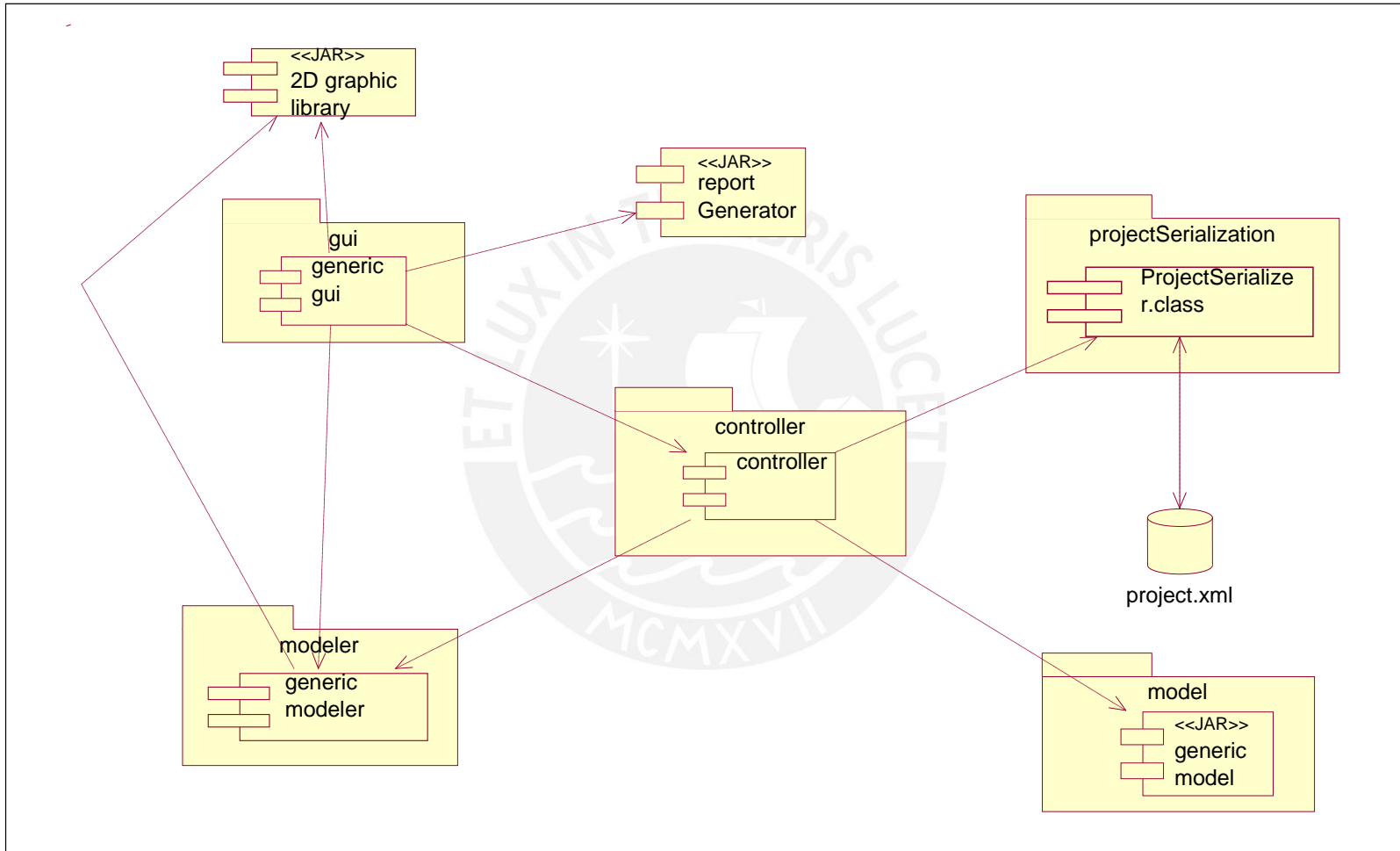


Figura 3.2: Diagrama de componentes de la herramienta [27]

Paquete: model	
Descripción	
Paquete que contiene las clases necesarias para almacenar la información del proyecto y del WBS. Es invocado por el paquete controller.	
Componente	Descripción
generic model	Componente que representa a una clase genérica del modelo.

Paquete: modeler	
Descripción	
Paquete que contiene las clases necesarias para almacenar la información gráfica de los elementos de los WBS modelados en la herramienta. El paquete gui se encarga de invocar a los componentes de este paquete.	
Componente	Descripción
generic modeler	Componente que representa a una clase que almacena los atributos gráficos de un elemento del WBS.

Paquete: projectSerialization	
Descripción	
Paquete que sirve para almacenar la información de un proyecto en un archivo del disco duro. Es invocado por el paquete controller.	
Componente	Descripción
ProjectSerializer.class	Componente que está compuesto por una clase que almacena la información del proyecto en un archivo del disco duro.

Otros componentes	
Componente	Descripción
2D graphic library	Componente que representa a una librería de clases que permite visualizar de forma gráfica el WBS.
project.xml	Archivo en formato XML generado por la herramienta para leer y almacenar proyectos.
report Generator	Componente que representa a una librería que permite visualizar reportes y generar documentos en diferentes formatos.

Los paquetes que representan a la vista son gui y presentation. El paquete gui invoca al paquete controller para actualizar los datos del modelo. A su vez, también utiliza las librerías report Generator y 2D graphic library para la visualización de reportes, generación de documentación y la visualización gráfica de los diagramas WBS. El paquete modeler contiene clases que almacenan los atributos gráficos de cada elemento del WBS. Este paquete es invocado por el paquete controller para almacenar los atributos gráficos del modelo de forma persistente.

El paquete model contiene las clases que permiten almacenar la información de un proyecto y de sus modelos WBS. Para los elementos de un diagrama WBS se usan las clases del metamodelo identificadas en la subsección 2.3.2, en el capítulo de Análisis. Para la información general del proyecto (versiones, solicitudes de cambio, archivos adjuntos) se usan las clases de un proyecto elaborado en la herramienta, identificadas en la misma subsección.

El controlador es representado por el paquete controller, que viene a ser la capa lógica entre la vista y el modelo.

3.1.3. Beneficios de la arquitectura

Las características de la arquitectura propuesta en la subsección 3.1.2 brinda las siguientes ventajas:

Característica	Descripción
Mantenibilidad	Al presentar 3 capas independientes entre sí se hace más sencilla la modificación y mejora de cada una de estas capas sin impactar significativamente en la herramienta.
Despliegue / Instalación	El despliegue de la herramienta es limitado pero suficiente ya que se trata de una aplicación <i>standalone</i> , que puede ser ejecutada independientemente en cada computador.
Independencia entre los datos y la interfaz gráfica	El uso del patrón MVC permite abstraer lo que el usuario ve (vista, para la herramienta sería el diagrama WBS y las opciones para su manipulación) de la estructura de datos compleja que se utiliza para almacenar la información.

3.2. Diseño de Interfaz Gráfica

En esta sección se presentará la estructura de la interfaz gráfica de la solución y los criterios utilizados para su diseño.

- **Pantalla principal**

La pantalla principal sigue la estructura común de una herramienta que permite modelado gráfico y se muestra en la Figura 3.3. Las secciones principales de la pantalla principal de la herramienta son las siguientes:

1. **Menú principal:** Barra de menú en la que se agrupan todas las funcionalidades de la herramienta.
2. **Barra de herramientas:** Proporciona íconos para el acceso directo a las funciones más importantes de la herramienta.
3. **Árbol de navegación de versiones:** Permite al usuario navegar entre las versiones del alcance de un proyecto. Muestra además las solicitudes de cambio de cada versión y los archivos adjuntos de cada solicitud de cambio. Es útil para el usuario ya que le permite consultar las versiones pasadas del modelo WBS o abrir la versión vigente para su modificación.
4. **Árbol de navegación de nodos del WBS:** Al abrir un modelo WBS, en esta sección se muestran, de forma jerárquica, todos los nodos que componen el modelo. De esta manera se permite al usuario consultar todos los nodos del modelo e ingresar al detalle de alguno si lo requiere.
5. **Área de trabajo:** En esta sección de la pantalla se abren los diagramas WBS.

