



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

DISEÑO E IMPLEMENTACIÓN DE HERRAMIENTA PARA LA CONFIGURACIÓN
DE CONTEXTOS EN PROYECTOS DE SOFTWARE

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN COMPUTACIÓN

DANIEL ANTONIO ORTEGA NORAMBUENA

PROFESOR GUÍA:
MARÍA CECILIA BASTARRICA PIÑEYRO

MIEMBROS DE LA COMISIÓN:
JOSÉ A. PINO URTUBIA.
PABLO GONZÁLEZ JURE.

SANTIAGO DE CHILE
AGOSTO 2012

Resumen

Pocas PyMEs de desarrollo de software en Chile tienen un proceso organizacional definido, aunque son concientes de su importancia.

El proyecto ADAPTE ¹ tiene como objetivo adaptar el proceso de desarrollo organizacional, generando un proceso acorde a un proyecto de software específico. Para poder llevar a cabo esta tarea, es necesario tener conocimiento del proceso organizacional y del contexto de cada empresa. Actualmente, la definición del proceso organizacional se realiza con una herramienta llamada *Eclipse Process Framework Composer (EPF)*, y para la definición del contexto se utiliza la herramienta *Eclipse Modeling Tools (EMT)*. Una vez obtenidas ambas definiciones, se realizan las transformaciones necesarias para obtener un proceso organizacional adaptado.

El problema de este proceso, se presenta en el uso de la herramienta *EMT*, ya que es una herramienta específica, la cual requiere de un conocimiento previo para operar en ella. El no tener este conocimiento puede generar costos tanto en tiempo, esfuerzo y dinero para la empresa.

Es necesario que esta definición se realice en dos etapas distintas. La primera etapa es la definición del modelo de contexto organizacional, la cual se debe realizar por un Ingeniero de Procesos de la organización. La segunda etapa consta de configurar el modelo de contexto definido en la primera etapa, obteniendo un contexto concreto en un proyecto de software específico. Esta labor la debe llevar a cabo un Jefe de Proyecto. Ambas etapas se deben realizar de manera fácil, rápida e intuitiva, sin necesidad de tener conocimiento de alguna herramienta específica como *EMT*.

Un modelo de contexto está construido por tres elementos fundamentales: las dimensiones, los atributos y los posibles valores. Estos elementos, se relacionan de la siguiente manera: las dimensiones poseen atributos y los atributos tienen posibles valores.

La solución propuesta es construir una aplicación web, la cual consta de dos módulos. El primero, destinado para la definición del modelo de contexto, y el segundo para la configuración del contexto concreto. Para el manejo y desarrollo de las interfaces, en una aplicación web existe mayor libertad por parte del desarrollador para lograr un alto nivel de usabilidad. Además, las personas en general están muy familiarizadas con este tipo de interfaces, ya que conviven con estas día a día.

Para validar la solución se realizaron pruebas de usabilidad y robustez. Las pruebas de usabilidad se enfocaron en validar el nivel de satisfacción que genera la aplicación en el usuario y el nivel de aprendizaje de la interfaz. Se utilizó una entrevista con preguntas cuantitativas y cualitativas, permitiendo tener una calificación de la aplicación y mayor retroalimentación por parte del usuario.

Para las pruebas de robustez, se utilizó la aplicación en situaciones en las cuales, de no responder de manera correcta, el resultado obtenido no sería el esperado por el usuario. Estas situaciones fueron: tratar de eliminar todas las dimensiones del modelo de contexto, todos los atributos de una dimensión, y todos los posibles valores de un atributo. También se intentó crear elementos repetidos, ya sean dimensiones, atributos de una misma dimensión o posibles valores de un mismo atributo.

¹Sigla de Adaptable Domain and Process Technology Engineering. Proyecto Fondef cuyo objetivo es optimizar y apoyar el mejoramiento de procesos en PyMEs de tecnologías de la información y la comunicación

*Este trabajo de título está dedicado a mis padres quienes siempre
confiaron en mi y me apoyaron incondicionalmente.
También se lo dedico a mi novia quien fue el pilar fundamental
durante todos mis años de estudio universitario.*

Agradecimientos

Quiero agradecer a los profesores Cecilia Bastarrica y Sergio Ochoa, quienes supieron apoyarme y aconsejarme durante el período en el que desarrollé mi trabajo de título. Además, quiero agradecer a mi familia y a mi novia quienes supieron entender y respetar el poco tiempo que les pude dedicar mientras estaba trabajando en mi memoria. También quiero agradecer a mi amigo Josué Pino, quien me ayudó y asesoró con la redacción y ortografía del presente trabajo de título.

Índice

1. Introducción	1
1.1. Objetivos	4
1.2. Relevancia/Motivación	4
1.3. Alternativas Analizadas	5
1.3.1. Aplicación de Escritorio	5
1.3.2. Aplicación Web	6
1.4. Descripción general de la solución	6
1.4.1. Definición del modelo de contexto (módulo Modelo de Contexto)	6
1.4.2. Configuración del contexto concreto (módulo Contexto Concreto)	7
1.5. Resultados de la solución implementada para resolver el problema	7
2. Antecedentes	8
2.1. MVC	8
2.2. Framework	8
2.3. Eclipse Modeling Framework	8
2.4. Usabilidad	10
2.5. Robustez	10
2.6. Contexto	10
2.7. Escala de Likert	11
2.8. jQuery	11
3. Especificación del problema	12
3.1. Problema	12
3.2. Requisitos Específicos	13
3.3. Casos de Uso	14
4. Descripción de la solución	16
4.1. Especificación de contextos	16
4.2. Software Project Context Modeling Language (SPCML)	16
4.3. Arquitectura de software	18
4.4. Arquitectura de hardware	18
4.5. Diseño de archivos que almacenan la información procesada	20
4.6. Diseño de clases	21
4.6.1. Package Controller	22
4.6.2. Package Domain	23
4.6.3. Package Utils	24
4.6.4. Formularios Dinámicos en las Interfaces	28
4.7. Cómo la solución planteada resuelve el problema	29

5. Funcionamiento de la aplicación paso a paso	32
5.1. Modelo de Contexto	32
5.1.1. Inicio	32
5.1.2. Definición del modelo de contexto	32
5.1.3. Creación de links de prioridad entre atributos	34
5.1.4. Creación exitosa y descarga de los resultados	37
5.2. Contexto Concreto	38
5.2.1. Inicio	38
5.2.2. Configuración del contexto concreto	39
5.2.3. Creación exitosa y descarga de los resultados	39
6. Validación de la solución	42
6.1. Usabilidad en la Aplicación	42
6.1.1. Muestra	42
6.1.2. Instrumentos	42
6.1.3. Tareas	43
6.1.4. Resultados Entrevista	43
6.1.5. Resultados Cuestionario del Usuario Final	44
6.1.6. Análisis de los resultados obtenidos	52
6.2. Robustez en la Aplicación	54
6.2.1. Diálogos de confirmación al momento de eliminar	54
6.2.2. Diálogo de confirmación al momento de crear el modelo de contexto	54
6.2.3. Diálogo informativo que no permite la eliminación de todas las dimensiones, atributos o posibles valores	56
6.2.4. Diálogo informativo que no permite tener valores repetidos en las dimensiones, atributos o posibles valores	56
6.3. Contexto Concreto Generado (XMI)	57
6.3.1. Demostrando el correcto formato del archivo XMI mediante la herramienta EMF	57
7. Análisis y Discusión	61
8. Conclusión	62
8.1. Trabajo Futuro	63
9. Referencias	64
A. Anexo I: Archivo XML. Resultado de la definición de un modelo de contexto en el modulo Modelo de Contexto	66

B. Anexo II: Archivo XMI. Resultado de la configuración de un contexto concreto en el modulo Contexto Concreto	66
C. Anexo III: Pauta de entrevista	68
D. Anexo IV: Cuestionario usuario final	69

Índice de figuras

1.	Modelo de como ADAPTE pretende abordar el problema	3
2.	Representación jerárquica de un contexto en formato XMI. Este archivo XMI es el resultado de un contexto configurado con la herramienta implementada en el presente trabajo.	9
3.	Etapas para realizar el proceso de adaptación con ADAPTE.	12
4.	Casos de uso para el módulo Modelo de Contextos	14
5.	Casos de uso para el módulo Contexto Concreto	15
6.	Diagrama Metamodelo SPCML	17
7.	Diagrama Modelo de Contexto	18
8.	Funcionamiento del patrón MVC	19
9.	Arquitectura Cliente/Servidor	19
10.	Diagrama de clases del package Domain	24
11.	Diagrama que ilustra la interacción de todas las vistas, controladores y entidades de la aplicación.	30
12.	Diagrama del flujo que sigue la aplicación para crear el contexto concreto. Se comienza con la definición del modelo de contexto organizacional por parte de un Ingeniero de Procesos. Una vez definido, un Jefe de Proyecto configura el modelo de contexto para obtener finalmente el contexto concreto para un proyecto de software específico. . . .	31
13.	Interfaz inicio módulo Modelo de Contexto	32
14.	Interfaz panel modificación datos modelo de contexto módulo Modelo de Contexto .	33
15.	Panel administrador de dimensiones módulo Modelo de Contexto	34
16.	Panel administrador de todos los valores de la dimensión seleccionada módulo Modelo de Contexto	35
17.	Interfaz definición modelo de contexto módulo Modelo de Contexto	36
18.	Ejemplo de cómo opera la grilla	36
19.	Grilla con checkboxes seleccionados	37
20.	Interfaz creación de links entre atributos módulo Modelo de Contexto	37
21.	Interfaz final módulo Modelo de Contexto	38
22.	Interfaz inicio módulo Contexto Concreto	39
23.	Interfaz configuración contexto concreto módulo Contexto Concreto	40
24.	Interfaz final módulo Contexto Concreto	41
25.	Resultados entrevista	44
26.	Respuestas a todas las preguntas del cuestionario usuario final	52
27.	Mensajes de confirmación antes de eliminar una Dimensión, Atributo o un Posible Valor	55
28.	Mensajes de confirmación antes de enviar los formularios al servidor	56

29.	Mensajes informativos, indicando que deben existir al menos una Dimensión, un Atributo y un Posible Valor	58
30.	Mensajes informativos, indicando al usuario la existencia de elementos con el mismo nombre.	59
31.	Demostración de la correcta generación de los archivos XMI en el módulo Contexto Concreto	60
A1.	Archivo XML que representa un modelo de contexto	66
B1.	Archivo XMI que representa un contexto concreto	67

1. Introducción

Contexto

Un proceso de desarrollo de software permite la construcción de productos de calidad, en forma predecible y planificable, tanto en términos de tiempo como de recursos. Las organizaciones que se dedican a la construcción de software, definen procesos de desarrollo organizacionales con el fin de mejorar su productividad, desempeño y así aumentar su competitividad.

La mayor parte de las organizaciones de desarrollo de software en Chile son PyMEs, y muy pocas de ellas tiene un proceso de desarrollo organizacional definido, aunque muchas son conscientes de su relevancia [GECHS et. al.,2008] [10].

Estas empresas en general desarrollan proyectos cortos y de mediana complejidad, cuyo objetivo es esencialmente la construcción del software en sí, reduciendo al máximo posible las actividades relativas a la gestión del proyecto.

Un problema de tecnología de información y computación

Debido a que las organizaciones tienen proyectos de distinta naturaleza, un único proceso organizacional normalmente no se aplica de la misma forma a cada proyecto [Killisperger et. al., 2009] [13].

Existen varios factores de los cuales depende la naturaleza de un proyecto; entre los que se pueden reconocer son el tamaño del equipo, la calificación del personal, el conocimiento del dominio y el riesgo asociado, entre otros [Boehm et. al., 1995] [5].

Es por esto que un proceso organizacional debe ser continuamente adaptado a las necesidades específicas de cada proyecto. Esta adaptación es un proceso que debe hacerse al comenzar cada proyecto de desarrollo de software.

Dado que el proceso adaptado es la base para la planificación, la adaptación se debe hacer de forma rápida y eficiente.

Actualmente, existen estrategias de adaptación del proceso organizacional a un proceso específico del proyecto, entre las cuales se encuentran:

1. La aplicación del mismo proceso organizacional estrictamente como fue definido, siempre.
2. Un repositorio de método y un configurador de proceso [Brinkkemper et. al., 1998; Mirbel et. al., 2005; Aharoni and Reinhartz-Berger, 2008] [6].
3. Un conjunto de procesos de referencia para varios tipos de proyecto identificados en la organización, similar al conjunto de procesos plantilla sugeridos por Crystal Methodology [Cockburn, 2002] [4].
4. La aplicación de las dos últimas estrategias en forma combinada.

ADAPTE

ADAPTE [1] son las siglas de *Adaptable Domain and Process Technology Engineering*, el cual es un proyecto Fondef cuyo objetivo es optimizar y apoyar el mejoramiento de procesos en pequeñas y medianas empresas (PyMEs) de tecnologías de la información y la comunicación (TICs).

ADAPTE es un proyecto que está llevando a cabo un equipo de investigadores provenientes de tres universidades chilenas, con vasta experiencia profesional y académica en computación e informática. Las Universidades que participan en el proyecto son:

- Universidad Técnica Federico Santa María
- Universidad de Chile
- Pontificia Universidad Católica de Valparaíso

En conjunto con este proyecto se encuentran también trabajando seis PyMEs nacionales, las cuales se mencionan a continuación:

- AMISOFT
- DTS
- KI TEKNOLOGY
- TRIGENSOFT
- RHISCOM
- IMAGEN

Solución propuesta por ADAPTE [1]

El proyecto plantea una nueva estrategia de adaptación de procesos que implica trasladar parte del esfuerzo de la adaptación a la definición del proceso organizacional. Para ello se propone un enfoque planificado de reuso de activos en la definición del proceso organizacional, de manera similar a como se planifica el reuso de software.

Existe un número finito pero potencialmente grande de procesos adaptados, por lo que se requiere de un mecanismo genérico bien definido con el fin de generar el proceso adecuado para cada proyecto. Para ello, la definición del proceso organizacional puede ser vista como una línea de producción de procesos de software. Esta estrategia logra resolver muchos de los problemas de los otros enfoques, como la proliferación de procesos adaptados (si se define un alcance), el enorme tiempo de adaptación (si hay un mecanismo de producción eficiente) y menos probabilidad de error (si hay un mecanismo de producción eficaz).

En el proyecto se desarrollará una solución metodológica y tecnológica para producir familias de procesos de software y validarla en el contexto de al menos seis PyMEs de software chilenas. Para ello, la solución involucra la definición de los siguientes aspectos:

- i. Una forma para especificar la variabilidad en los procesos organizacionales.
- ii. Una forma de caracterizar y representar el contexto de un proyecto.
- iii. Una forma de catalogar las herramientas que forman parte de la plataforma tecnológica de la organización.
- iv. Un método con soporte de herramientas que permita adaptar los procesos organizacionales a contextos específicos a un proyecto, y que permita también definir la plataforma tecnológica requerida para su ejecución.