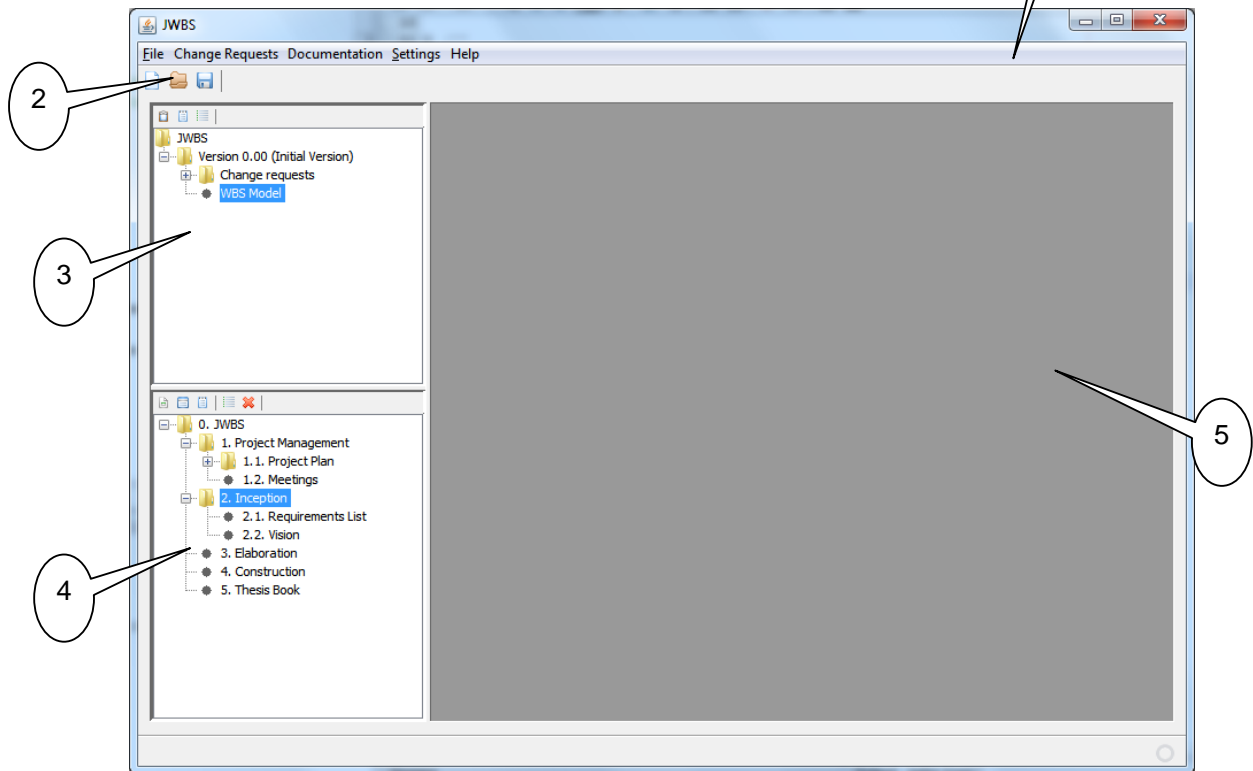


Figura 3.3: Pantalla principal de la interfaz de usuario [27]

- **Pantalla de ingreso de información**

Las pantallas para ingreso de información son ventanas modales que se abren al centro de la ventana principal. Estas ventanas tienen un tamaño menor al de la ventana principal. En estas ventanas se muestran campos obligatorios y opcionales que pueden ser llenados o no según sea el caso. Si es necesario mostrar más campos sin querer hacer excesivo el tamaño de la ventana se usan pestañas, en las que se agrupa información similar. Un ejemplo de este tipo de ventanas se muestra en la Figura 3.4.

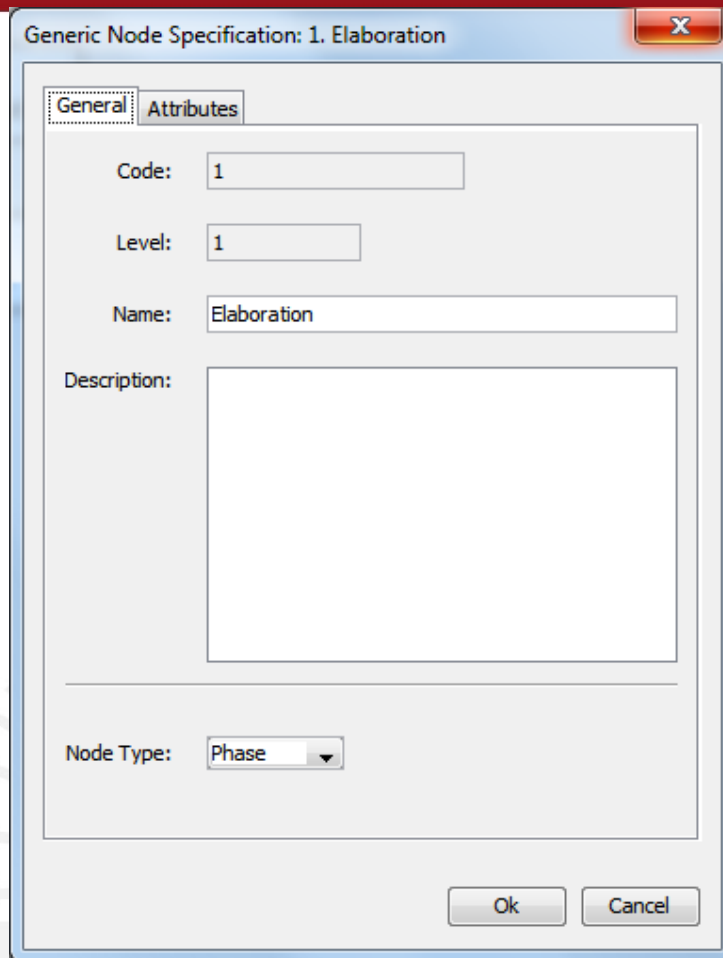


Figura 3.4: Pantalla de ingreso de información [27]

- **Árbol de navegación de versiones**

En el árbol de navegación de versiones se pueden visualizar todas las versiones de un proyecto y el contenido asociado a cada versión. En el contenido de la versión se muestra su modelo WBS, así como las solicitudes de cambio y los archivos adjuntos de la versión. Cada solicitud de cambio puede ser expandida (al igual que la versión) para consultar su modelo WBS. En la figura 3.5 se muestra el árbol de navegación de versiones.

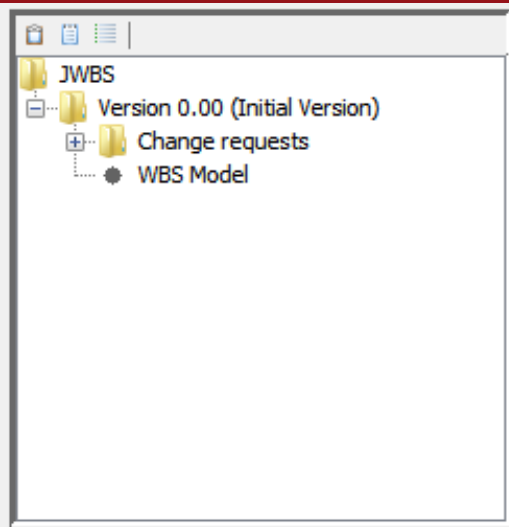


Figura 3.5: Árbol de navegación de versiones del proyecto [27]

- **Árbol de navegación de nodos**

El árbol de navegación de nodos permite al usuario consultar los nodos de un modelo WBS, así como la creación y especificación de nodos del modelo, a través de los botones de la parte superior del árbol. Al expandir un nodo se muestran sus hijos. Para que este árbol esté habilitado, previamente se tiene que seleccionar un modelo WBS en el árbol de navegación de versiones. La figura 3.6 muestra un ejemplo del árbol de navegación de nodos.

- **Modelo gráfico del WBS**

El modelo gráfico de un WBS se muestra en el área de trabajo de la herramienta, en ventanas MDI (por sus siglas en inglés: *Multi Document Interface*). El usuario puede visualizar un modelo gráfico cada vez que selecciona un modelo WBS desde el árbol de versiones. Al tratarse de una ventana MDI, el usuario puede minimizarla, maximizarla y/o cerrarla cuando lo necesite. También presenta una barra de herramientas en la parte superior, que permite al usuario generar documentación asociada al modelo WBS. En la figura 3.7 se muestra un ejemplo del modelo gráfico de un WBS.

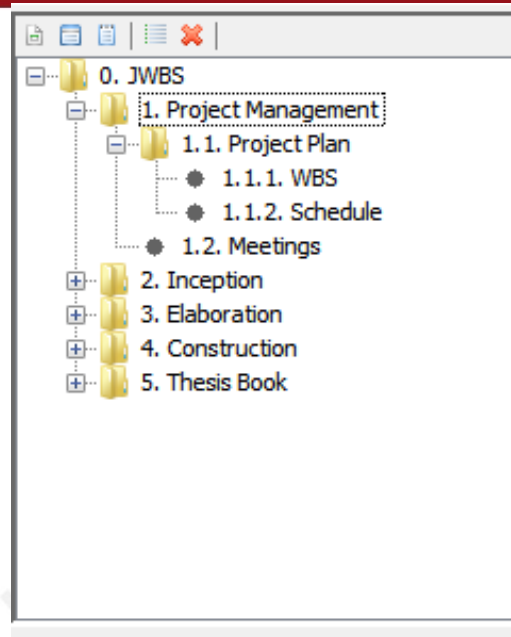


Figura 3.6: Árbol de navegación de nodos de un modelo WBS [27]

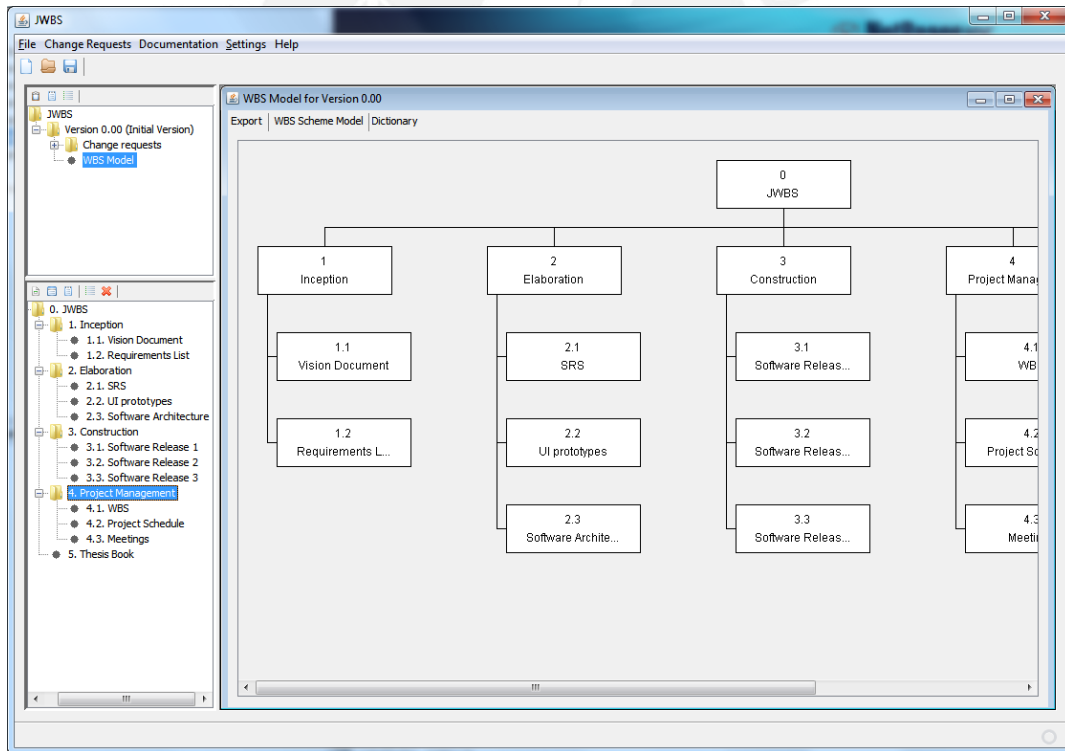


Figura 3.7: Modelo gráfico de un WBS [27]

- **Pantalla de configuración de atributos de nodo**

Cada tipo de nodo (nodo genérico, entregable o paquete de trabajo) puede tener un conjunto de atributos configurables. Estos atributos se configuran desde una ventana modal, en la que se permite al usuario seleccionar el tipo de nodo cuyos atributos se desea modificar. Una vez seleccionado el tipo de nodo, el usuario puede agregar, modificar o eliminar los atributos asociados al tipo. Para cada atributo se escoge un nombre y un tipo de dato asociado. La figura 3.8 muestra la estructura de esta ventana.

- **Pestaña de atributos de nodo**

La pantalla de especificación de nodos presenta una pestaña en la cual se listan los atributos configurados para ese tipo de nodo. Los atributos se presentan en una tabla de dos columnas. La columna del lado izquierdo muestra el nombre del atributo configurado mientras que la del lado derecho presenta un componente apropiado para el ingreso de datos del tipo que se configuró para el atributo. La figura 3.9 presenta un ejemplo de esta pantalla.

- **Pantallas de generación de documentación**

Las pantallas de generación de documentación presentan una barra de herramientas con opciones de visualización, impresión y guardado del documento visualizado. La figura 3.10 presenta un ejemplo de este tipo de pantallas.

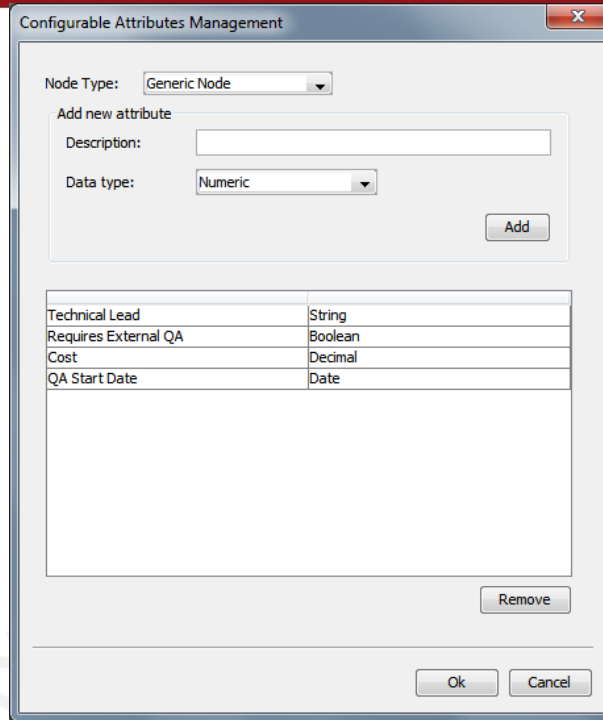


Figura 3.8: Pantalla de configuración de atributos de nodo [27]