Esto le dará más valor a la definición del proceso organizacional, ya que tendrá incorporada la posibilidad de adaptación. Por tanto, en el contexto de estas organizaciones habrá una reducción en el esfuerzo y en el costo de los proyectos. Esto mejorará el camino hacia una industria de software nacional más competitiva, permitiendo también desarrollar productos de mejor calidad.

A continuación se presenta un diagrama que puede clarificar cómo ADAPTE pretende abordar el problema.

El problema que se aborda en el presente trabajo de título se enfoca en el centro de la imagen de



Figura 1: Modelo de como ADAPTE pretende abordar el problema

la Figura 1, es decir, en cómo definir el contexto de un proyecto de software utilizando la tecnología y permitir, al usuario final, hacerlo de la forma más fácil e intuitiva posible. Este problema será descrito en detalle en la sección 4.

1.1. Objetivos

El objetivo principal es diseñar e implementar una herramienta que permita configurar contextos para proyectos de software, es decir, proveer una herramienta que, independientemente de la empresa y del proyecto de software, pueda generar un modelo de contexto, para luego poder configurar el contexto concreto más acorde al proyecto. Luego, este contexto concreto será procesado para así poder realizar el proceso de adaptación.

Los objetivos específicos que se derivan del objetivo principal, son los siguientes:

1. Desarrollar una interfaz de administración que permita definir las dimensiones y atributos que componen el modelo de contexto relevante para una empresa particular, así como también los valores potenciales de cada uno de los atributos.
2. Desarrollar una interfaz tal que, dada la definición del modelo de contexto, permita configurar un contexto concreto para un proyecto de software definido.
3. Definir la forma en que el modelo de contexto será entregado a la aplicación que configura el contexto concreto.
4. Implementar el *output* de la aplicación que permite definir el modelo de contexto.
5. Implementar la estructura del archivo de salida que será generado al momento de configurar el contexto concreto. El archivo debe tener extensión *XMI (XML Metadata Interchange)*.
6. Lograr que las interfaces desarrolladas posean un nivel de usabilidad alto.
7. Lograr un gran nivel de robustez para la aplicación desarrollada.

1.2. Relevancia/Motivación

Debido a los problemas asociados a las estrategias de adaptación de procesos organizacionales a procesos específicos de proyectos de desarrollo de software que existen hoy en día, es realmente necesario contar con una nueva estrategia de adaptación, la cual requiera menor esfuerzo humano, tiempo y costo.

El presente trabajo de título, resulta ser suficientemente complejo, ya que dentro de empresas distintas, aunque se dediquen al mismo rubro, los contextos que éstas manejan pueden ser muy diferentes. De la misma forma, cada proyecto de software dentro de una misma empresa puede ser muy distinto a los otros, generando así un contexto distinto para cada proyecto. Aquí es donde se encuentra el desafío, ya que la herramienta a implementar debe ser capaz de soportar cualquier contexto que se intente configurar, con el número de variables que se estime conveniente para el proyecto, sin disminuir la usabilidad de la interfaz. Por lo tanto, la herramienta a implementar debe ser lo suficientemente usable para que el encargado de definir el modelo de contexto pueda realizar su labor de forma sencilla e intuitiva teniendo las mayores libertades que la construcción del modelo

requiera. Además, la aplicación debe contar con un gran nivel de robustez, ya que esto impedirá la duplicidad de datos, las incoherencias y permitirá la creación correcta del modelo de contexto y el contexto concreto a definir.

La principal motivación de este tema de memoria es su relevancia para mejorar el proceso de desarrollo de software y de esta manera generar una mayor productividad para aumentar la competitividad de las PyMES nacionales, permitiéndoles tomar mayores desafíos en la construcción de software. Esto será posible ya que, gracias a la herramienta desarrollada en este trabajo de título, la adaptación automática de los procesos resultará ser una tarea posible.

Sin esta herramienta, la construcción de los modelos de contexto deberían hacerse configurando directamente el *EMT*², lo cual es bastante crítico y ajeno a la costumbre de los usuarios finales de la herramienta desarrollada.

Otra gran motivación está relacionada con la idea inicial del autor al ingresar a la carrera de Ingeniería Civil en Computación: poder resolver problemas complejos para las personas de una forma sencilla en base a tecnología, en este caso, en un producto de software. Justamente esto es lo que se quiere lograr con la implementación de la herramienta en cuestión, ya que como se mencionó, adaptar el proceso organizacional de una empresa de desarrollo de software, a los distintos proyectos que ésta lleva a cabo, es una tarea bastante compleja y costosa. Esto se debe a que actualmente la adaptación e instanciación requiere de tres tipos de conocimientos: diseño de procesos, caracterización de proyectos y caracterización del entorno de desarrollo.

1.3. Alternativas Analizadas

Para el desarrollo de la aplicación propuesta en los objetivos específicos se analizó dos alternativas.

1.3.1. Aplicación de Escritorio

La primera alternativa considerada fue crear una aplicación de escritorio en el lenguaje Java, debido a sus características: es orientado a objetos, es robusto, multiplataforma, portable, entre otras. Estas características permiten que al momento de desarrollar se piense e implementen conceptos como la escalabilidad y extensibilidad. Estas propiedades son muy útiles e importantes en un producto de software, ya que permiten realizar mejoras constantes sin modificar mucho el código.

Sin embargo, esta alternativa no fue la escogida, debido a que el manejo de las interfaces gráficas en Java no es tan manipulable, como lo son las interfaces web. Por lo tanto, al ser la usabilidad un punto muy importante en el presente trabajo, esta alternativa no resultó ser la más adecuada.

²Sigla de Eclipse Modeling Tools, herramienta específica que permite modelar contextos.

1.3.2. Aplicación Web

La segunda alternativa considerada fue una aplicación web. Para escoger el lenguaje de programación, se consideró las características propias de éste, permitiendo una aplicación más robusta, otro punto importante en el presente trabajo. Para las vistas el manejo es independiente del lenguaje, ya que estas siempre son codificadas a *HTML*.

Hoy en día, la gran mayoría de las personas navegan por páginas en internet, aprendiendo y teniendo el conocimiento de una gran cantidad de funcionalidades, disposición de la información, comportamiento de ciertos objetos dentro de una página y de la iconografía existente. Este aprendizaje permite que el desarrollo web posea un nivel de usabilidad mayor que una aplicación de escritorio.

Considerando lo antes mencionado, se eligió la realización de una aplicación web, ya que en el manejo de interfaces el programador posee libertad absoluta para diseñar e implementar las vistas, además de la existencia de bastantes librerías que permiten diseños y animaciones atractivas y conocidas por los usuarios. El lenguaje escogido fue Java, dadas las características expuestas en la primera alternativa.

Dentro de la programación web con el lenguaje Java, existen diversos *frameworks* que permiten al desarrollo de la aplicación ser mucho más ordenada y tener una arquitectura definida. En el caso de la aplicación a desarrollar, se escogió un *framework* web llamado *SpringMVC*, el cual permite desarrollar una aplicación bajo la arquitectura MVC (Modelo Vista Controlador).

1.4. Descripción general de la solución

La solución propuesta consiste en una aplicación web dividida en dos módulos y que está realizada en *SpringMVC*, un *framework* web cuya arquitectura está basada en el patrón Modelo Vista Controlador (MVC).