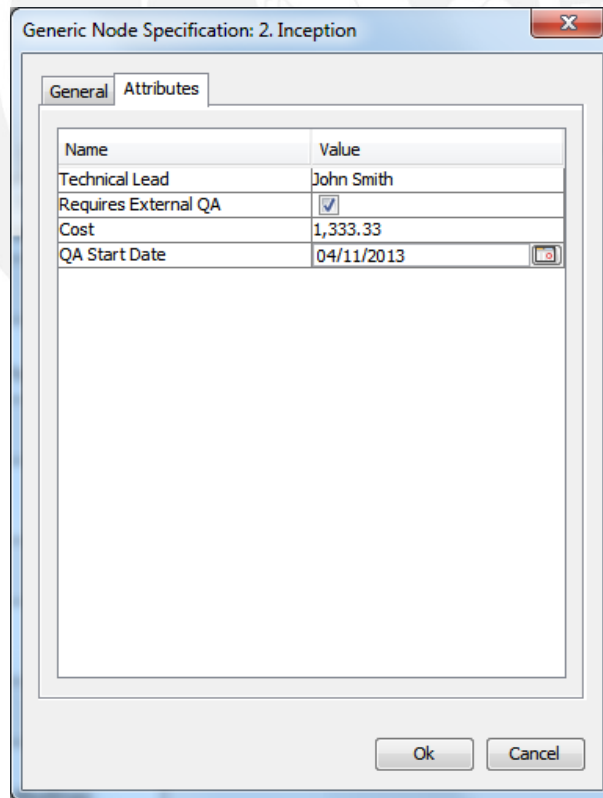


Figura 3.9: Pestaña de atributos de nodo [27]

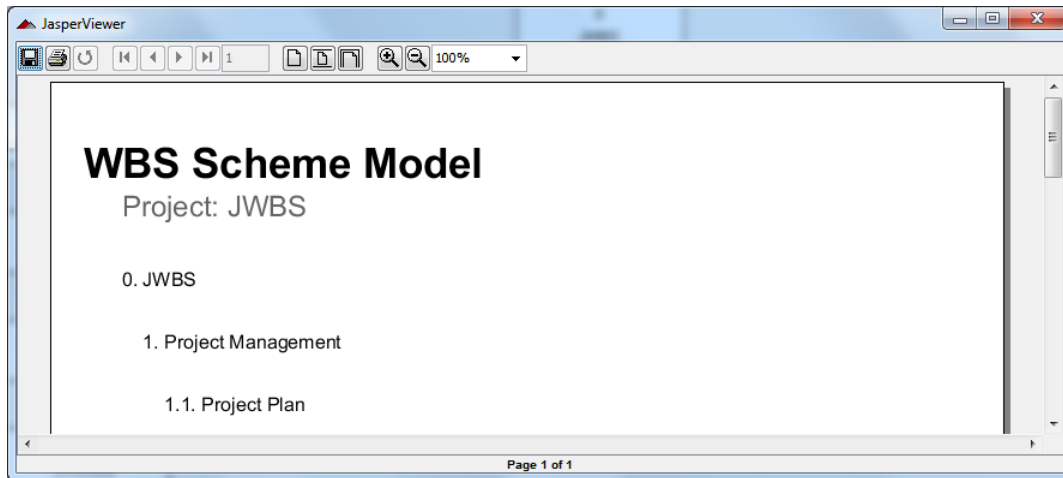


Figura 3.10: Pantalla de generación de documentación [53]



4. Construcción

En este capítulo se presentan las decisiones de construcción de la solución y luego se presenta la estrategia de pruebas seguidas para verificar la correcta implementación del producto.

4.1. Construcción

En esta sección se presentarán las tecnologías, patrones y *frameworks* utilizados para la construcción de la herramienta y se justificará su selección.

4.1.1. Lenguajes

La selección del lenguaje de programación debe asegurar el cumplimiento de todos los requisitos funcionales y no funcionales identificados, así como servir como base para la arquitectura de software definida. Es por este motivo que el lenguaje de programación elegido para el desarrollo de la herramienta es Java [10], siendo los principales argumentos para la decisión los siguientes:

- La característica multiplataforma de Java permite que la herramienta pueda ser ejecutada en múltiples sistemas operativos, ya que la JVM [11] (*Java Virtual*

Machine) puede correr sobre la mayoría de sistemas operativos populares de la actualidad incluyendo Windows, Linux, Macintosh, Solaris, entre otros.

- Al ser Java un lenguaje de programación libre, se cuentan con amplias fuentes de información y gran variedad de librerías desarrolladas por terceros, lo que es beneficioso para el desarrollo de este proyecto.
- La implementación de Sun de la plataforma Java es libre, por lo que el desarrollo de software sobre esta plataforma no tiene costo por licencias.
- Existen diversas librerías en Java que permiten la conectividad a diversas fuentes de datos. La mayoría de motores de base de datos pueden ser manejados en Java a través de *drivers* JDBC [12], así como archivos de texto y archivos XML.
- *Swing* [13] es una librería gráfica, desarrollada en Java e independiente de la plataforma que brinda un conjunto de componentes para crear interfaces de usuario. La herramienta a desarrollar requiere la interacción del usuario a través de una interfaz gráfica, por lo que el uso de estas librerías cubre esta necesidad.
- Se posee experiencia en el desarrollo de aplicaciones con Java.

Para la persistencia de los proyectos y el almacenamiento de los archivos de configuración que usará la herramienta se eligió XML [14] (por sus siglas en inglés *eXtensible Markup Language*) ya que, al ser un lenguaje de etiquetas cuya finalidad es la definición de otros lenguajes, permite crear una estructura personalizada que contenga toda la información de nuestro proyecto. Por cada proyecto, esta estructura puede ser almacenada en un archivo de texto independiente y utilizar XML es más sencillo y suficiente para el proyecto, descartando opciones menos ligeras como instalar y utilizar una base de datos relacional.

Para la definición de esta estructura se usó un esquema XSD [15] (por sus siglas en inglés *XML Schema Document*), en el cuál se detalla la organización, el contenido y la semántica de un archivo XML.

4.1.2. Entorno de programación

Como entorno de programación se eligió NetBeans [16], ya que es una herramienta libre y gratuita que permite el desarrollo de múltiples tipos de aplicaciones en Java, dando soporte además a otras tecnologías afines como XML. Es además uno de los pocos entornos de programación que posee un completo diseñador gráfico para

aplicaciones Java de escritorio, dando la posibilidad de crear una interfaz de usuario con componentes *Swing*.

4.1.3. Tecnologías, frameworks y patrones utilizados

- **Swing Application Framework (Framework)**

Swing Application Framework [17] es una librería que brinda clases que facilitan el desarrollo de aplicaciones de escritorio en Java, solucionando los problemas comunes que un desarrollador encuentra al usar una interfaz gráfica basada en *Swing*.

El *Swing Application Framework* da la base para una arquitectura y un conjunto de servicios comúnmente usados en aplicaciones *Swing*. El objetivo principal del *framework* es brindar un *kernel* para una típica aplicación *Swing*, ayudando a los desarrolladores a concentrarse en las funcionalidades de su aplicación a la vez que adoptan buenas prácticas para ciertos elementos comunes en el desarrollo con *Swing* tales como: ciclo de vida de la aplicación, manejo de recursos, manejo de eventos, concurrencia, manejo de estado de la sesión y almacenamiento local.

Las características del *framework* que se están considerando en el desarrollo de la herramienta son las siguientes:

- **Manejo del ciclo de vida de la aplicación:** Las aplicaciones que usan el *framework* poseen métodos definidos que pueden ser sobrescritos para indicar qué debe hacer la aplicación en su inicialización y cierre. En el caso de la herramienta, se usan diferentes métodos (según corresponda) para la creación, inicialización y visualización de la interfaz gráfica así como confirmaciones de salida de la aplicación y acciones que se deben realizar al ser cerrada.
- **Manejo, carga e inyección de recursos:** El *framework* permite separar recursos como cadenas, mensajes, imágenes, colores y fuentes en archivos un archivo aparte por cada formulario. Las propiedades de los componentes de la herramienta pueden ser configuradas en estos archivos y su modificación no implicaría modificar líneas de código. Estos archivos

pueden ser usados también para cambiar el idioma en el que se presentan las opciones de la herramienta.

- **Definición, manejo y asociación de acciones:** Se permite definir y encapsular acciones en métodos que pueden ser usados en diferentes eventos de componentes. De esta manera se permite reutilizar el código de una acción desde distintos componentes, asociando la misma acción a varios eventos. La acción también se puede personalizar en el archivo de recursos, asociándole un texto e iconos de diferente tamaño. Se le encuentra utilidad a esto al tener una acción que puede ser llamada desde un menú, una barra de herramientas o un botón.

Cabe resaltar también que desde la versión 6.0 del NetBeans se da soporte a este *framework*, integrándolo al diseño de aplicaciones Swing por medio de plugins que facilitan su uso.

- **XStream (Tecnología)**

XStream [18] es una librería para serializar y deserializar objetos hacia y desde XML. Su principal ventaja respecto a otras librerías es su facilidad de uso, ya que es mínima la codificación/configuración requerida para hacer conversiones simples. Por supuesto, la librería también puede soportar la manipulación de formatos XML más complicados.

Las principales características de la librería son las siguientes:

- **Facilidad de uso:** Brinda una interfaz de alto nivel que permite simplificar las conversiones comunes.
- **No mapeos:** La mayoría de objetos pueden ser serializados sin necesitar mapeos específicos.
- **Rendimiento:** La velocidad y una baja utilización de memoria son principios cruciales en el diseño de la librería, siendo esta adecuada para grafos complejos de objetos o en sistemas que intercambian mensajes largos.
- **XML limpio:** Ninguna información que puede ser obtenida por reflexión es duplicada. Esto resulta en archivos XML más fáciles de leer (legibles) y más compactos que la serialización nativa de Java.

- **No requiere modificaciones a objetos:** Todos los campos internos de un objeto son serializados, incluso los marcados como privados y constantes. Tanto las clases no públicas como las anidadas son soportadas. No es necesario agregar un constructor por defecto para que la serialización funcione.
- **Soporte completo para grafos de objetos:** Las referencias duplicadas encontradas en un modelo de objetos se mantienen. Las referencias circulares también son soportadas.
- **Estrategias de conversión personalizables:** Es posible crear y registrar nuevas estrategias de conversión, permitiendo personalizar la manera en la que clases particulares son serializadas.
- **Manejo de errores:** Se presenta un diagnóstico detallado en la ocurrencia de errores causados por un archivo XML malformado, a fin de identificar y aislar rápidamente el problema.

La utilización de esta librería permite reutilizar directamente (sin modificaciones) la estructura de objetos diseñada e implementada para almacenar la información de los proyectos. Al utilizar directamente la estructura ya implementada, la codificación necesaria para almacenar/abrir un proyecto se simplifica a unas cuantas líneas, sin necesidad de hacer configuraciones extensas. Así mismo, las estructuras de arreglos y las relaciones padre-hijo que manejan los modelos WBS no representan ninguna dificultad para la librería, gracias a su soporte completo para grafos de objetos.

- **Log4J (Tecnología)**

Log4J [19] es un paquete para Java permite hacer seguimiento al código ejecutado en la herramienta mediante la inserción de logs que pueden ser almacenados de forma persistente para su posterior revisión. De esta forma se pueden detectar rápidamente errores en la etapa de desarrollo. Incluso puede usarse como una herramienta de auditoría cuando la herramienta ya esté siendo usada, almacenando sucesos importantes y/o errores que puedan ocurrir.

El paquete Log4j tiene las siguientes características, según consta en la página oficial del producto:

- Es de uso libre, del proyecto Apache de Servicios Log. En este caso, es la entrega para su integración con lenguaje Java.
 - Es de fácil uso. Abarca pocas líneas de código.
 - El impacto en la performance de la aplicación que lo invoca es bastante bajo.
 - Tiene definida una jerarquía de niveles de mensajes: DEBUG, INFO, WARN, ERROR y FATAL. Los niveles permiten clasificar los mensajes a criterio del desarrollador y filtrar los niveles a ser mostrados.
 - El ambiente de auditoría puede ser modificado fácilmente a través de archivos de configuración. Esto incluye el formato de logs, el destino de salida (archivos de texto, consola, un servidor log4j remote, etc.) y los niveles de mensajes.
 - Maneja el concepto de herencia entre loggers.
 - Posee una comunidad amplia de usuarios y foros de soporte.
- **JasperReports (Tecnología)**

JasperReports [26] es una librería libre para la generación de reportes utilizando Java. Puede conectarse a cualquier origen de datos y producir documentos que pueden ser visualizados, impresos o exportados a diversos formatos como HTML, PDF, Microsoft Excel, OpenOffice y Microsoft Word.

La generación de documentación de la herramienta (diccionario de WBS, modelo de esquema, etc.) se implementó utilizando *JasperReports* a fin de aprovechar la ventaja que tiene para exportar a diferentes formatos.

- **Patrón Singleton (Patrón)**

Para las clases de la capa del controlador se usó el patrón *Singleton*, ya que, en su mayoría, las clases usadas en esta capa son gestores que sólo necesitan ser instanciados una sola vez por cada ejecución de la aplicación.

- **Patrón MVC (Modelo-Vista-Controlador) (Patrón)**

Se usó el patrón MVC para separar la capa de interfaz gráfica de la complejidad del modelo de datos. Tanto la separación como la comunicación

entre estas dos capas lógicas se realizan a través del Controlador, que es el encargado de recibir las peticiones que el usuario hace desde la vista y actualizar convenientemente el modelo y la vista de acuerdo a estas peticiones.

El resumen de tecnologías, patrones y *frameworks* utilizados por cada capa de la arquitectura se muestra en la Tabla 4.1.

Capa	Tecnologías/Patrones/Frameworks
Vista	<ul style="list-style-type: none"> ▪ Swing ▪ Swing Application Framework
Modelo	<ul style="list-style-type: none"> ▪ Clases Java ▪ XStream
Controlador	<ul style="list-style-type: none"> ▪ Clases Java ▪ Patrón Singleton
Seguridad	<ul style="list-style-type: none"> ▪ Log4J

Tabla 4.1: Resumen de tecnologías utilizadas por cada capa de la arquitectura [27]

4.1.4. Tópicos relevantes de la construcción

- **Presentación de datos en componentes gráficos complejos**

Una de las principales características de *Swing* es que la arquitectura de sus componentes está basada en una variación del patrón MVC llamada Arquitectura de Modelo Separable [13] que separa la responsabilidad de visualización y respuesta a la interacción del usuario (Vista y Controlador) de los datos mostrados y/o contenidos en el componente (Modelo). Para componentes simples como cajas de texto o botones este enfoque puede parecer demasiado complicado (y de hecho existen alternativas para no tener que manejar programáticamente el modelo de este tipo de componentes) sin embargo es bastante útil para componentes con estructuras de datos complejas como listas, árboles o tablas ya que es posible personalizar a detalle tanto la visualización del componente (incluso a nivel de cada elemento del modelo) como el comportamiento del mismo.

Cada clase que representa a un componente gráfico de *Swing* (*JComboBox*, *JTree*, *JTable*, etc.) está asociada a otra clase que representa al modelo

específico con las particularidades que requiere el componente. Por ejemplo, una clase que representa al modelo de una tabla (*JTable*) tendría que implementar la interface *TableModel* mientras que una clase que representa al modelo de una lista desplegable (*JComboBox*) implementaría la interface *ComboBoxModel*. Evidentemente, ambas clases tienen una implementación y comportamiento diferente, ligada al componente gráfico que complementan.

Esta arquitectura se aprovechó para la construcción de los principales componentes de la aplicación, como el árbol de navegación de versiones, el árbol de navegación de nodos y las tablas de configuración e ingreso de atributos.

- **Creación de copia en memoria de un modelo WBS**

Las funcionalidades de control integrado de cambios requieren que se cree un nuevo modelo WBS a partir de uno ya existente, tanto para la creación de una nueva solicitud de cambio (ya que permite hacer modificaciones al modelo WBS actual del proyecto) como en la aprobación de una solicitud (ya que el modelo WBS de la solicitud de cambio pasa a ser el modelo del proyecto).