1.4.1. Definición del modelo de contexto (módulo Modelo de Contexto)

Este módulo está compuesto por cuatro interfaces. La primera interfaz es una página de inicio, en la cual se da la opción al usuario de cargar un modelo de contexto mediante un archivo XML, el cual tiene un formato definido. La otra opción es ir a la siguiente interfaz sin cargar ningún archivo, cargandose así un modelo de contexto por defecto. Este modelo de contexto, se basa en el metamodelo *SPCML* (*Software Project Context Modeling Language*), explicado en detalle en la sección 4.2. Este consta de tres elementos básicos: dimensiones, atributos y los posibles valores para estos atributos.

El metamodelo *SPCML*, además de los tres elementos mencionados anteriormente, permite la creación de enlaces o *links* de prioridad entre los atributos, estableciendo prioridad o una relación de importancia jerárquica entre dos o más atributos del modelo de contexto.

En la segunda interfaz, el usuario tendrá todas las libertades de agregar, editar y eliminar

cualquier elemento presente en el modelo, ya sea una dimensión, un atributo o posibles valores de un atributo. Una vez definido el modelo de contexto, se avanza a la siguiente interfaz, la que consiste en crear los *links* de prioridad entre los atributos del modelo de contexto. En esta interfaz se despliega una grilla de cajas seleccionables, las cuales permiten establecer prioridades entre los atributos. Aquí el usuario tiene la opción de crear los enlaces seleccionando alguna de las cajas de la grilla o no crear ningún enlace de prioridad dejando todas las cajas en blanco. Luego se avanza a la siguiente interfaz.

Finalmente, en la cuarta interfaz se le informa al usuario de la creación exitosa del modelo de contexto, permitiéndole descargar un archivo *XML*, el cual contendrá la definición completa del modelo de contexto. Este archivo será el *input* o dato de entrada para el siguiente módulo.

1.4.2. Configuración del contexto concreto (módulo Contexto Concreto)

Este módulo se compone de tres interfaces. La primera interfaz, bastante similar a la interfaz inicial del módulo anterior, consiste en dar la bienvenida al usuario y solicitarle que cargue el archivo que contiene el modelo de contexto. Aquí no será opcional cargar el archivo, ya que éste debe ir obligatoriamente. Una vez cargado el archivo, se puede acceder a la siguiente interfaz.

La segunda consiste en configurar el contexto concreto. Esta interfaz es bastante sencilla para el usuario, ya que se muestran los atributos categorizados por su dimensión donde para cada atributo es posible seleccionar un valor entre todos los posibles que posea. Una vez asignados todos los valores es posible avanzar a la tercera y última interfaz de este módulo.

La tercera interfaz consta de un mensaje al usuario, informándole de la correcta creación del contexto concreto, permitiéndole descargar un archivo *XMI* que contendrá el contexto concreto configurado.

El *output* o archivo de salida de este módulo será uno de los archivos que pasarán por un proceso de transformación, el cual dará como resultado el proceso organizacional adaptado para un proyecto de software en particular para el contexto modelado.

1.5. Resultados de la solución implementada para resolver el problema

Los resultados obtenidos son detallados en las secciones 5 y 6. En la sección 5 se describe paso a paso el uso de la aplicación, validando todos los requisitos funcionales de ésta, ya que es posible verificar la existencia de todas las funcionalidades necesarias de la solución.

En la sección 6 se muestra en detalle los resultados de pruebas y encuestas de usabilidad efectuadas a cuatro personas, cuyo perfil es el de los usuarios finales de la aplicación (Ingeniero de Procesos y Jefe de Proyecto). También se realizan pruebas de robustez, las cuales confirman que la aplicación entrega datos coherentes y correctos. Finalmente, en esta misma sección se comprueba que el formato del archivo final de salida (*XMI*), que se obtiene del módulo Contexto Concreto de la aplicación sea correcto. Esta validación se realiza a través de la herramienta *Eclipse Modeling Framework (EMF)*.

2. Antecedentes

A continuación se presentan una serie de conceptos, que son la base del marco teórico necesario para poder implementar la solución más adecuada al problema descrito en la sección 3.

2.1. MVC

Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos. Este patrón se ve frecuentemente en sistemas de información y aplicaciones web, donde el Modelo representa al Sistema de Gestión de Base de Datos y a la Lógica de negocio, es decir, la representación específica de la información con la cual el sistema opera; la Vista es la interfaz de usuario y el código que provee de datos dinámicos a la página; y el Controlador es el responsable de recibir los eventos o peticiones de entrada desde la Vista y ser el que interactúa entre los datos y las vistas, decidiendo qué mostrar dado un evento específico.

2.2. Framework

La palabra *framework* define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software es una estructura conceptual y tecnológica de soporte definido.

2.3. Eclipse Modeling Framework

Eclipse Modeling Framework (EMF) es un *framework* de modelamiento que facilita la generación de código para la creación de herramientas y otras aplicaciones MDE. Desde la especificación de un modelo descrita en XMI, EMF provee herramientas y soporte en ejecución para producir un conjunto de clases Java para el modelo, un editor para describir modelos y soporte en ejecución para los mismos, incluyendo un notificador de cambios, soporte a la persistencia con serialización XMI por defecto y una API muy completa para manipular objetos EMF de manera genérica [8].

Una de las principales funcionalidades que se puede hacer con este framework es la definición de un metamodelo el cual se formaliza en un archivo de extensión *.ecore*. Además, la herramienta permite la visualización de archivos XMI de forma comprensible, entregando representaciones jerárquicas por medio de estructuras de árbol. La Figura 2 ilustra un archivo XMI representado por medio de una estructura de árbol.