Dado que la visualización, edición y persistencia de proyectos se basa en el estado en memoria de los objetos que los conforman, fue necesario implementar un mecanismo de copia de modelos WBS. Este mecanismo permite mantener varios modelos WBS asociados al proyecto, ya sean modelos activos de la versión actual del proyecto o como modelos asociados a una solicitud de cambio. También asegura que la modificación en un modelo sea independiente y no afecte a ninguno de los otros modelos del proyecto.

El objeto que representa a un modelo WBS pertenece a la clase *ModeloWBS* y está asociado a un objeto de la clase *Nodo*, que referencia al nodo raíz del modelo. Dada la estructura de árbol del modelo WBS, cada objeto *Nodo* tiene una composición de otros objetos *Nodo*, que representan a los hijos de cada elemento del WBS. A su vez, cada objeto *Nodo* también está asociado a su objeto *Nodo* padre. Es por este motivo que la copia en memoria del modelo requiere un recorrido recursivo que, además de copiar los atributos de cada nodo, también mantenga apropiadamente las referencias padre/hijos entre los nodos copiados.

La lógica de copia de un objeto de la clase *Nodo* se implementó en el método *clone*, que todos los objetos de Java heredan de la clase *Object* (y cuyo propósito es justamente duplicar un objeto). El copiado de un objeto *ModeloWBS* también se implementó en el método *clone*, recorriendo recursivamente todos los objetos *Nodo* partiendo desde el nodo raíz. Cada ejecución se encarga de copiar un objeto *Nodo*, asociar el objeto padre que le corresponde y repetir este mismo proceso para todos sus hijos.

El pseudocódigo del algoritmo implementado se presenta la Figura 4.1.

```

Procedimiento copiarNodos ( nodoACopiar , nodoCopia )

  Para Cada hijoACopiar En nodoACopiar.hijos ( ) Hacer

    //Copiar cada uno de los hijos del nodo actual y
    //asociarlo con su padre
    hijoCopia ← hijoACopiar.clonar ( )
    hijoCopia.Padre ← nodoCopia
    nodoCopia.AgregarHijo ( hijoACopiar , hijoCopia )

    //Llamada recursiva para cada uno de los hijos
    copiarNodos ( hijoACopiar , hijoCopia )

  Fin Para Cada
Fin Procedimiento
  
```

Figura 4.1: Pseudocódigo para el algoritmo de copia de un modelo WBS [27]

La llamada inicial del algoritmo requiere tanto el nodo raíz del modelo como la copia previamente generada del mismo. La copia es requerida para asociar el padre de cada uno de los nodos hijos a copiar.

4.2. Pruebas

En esta sección se presenta y detalla la estrategia de pruebas y los tipos de prueba usados en la construcción de la herramienta. Se presenta también el formato que se utilizó para el reporte de pruebas y se describe la manera en la que ha sido utilizado.

4.2.1. Estrategia de pruebas

La ejecución de las pruebas debe realizarse al finalizar cada iteración de la etapa de construcción. Para el desarrollo de la herramienta, se está tomando una construcción incremental con 2 iteraciones. Al finalizar cada iteración se probarán los casos de uso implementados. Los casos de uso que se implementarán y el orden en que serán probados por cada iteración se muestran a continuación:

- **Primera iteración:**

- 1) Mantener Proyecto
- 2) Explorar Versiones del Proyecto
- 3) Mantener Atributos de Nodos
- 4) Mantener Nodos Genéricos
- 5) Mantener Entregables
- 6) Mantener Paquetes de Trabajo
- 7) Explorar Nodos del WBS
- 8) Actualizar Estado de Entregables
- 9) Mantener Solicitudes de Cambio
- 10) Actualizar Estado de Solicitud de Cambio

- **Segunda iteración:**

- 1) Almacenar Proyecto
- 2) Cargar Proyecto
- 3) Visualizar Modelo del WBS
- 4) Visualizar Diseño de Esquema del WBS
- 5) Exportar Gráfico del WBS
- 6) Exportar Diseño de Esquema del WBS
- 7) Visualizar Diccionario del WBS
- 8) Exportar Diccionario del WBS
- 9) Generar Cronograma del Proyecto
- 10) Visualizar Vista de Estado Entregables
- 11) Visualizar Reporte de Estado de Entregables
- 12) Visualizar Solicitud de Cambio

Como se puede apreciar, en la primera iteración se implementan todas las funcionalidades de construcción del modelo WBS mientras que en la segunda se

implementan funcionalidades que explotan la información que se genera al construir el modelo WBS (almacenar y cargar proyectos y visualizar y exportar reportes). Se separó de esta manera ya que para probar el almacenamiento del proyecto y la visualización de reportes y documentos se requiere que existan modelos WBS construidos. La integración de los desarrollos de la segunda iteración no debe impactar significativamente en lo desarrollado en la primera debido a que las funcionalidades de la segunda iteración sólo explotan la información que se crea con las funcionalidades de la primera.

4.2.2. Tipos de prueba

Se usarán dos tipos de pruebas:

- **Pruebas de casos de uso:** Se probará de forma aislada cada caso de uso y se verificará la correcta implementación de los flujos básicos y alternativos de todos los casos de uso a probar en cada iteración. Las pruebas de cada caso de uso pueden abarcar una o varias pantallas.
- **Prueba de integración:** Se probará la integración de todas las funcionalidades de la herramienta para asegurar su correcto funcionamiento. Consistirá en una prueba que abarca el modelado de un modelo WBS con todos sus nodos, la creación de múltiples solicitudes de cambio del alcance para la primera versión y la aprobación de la solicitud de cambio, generando una nueva versión. Una vez realizado esto se probarán los documentos y reportes que pueden ser generados en la herramienta. Esta prueba se realizará de manera implícita al realizar las pruebas de casos de uso.

4.2.3. Catálogo de pruebas

A continuación se mostrarán los principales casos de prueba que serán utilizados para verificar que la herramienta desarrollada cumpla con los requisitos identificados. Los casos de prueba se orientarán a la prueba de cada caso de uso y se clasificarán de la siguiente manera:

- Pruebas de las capas de arquitectura: Se probarán las capas lógicas Modelo y Vista. No se realizarán pruebas del Controlador ya que se está verificando que funcione correctamente al probar la Vista y el Modelo.

- Pruebas de la capa de datos: Se probará la funcionalidad de almacenar y cargar proyectos en un archivo XML.
- Pruebas de generación de documentación: Para probar la exportación a archivos de los modelos WBS, los reportes y la documentación del proyecto.

El catálogo completo se encuentra en el Anexo D: Catálogo de Pruebas. La numeración que se muestra de las pruebas corresponde a la numeración que tienen en el Anexo D.

Pruebas del Modelo

Caso de Uso: Mantener Proyectos

Prueba 1	
Objetivo Prueba:	Se probará el flujo básico del caso de uso.
Precondición:	Ninguna
Descripción de la prueba:	Ingresa los datos completos de un proyecto. Ingresa al menú "File" y selecciona la opción "New Project". Sigue el Wizard con la opción "A new project" y coloca los datos especificados para la prueba.
Datos de prueba:	Proyecto Ejemplo: <ul style="list-style-type: none"> • <i>Name</i>: Ejemplo JWBS • <i>Description</i>: Este es un proyecto prueba • <i>Project Manager</i>: Germán Arellano • <i>Currency</i>: S/. • <i>Start Date</i>: 01/01/2012 • <i>End Date</i>: 31/05/2012
Resultados Esperados:	Se crea un nuevo proyecto con la versión inicial del alcance. Verifica que los datos ingresados coincidan con los que se visualizan en la opción "Project Properties" del menú "File".

Caso de Uso: Mantener Nodos Genéricos

Prueba 5	
Objetivo Prueba:	Se probará el flujo básico del caso de uso.
Precondición:	Modelo WBS seleccionado, Nodo raíz seleccionado.
Descripción de la prueba:	Ingresar los datos completos de un nodo genérico. Seleccionar el nodo raíz del Proyecto Ejemplo cuyo nombre es "0. Ejemplo JWBS". Seleccionar la opción " <i>Create new Generic Node</i> ".
Datos de prueba:	Nodo Genérico 1: <ul style="list-style-type: none"> • <i>Name</i>: Nodo Genérico 1 • <i>Type</i>: Subproject
Resultados Esperados:	Se crea un nuevo nodo genérico, hijo del nodo seleccionado. Seleccionar el nodo creado e ingresar a la opción " <i>Edit Node specification</i> " para verificar que los datos que se visualizan coincidan con los ingresados y que los campos " <i>Code</i> " y " <i>Level</i> " tengan el valor de "1".

Prueba 6	
Objetivo Prueba:	Se probará el flujo alternativo "Modificar un Nodo Genérico"
Precondición:	Modelo WBS seleccionado, Nodo Genérico seleccionado.
Descripción de la prueba:	Modificar los datos de un nodo genérico ya existente. Seleccionar el nodo "Nodo Genérico 1" y seleccionar la opción " <i>Edit Node specification</i> ". Cambiar los valores con los datos especificados para la prueba.
Datos de prueba:	Nodo Genérico 1: <ul style="list-style-type: none"> • <i>Name</i>: Nodo Genérico 1 modificado • <i>Node Type</i>: Phase • <i>Description</i>: He agregado una descripción
Resultados Esperados:	Los datos del nodo genérico se encuentran actualizados. Verificar que el nombre del nodo se haya actualizado en el árbol. Reingresar a la opción " <i>Edit Node specification</i> " y verificar que los datos se encuentren actualizados.

Caso de Uso: Actualizar Estado de Solicitud de Cambio

Prueba 15	
Objetivo Prueba:	Se probará el flujo básico del caso de uso
Precondición:	Proyecto abierto. Solicitud de cambio seleccionada.
Descripción de la prueba:	Actualizar el estado de una solicitud de cambio del alcance. Seleccionar la solicitud de cambio "SC-0001" del proyecto "Ejemplo JWBS con solicitud cambio". Seleccionar la opción "Edit/View a Scope Change Request". Abrir la pestaña "Status" y cambiar el estado a "Approved".
Datos de prueba:	<p>SC-0001:</p> <ul style="list-style-type: none"> • <i>Approved by:</i> Germán Arellano • <i>Approval Date:</i> 10/03/2012 • <i>Comments:</i> Este es un comentario en solicitud de cambio <p>Nueva versión:</p> <ul style="list-style-type: none"> • <i>Name:</i> Primera Versión • <i>Code:</i> 1
Resultados Esperados:	La solicitud de cambio se encuentra aprobada. Verificar que se haya generado una nueva versión del alcance del proyecto con el nombre y código que se ingresó. Verificar también que al reingresar a la solicitud de cambio los datos se encuentren actualizados.

Pruebas de la Vista

Caso de Uso: Visualizar Modelo WBS

Prueba 18	
Objetivo Prueba:	Se probará el flujo básico del caso de uso.
Precondición:	Modelo WBS seleccionado.
Descripción de la prueba:	Visualizar el modelo WBS en el área de trabajo de la pantalla.
Resultados Esperados:	Se debe mostrar el gráfico del modelo WBS seleccionado en la interfaz de usuario.

Pruebas de la Capa de Datos

Caso de Uso: Cargar Proyecto

Prueba 16	
Objetivo Prueba:	Se probará el flujo básico del caso de uso.
Precondición:	Ninguna
Descripción de la prueba:	Se escoge un proyecto de una ruta del disco duro para su carga en la herramienta. Abrir el proyecto "Ejemplo JWBS con solicitud cambio".
Resultados Esperados:	Se deberían cargar todos los datos del proyecto "Ejemplo JWBS con solicitud cambio" y ser mostrados en la interfaz de usuario.

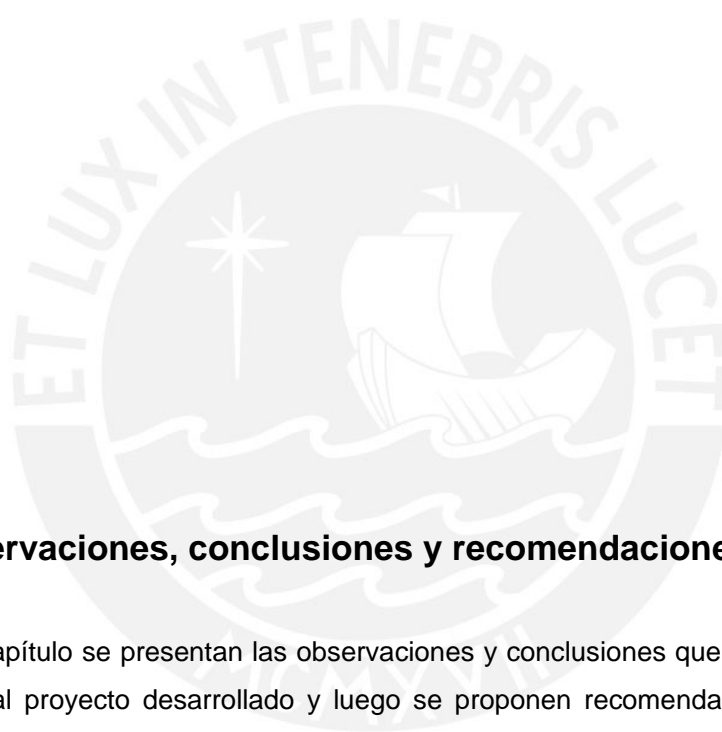
4.2.4. Reporte de pruebas

Al ejecutar cada una de las pruebas, se registrará el resultado de las mismas en un reporte de pruebas. El formato del reporte de pruebas se muestra en la Figura 4.2. La descripción de cada campo del reporte se detalla a continuación:

- Orden: Indica la secuencia en la que se realizarán las pruebas.
- Nro. de Prueba: Corresponde al número que tiene asignado la prueba en el Catálogo de Pruebas.
- Caso de Uso: Debe indicar el nombre del caso de uso que se está probando.
- Prueba: Corresponde al nombre de la prueba en el Catálogo de Pruebas.
- Descripción: Puede ser usado para describir brevemente en qué consiste la prueba y para indicar los pasos a seguir.
- Anexos: Contiene un hipervínculo a otra hoja del reporte en la que se pueden incluir a detalle los pasos y capturas de pantalla útiles para el desarrollo de la prueba.
- Resultado Esperado: Se incluye lo que debería ocurrir una vez ejecutada la prueba.
- Campos de Resultados: Se incluyen los resultados de la prueba. En estos se indica el estado de la prueba y las observaciones que se hayan podido encontrar.

Casos de Prueba						Campos de Resultados			
Orden	Nro. De Prueba	Caso de Uso	Prueba	Descripción de la prueba	Anexos Obligatorios	Resultado Esperado	Aceptado	Rechazado	Observaciones
1	1	Mantener Proyectos	Se probará el flujo principal Crear Proyecto.	Crear un nuevo proyecto en la herramienta.	Anexo 1	Se debería mostrar un formulario con los datos del proyecto. Al ingresar los datos, se debería crear un nuevo proyecto con la versión inicial del alcance.	X		
2	2	Almacenar Proyecto	Se probará el flujo principal del caso de uso.	Almacenar un proyecto modelado en la herramienta.	Anexo 2	Se debería permitir al usuario elegir la ruta en la que desea guardar el proyecto. Una vez que escoge la ruta y se confirma el guardado el proyecto debería estar grabado en el disco.		X	No grabó correctamente el proyecto.

Figura 4.2: Ejemplo de reporte de pruebas



5. Observaciones, conclusiones y recomendaciones

En este capítulo se presentan las observaciones y conclusiones que se pueden hacer respecto al proyecto desarrollado y luego se proponen recomendaciones y trabajos futuros relacionados con el proyecto.