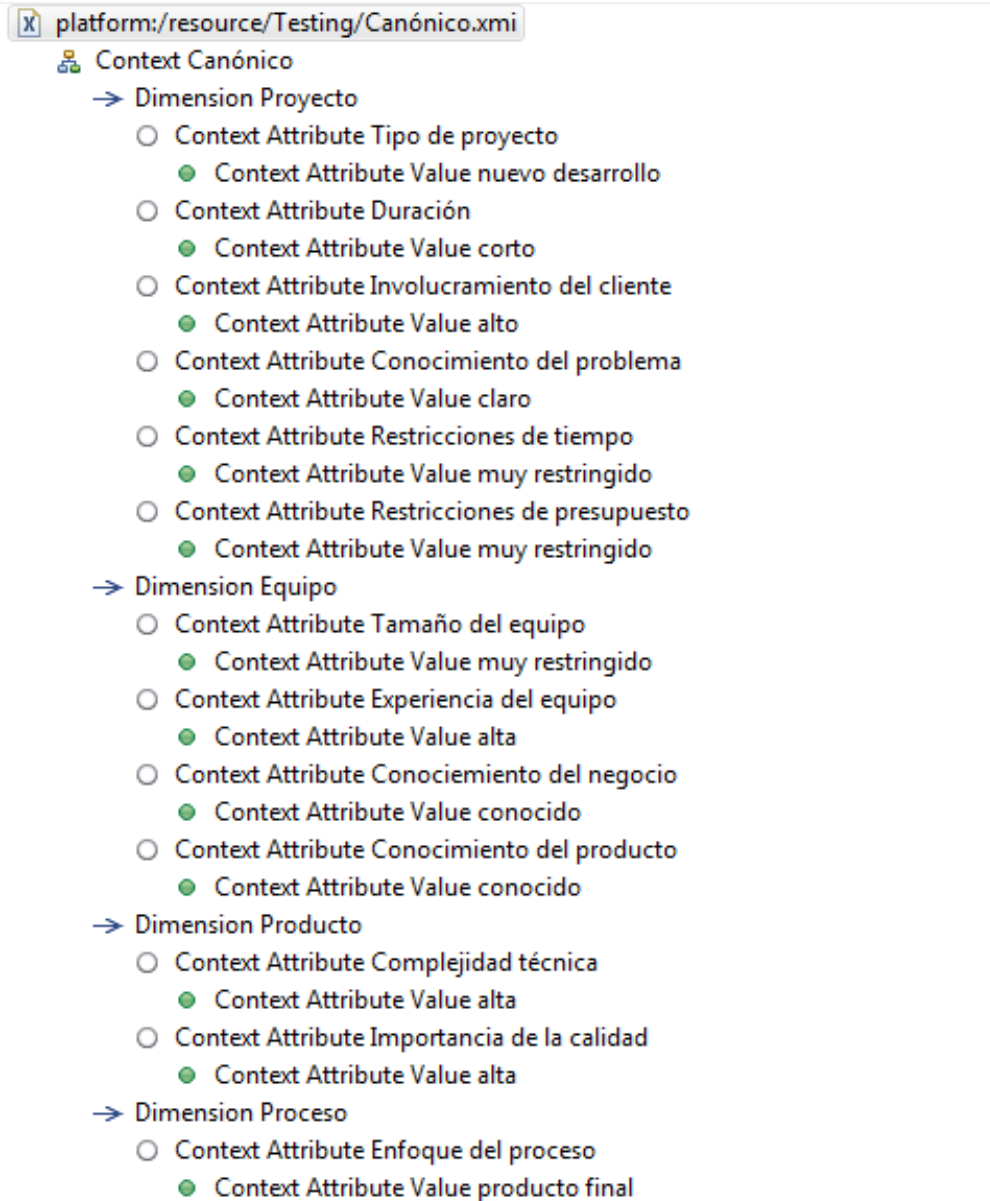


Figura 2: Representación jerárquica de un contexto en formato XMI. Este archivo XMI es el resultado de un contexto configurado con la herramienta implementada en el presente trabajo.

2.4. Usabilidad

Se puede definir usabilidad como la medida en la cual un producto puede ser usado por usuarios específicos para conseguir ciertos objetivos con efectividad, eficiencia y satisfacción en un contexto de uso especificado. La usabilidad hace referencia a la rapidez y facilidad con que los usuarios finales llevan a cabo sus tareas propias a través del uso del producto objeto de interés.

Dentro de la usabilidad en Software/Sitio Web, se encuentran los siguientes atributos, los cuales ayudan a definir de mejor forma cuán usable es un producto de software: [7]

1. Aprendizaje: responde a la pregunta ¿puede el Software/Sitio Web ser fácilmente aprendido de manera que el usuario realice rápidamente una tarea con el apoyo del sistema?
2. Eficiencia: responde a la pregunta ¿es el sitio eficiente en su uso de manera que una vez aprendido a utilizarlo pueda generarse mayor productividad?
3. Recuerdo: responde a la pregunta ¿es el sitio fácil de recordar de manera que un usuario casual sea capaz de retornar al sitio, después de un período sin utilizarlo, sin tener que aprender todo de nuevo?
4. Errores: responde a la pregunta ¿tiene el Software/Sitio Web una baja tasa de errores de forma que los usuarios los cometan poco durante el uso del sitio y si ellos los cometen puedan salir fácilmente de ellos, por lo que errores catastróficos no debieran ocurrir?
5. Satisfacción: responde a la pregunta ¿es el Software/Sitio Web placentero al utilizarlo de forma que los usuarios están subjetivamente satisfechos cuando lo usan y les gusta?

2.5. Robustez

Cuando se habla de robustez, se refiere a la capacidad y proceso de reacción apropiada ante condiciones que se encuentren fuera del alcance del software. Cabe destacar que estas condiciones son excepcionales. La robustez de un software es la que permite que, ante situaciones límites, la aplicación pueda responder de la mejor forma posible. Si se concluye la ejecución del software a consecuencia de un error, ésta pueda terminar la ejecución de la forma más limpia posible, manteniendo la coherencia entre los datos e información procesada.

2.6. Contexto

En la literatura existen muchas definiciones para contexto en ciencias de la computación. En el caso del desarrollo de software, se define contexto de un proyecto como el conjunto de atributos que caracterizan el proyecto, el producto a ser desarrollado, los recursos que participan dentro de éste, las herramientas que serán utilizadas y las condiciones ambientales [11].

Para el desarrollo de este trabajo, se hablará de modelo de contexto, dentro del cual se destacan tres conceptos importantes: dimensiones, atributos y posibles valores de atributos. A continuación se da una pequeña definición de cada uno:

- Dimensiones: representan un grupo de atributos contextualmente relacionados. Los trabajos de investigación han llamado a las dimensiones con diversos nombres, tales como categorías y factores de escala.
- Atributo: representa a un elemento contextual particular dentro de una dimensión. Se les han asignado varios otros nombres, tales como característica, factor, criterio y restricción.
- Valor atributo: expresa un valor específico asumido por un atributo del contexto. Los valores de los atributos han sido llamados valor de factor, criterio y respuesta.

De esta manera, por ejemplo, se podrá decir que el modelo de contexto de un proyecto de software tendrá tres dimensiones: Proyecto, Personas y Producto. Los atributos relevantes para cada dimensión pueden ser:

- Proyecto:
 - Tipo (desarrollo, extensión, incidente, mantenimiento);
 - Duración (largo, mediano, corto)
- Producto:
 - Conocimiento del dominio (alto, medio, bajo);
 - Complejidad (alta, media, baja)
- Personas:
 - Tamaño del equipo (grande, mediano, chico);
 - Capacitación (alta, media, baja)

2.7. Escala de Likert

La escala de tipo Likert (también denominada método de evaluaciones sumarias) es una escala psicométrica comúnmente utilizada en cuestionarios, y es la escala de uso más amplio en encuestas para la investigación, principalmente en ciencias sociales. Al responder a una pregunta de un cuestionario elaborado con la técnica de Likert, se especifica el nivel de acuerdo o desacuerdo con una declaración (elemento, ítem o reactivo o pregunta). La escala se llama así por Rensis Likert, quien publicó en 1932 un informe donde describía su uso [15].

2.8. jQuery

jQuery es una librería de JavaScript rápida y concisa que simplifica el recorrido por el documento HTML, el manejo de eventos, y las interacciones Ajax para un desarrollo web rápido. jQuery está diseñado para cambiar la forma en que se escribe JavaScript [12].

3. Especificación del problema

3.1. Problema

Para resolver el problema que aborda ADAPTE anteriormente planteado se debe realizar una serie de procesos los cuales se logran visualizar de mejor manera en la siguiente imagen.

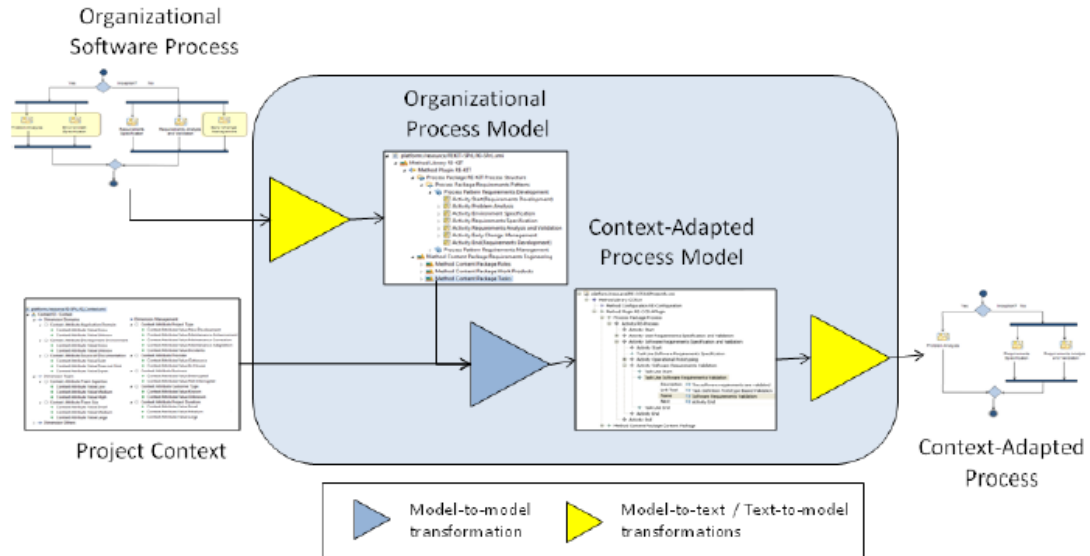


Figura 3: Etapas para realizar el proceso de adaptación con ADAPTE.

Como se observa, existen dos procesos que se pueden ejecutar en paralelo. Uno de estos comienza con la utilización de la herramienta EPF Composer, con el fin de modelar el proceso organizacional de la empresa. Una vez hecho esto, el resultado del modelamiento se exporta en un archivo XML (Extensible Markup Language). Luego este archivo es procesado por un parser, el cual convierte el archivo XML en un archivo XMI (XML Metadata Interchange).

El otro proceso consiste en configurar el contexto de un proyecto particular, dando como resultado un archivo XMI.

Ambos archivos serán procesados por una herramienta de modelamiento, obteniendo como resultado un nuevo y único archivo XMI con el proceso organizacional o de negocio adaptado.

Estos procesos tienen asociados problemas específicos. Uno de ellos, relacionado con el proceso de configuración de contextos, fue solucionado en la presente memoria.

El problema que presenta la configuración es que esta labor se debe realizar con personal especializado en la herramienta Eclipse Modeling Tools (EMT), involucrando costos en personal, tiempo y dinero. Por lo tanto, es necesario brindar una solución de tal manera que usuarios comunes, sin conocimiento en esta herramienta de modelamiento, logren definir un modelo de contexto, para luego poder configurar el contexto concreto que se adecúe a un proyecto de software en particular. De esta manera, se logra que la ejecución de esta tarea resulte de forma intuitiva y agradable para el usuario final.

En definitiva el gran problema en la creación de un contexto concreto para un proyecto de software en particular es la inexistencia de una herramienta intuitiva y fácil de usar por usuarios de la misma empresa, en este caso, un Ingeniero de Procesos y un Jefe de Proyecto.

3.2. Requisitos Específicos

El foco del problema se concentra en lograr una herramienta lo suficientemente usable, para que los usuarios finales puedan desarrollar su labor de la manera más simple e intuitiva posible, además de entregar una herramienta robusta que garantice un resultado completamente coherente y sin errores.

Los requisitos funcionales específicos del problema son:

1. Desarrollar una aplicación que permita definir el modelo de contexto de la empresa. La aplicación debe permitir:
 - Asignar un nombre y descripción al modelo de contexto a definir.
 - Editar, agregar y eliminar una dimensión.
 - Editar, agregar y eliminar un atributo a una dimensión específica.
 - Editar, agregar y eliminar un posible valor para un atributo específico.
 - Crear *links* de prioridad entre los atributos del modelo de contexto.
 - Descargar un archivo que contenga la definición del contexto.
2. Desarrollar una aplicación que permita configurar un contexto concreto de un proyecto de software específico. La aplicación debe permitir:
 - Cargar un archivo con un modelo de contexto definido.
 - Asignar un nombre y descripción al contexto concreto a configurar.
 - Asignar un valor para cada atributo presente en el modelo de contexto.
 - Descargar un archivo que contenga la configuración del contexto.
3. Definir e implementar el formato del archivo que se obtendrá al finalizar la definición del modelo de contexto.
4. Implementar el formato del archivo XMI que se obtendrá al configurar el contexto concreto.

Los requisitos no funcionales específicos del problema son:

1. Lograr altos niveles de usabilidad en la aplicación.
2. Lograr un alto nivel de robustez en la aplicación.

3.3. Casos de Uso

Los Casos de Uso de la aplicación desarrollada son:

- Los casos de uso para el módulo Modelo de Contexto de la aplicación, mencionado en la sección 1.4.1, se ilustran en la Figura 4

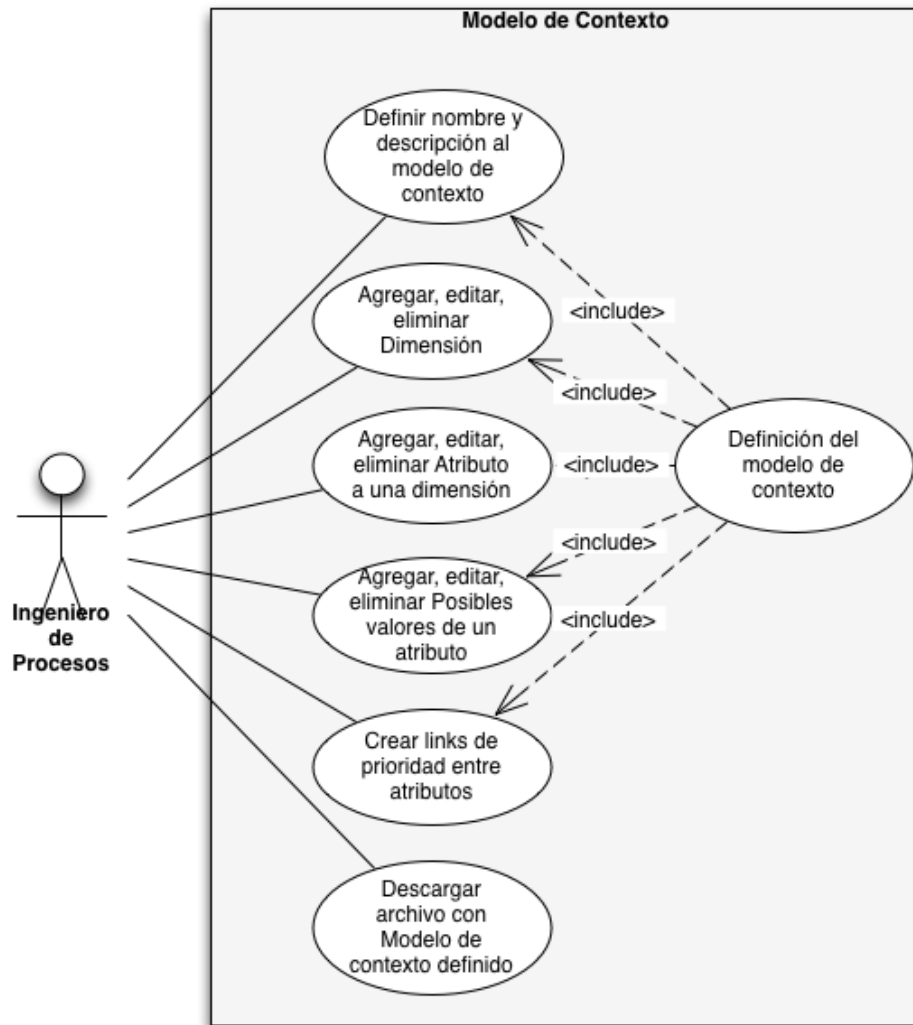


Figura 4: Casos de uso para el módulo Modelo de Contextos

- Los casos de uso para el módulo Contexto Concreto de la aplicación, mencionado en la sección 1.4.2, se ilustran en la Figura 5