5.1. Observaciones

- La mayoría de las herramientas basadas en WBS que existen actualmente se enfocan sólo en la creación del diagrama WBS y no explotan al máximo la capacidad que este tiene para la gestión del alcance del proyecto. Esto motivó a que el producto elaborado busque llenar dicho vacío.
- El uso de Java como lenguaje de programación permitió encontrar gran cantidad de documentación, componentes y tecnologías que han sido usadas para acelerar el proceso de desarrollo de la herramienta.

- La experiencia trabajando con Java y la gran cantidad de tecnologías y componentes reutilizables permitieron acelerar el desarrollo de la herramienta.
- El uso de XStream simplificó el almacenamiento y carga de los proyectos en archivos XML, delegando toda la serialización de los proyectos modelados utilizando sólo unas pocas líneas de código para su configuración. De cara a la mantenibilidad de la herramienta, la modificación de campos en alguna de las entidades del metamodelo WBS no requiere de ningún cambio en el código fuente para su correcto funcionamiento.

5.2. Conclusiones

- Se consiguió la implementación de una herramienta para la gestión de proyectos usando tecnologías y lenguajes de programación libres, aún si los requisitos incluyen funcionalidades complejas y de integración con otras herramientas y/o formatos. La gran cantidad de proyectos libres y la numerosa comunidad que rodea a tecnologías de desarrollo abierto como Java permite encontrar documentación, librerías y guías de buenas prácticas (patrones de diseño, estándares) que permiten que un proyecto de desarrollo de software se complete con éxito.
- La revisión del estado del arte de herramientas para la gestión del alcance de proyectos basadas en WBS y la revisión de las técnicas generalmente reconocidas como buenas prácticas en gestión del alcance permitieron la definición, especificación y priorización de requerimientos para la herramienta implementada en este proyecto de fin de carrera.
- Se consiguió diseñar un metamodelo para la representación de modelos WBS, basándose las relaciones que tienen los diferentes tipos de nodo que conforman un modelo.
- La utilización de una arquitectura basada en componentes permitió la obtención de un modelo de datos reutilizable en una librería independiente (JAR) que puede servir para futuros proyectos que utilicen conceptos de WBS.

5.3. Recomendaciones y trabajos futuros

- Se pueden realizar mejoras a la herramienta como las siguientes:
 - Adaptación de la herramienta para configurar recursos y roles para un proyecto: De esta manera se podría tener centralizados a todos los

- recursos de un proyecto y podrían ser referenciados desde campos que requieran la asignación de un recurso. Se podría desarrollar también módulos que permitan la integración de la herramienta con servidores de correo (por ejemplo *Microsoft Exchange* o *IBM Lotus Domino*) desde las cuales se puede obtener información de los recursos.
- Adaptación de la herramienta para gestionar el tiempo del proyecto: Se podría agregar elementos de la gestión del tiempo (por ejemplo configuración del calendario laborable, personalización del calendario de un recurso, configuración del período laboral, etc.) para permitir planificar las actividades de un proyecto, abriendo el camino para que la herramienta se integre efectivamente con otras herramientas especializadas en la gestión del tiempo o incluso permitir que desde la misma herramienta se gestione el tiempo del proyecto.
 - Reutilización del modelo de la herramienta (paquete “modelo”) para la elaboración de un sistema web de gestión del alcance de proyectos: El análisis realizado para la identificación de las clases del modelo de la herramienta puede ser utilizado para la construcción de un sistema web de gestión del alcance de proyectos que puede incluir funcionalidades como la consulta de la información del proyecto y los modelos WBS a múltiples usuarios, la definición de roles y privilegios para los usuarios brindando diferentes funcionalidades para cada rol, definición de un flujo de aprobación para las solicitudes de cambio, etc.

Bibliografía

- [1] Project Management Institute. “*Guía de los fundamentos de la dirección de proyectos (Guía del PMBOK)*” Tercera edición
- [2] Project Management Institute. “*Project Management Institute Practice Standard for Work Breakdown Structures*”, Second Edition (2006)
- [3] Work Breakdown Structure - http://www.hyperhot.com/pm_wbs.htm
Fecha de consulta: 13/08/2012
- [4] Félix Valdéz Torero, PMP – *Creando una buena Estructura de Descomposición del Trabajo (WBS)*
- [5] WBS Chart Pro Software
<http://www.criticaltools.com/wbschartprosoftware.htm>
Fecha de consulta: 13/08/2012
- [6] Visio 2010 Add-in for WBS Modeler
http://visio.microsoft.com/en-us/Samples_and_Downloads/Software_Add-ins/Pages/Visio_2010_Add-in_for_WBS_Modeler.aspx
Fecha de consulta: 05/11/2012
- [7] Malmö University: *Rational Unified Process - Artifacts*
http://www.ts.mah.se/RUP/RationalUnifiedProcess/process/artifact/ovu_arts.htm
Fecha de consulta: 13/08/2012
- [8] MVC XEROX PARC 1978-79 -
<http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
Fecha de consulta: 13/08/2012
- [9] How to use Model-View-Controller -
<http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>
Fecha de consulta: 19/09/2012
- [10] Java Platform Standard Edition 6 Overview
<http://docs.oracle.com/javase/6/docs/technotes/guides/index.html>
Fecha de consulta: 19/09/2012
- [11] Tim Lindholm, Frank Yellin – *The Java Virtual Machine Specification* Segunda Edición
<http://docs.oracle.com/javase/specs/jvms/se5.0/html/VMSpecTOC.doc.html>
Fecha de consulta: 19/09/2012

- [12] Java SE Technologies – Database
<http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136101.html>
Fecha de consulta: 19/09/2012
- [13] A Swing Architecture Overview
<http://www.oracle.com/technetwork/java/architecture-142923.html>
Fecha de consulta: 19/09/2012
- [14] Extensible Markup Language (XML) 1.0 Fourth Edition
<http://www.w3.org/TR/2006/REC-xml-20060816/>
Fecha de consulta: 01/11/2012
- [15] W3C XML Schema
<http://www.w3.org/XML/Schema>
Fecha de consulta: 01/11/2012
- [16] NetBeans IDE - Features
<http://www.netbeans.org/features/index.html>
Fecha de consulta: 01/11/2012
- [17] Using the Swing Application Framework (JSR 296)
<http://www.oracle.com/technetwork/articles/javase/swingappfr-136951.html>
Fecha de consulta: 01/11/2012
- [18] XStream
<http://xstream.codehaus.org/>
Fecha de consulta: 12/03/2012
- [19] Apache Log4J 1.4.2
<http://logging.apache.org/log4j/1.2/index.html>
Fecha de consulta: 01/11/2012
- [20] Jeremías Quintanilla R. “*Tesis para optar por el título de Ingeniero Informático: Herramienta CASE para la estimación y planificación de proyectos de software con casos de uso*”, Pontificia Universidad Católica del Perú, Lima, 2006
- [21] IBM – Rational Unified Process
<http://www-01.ibm.com/software/awdtools/rup/>
Fecha de consulta: 01/11/2012
- [22] Flores, L. “*Tesis para optar por el título de Ingeniero Informático: Herramienta CASE para el Desarrollo de proyectos de Ingeniería de Software*”. Pontificia Universidad Católica del Perú, Lima, 2003.
- [23] Ínjoque Vicente, A. / Wong Leung V. / Zelada Ramos N. “*Tesis para optar por el título de Ingeniero Informático: Herramienta para realizar la gestión de proyectos de software*”. Pontificia Universidad Católica del Perú, Lima, 2006.

- [24] Project Scope Management – Darren Wich
http://www.umsl.edu/~sauterv/analysis/6840_f09_papers/Wich/scopemanagement.html
Fecha de consulta: 18/03/2012
- [25] The importance of a Work Breakdown Structure
<http://smallbusiness.chron.com/importance-work-breakdown-structure-54294.html>
Fecha de consulta: 25/03/2013
- [26] JasperReports
<http://community.jaspersoft.com/project/jasperreports-library>
Fecha de consulta: 17/05/2013
- [27] Creación propia