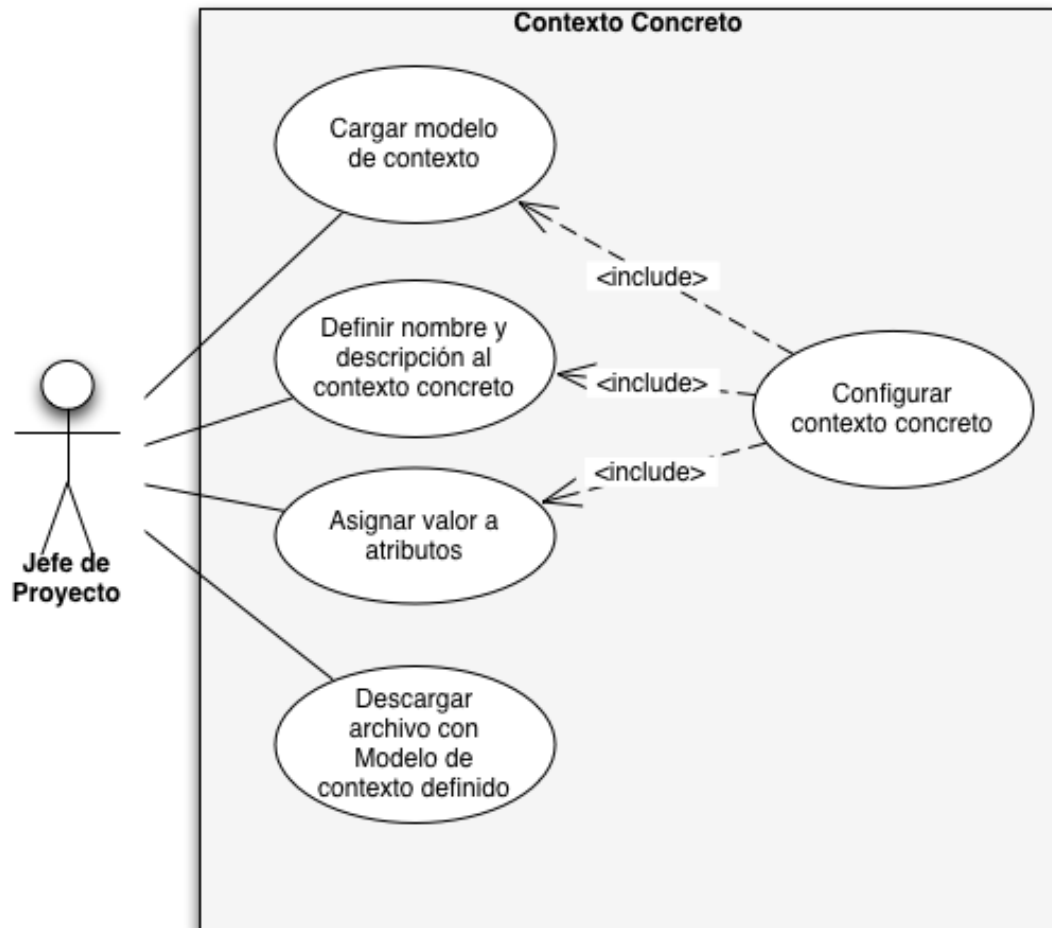


Figura 5: Casos de uso para el módulo Contexto Concreto

4. Descripción de la solución

4.1. Especificación de contextos

El contexto de un proyecto se define como el conjunto de instancias de atributos que caracterizan al proyecto, el producto a desarrollar, los recursos que participan, las herramientas a usar y las condiciones del ambiente. El contexto es el elemento bajo el cual la variabilidad de los procesos toma sentido. Si un mecanismo basado en un contexto no es usado en la adaptación, un modelo de proceso de software adaptable es insuficiente.

Para la configuración de contextos concretos, el desarrollo de la aplicación se basó en el reporte “A Context Modeling Language to Support Tailoring of Software Processes”[11].

Para la definición de los modelos de contextos se utilizó un modelo base o canónico, el cual se encuentra especificado en el metamodelo SPCML (Software Project Context Modeling Language) descrito a continuación.

4.2. Software Project Context Modeling Language (SPCML)

SPCML es una propuesta que ha sido obtenida desde experiencias empíricas de los autores y consultores en el mejoramiento y definición de procesos de software. Esta propuesta se ha hecho con el fin de considerar los factores relevantes para la adaptación de estos procesos en la práctica de las PyMEs.

Esta propuesta será el conjunto de datos mínimos para la interfaz del módulo de la aplicación que permite definir modelos de contextos. El lenguaje está compuesto por dos componentes básicos: los atributos del contexto y los enlaces entre estos atributos.

En la Figura 6 se ilustra el metamodelo SPCML mediante un diagrama.

Los atributos considerados en SPCML fueron agrupados en cuatro dimensiones de contexto: proyecto, equipo, producto y proceso. El caso canónico que se utilizó en la aplicación es el siguiente:

1. Proyecto

- a) Tipo de proyecto: {nuevo desarrollo, extensión, mantenimiento}
- b) Duración: {corto, medio, largo}
- c) Involucramiento del cliente: {alto, medio, bajo, conocido}
- d) Conocimiento del problema: {claro, ambiguo, no claro}
- e) Restricciones de tiempo: {muy restringido, típico, sin restricciones}
- f) Restricciones de presupuesto: {muy restringido, típico, sin restricciones}

2. Equipo

- a) Tamaño del equipo: {muy restringido, típico, sin restricciones}
- b) Experiencia del equipo: {alta, regular, baja}

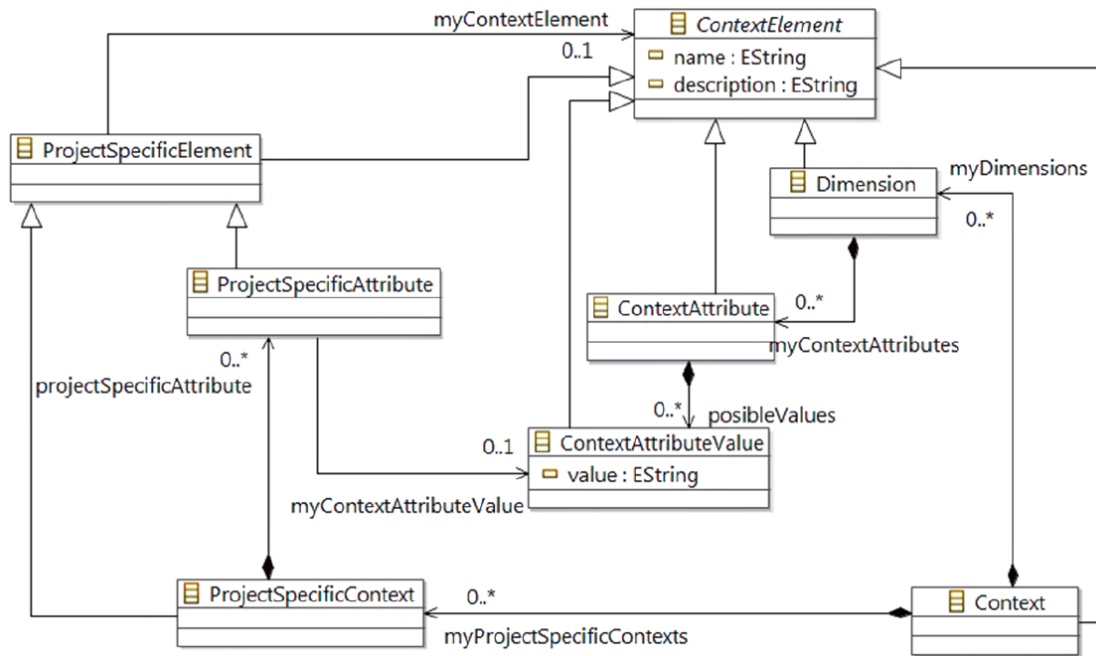


Figura 6: Diagrama Metamodelo SPCML

c) Conocimiento del negocio: {conocido, abordable, desconocido}

d) Conocimiento del producto: {conocido, abordable, desconocido}

3. Producto

a) Complejidad técnica: {alta, media, baja}

b) Importancia de la calidad: {alta, regular, mínima}

4. Proceso

a) Enfoque del proceso: {producto final, todo el producto}

Estas dimensiones y sus atributos han sido identificados en la literatura, en las experiencias de los autores y participantes en el proyecto ADAPTE. Además, en esta definición fueron consideradas las particularidades del desarrollo en el típico escenario presente en las PyMEs.

Con respecto a los enlaces considerados en SPCML, todos ellos establecen relaciones entre dos atributos del contexto. El lenguaje establece tres tipos de enlaces:

1. Enlace de prioridad: se representa por una flecha que se despliega desde el atributo fuente al atributo destino; en este caso se debe asignar prioridad al primero por sobre el segundo, es decir, se establece una relación jerárquica entre dos o más atributos de contexto.
2. Enlace *peer-to-peer*: indica que dos atributos están relacionados, pero que no existe una relación jerárquica entre ellos.

3. Sin relación: los atributos son independientes y por lo tanto pueden ser considerados por separado.

En el trabajo realizado, sólo se consideró la primera clase de *links*, es decir, los enlaces de prioridad.

A continuación se ilustra cómo se compone un modelo de contexto mediante un diagrama.

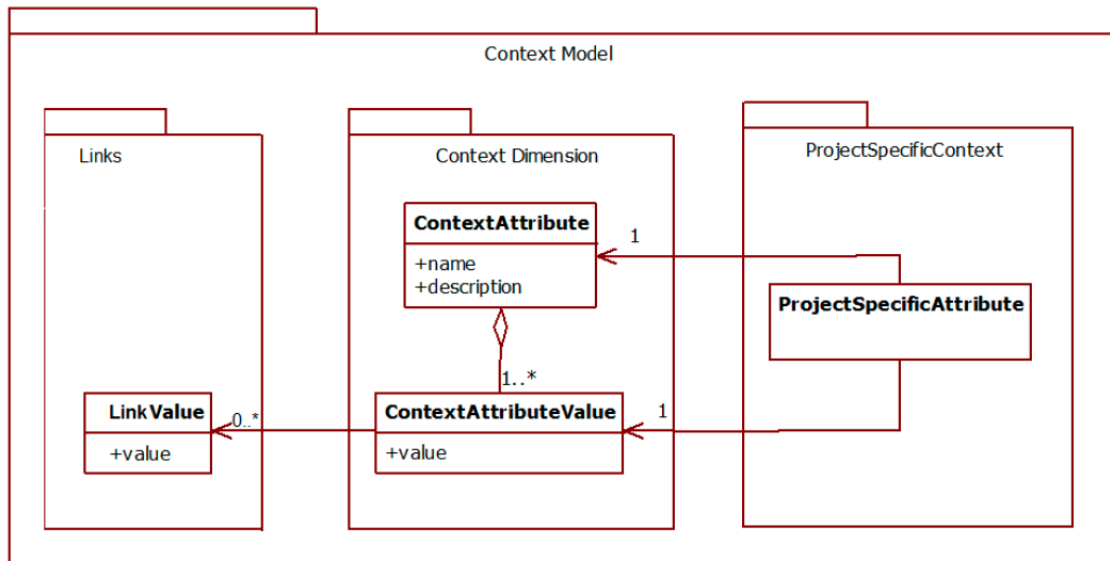


Figura 7: Diagrama Modelo de Contexto

4.3. Arquitectura de software

La aplicación desarrollada está basada en una arquitectura Modelo Vista Controlador (MVC). El desarrollo de aplicaciones web bajo esta arquitectura suele dar como resultado un producto de software de buena calidad, adquiriendo características deseables como lo son la extensibilidad, la mantenibilidad y escalabilidad, entre otras. Esta arquitectura permite separar de buena manera la Vista del Modelo (bases de datos o cualquier tipo de dato involucrado en el sistema). La Figura 8 ilustra de forma gráfica como funciona la arquitectura MVC.

Se decidió no persistir en una base de datos ya que no era necesario. Por el contrario sí es necesario generar un archivo que fuese el *output* del módulo que define el modelo de contexto, y a su vez, este archivo fuese el *input* del módulo que configura el contexto concreto, es decir, un archivo que sirviera de nexo entre ambos módulos.

4.4. Arquitectura de hardware

La arquitectura de hardware es la típica que se requiere para las aplicaciones web. Esta es la arquitectura Cliente/Servidor. La Figura 9 ilustra cómo funciona esta arquitectura.

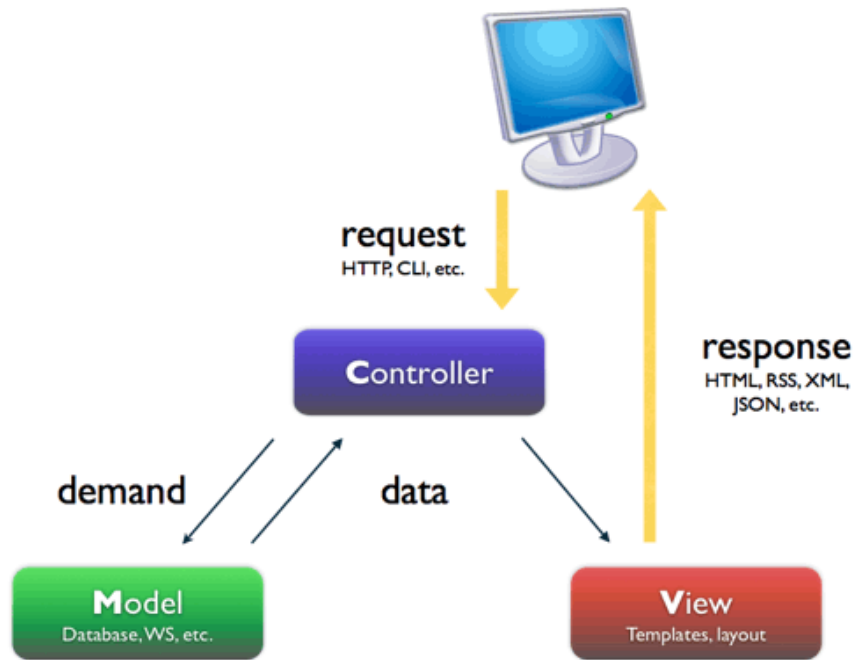


Figura 8: Funcionamiento del patrón MVC



Figura 9: Arquitectura Cliente/Servidor

Para la arquitectura Cliente/Servidor se requiere de una interfaz de usuario que se instala y se ejecuta en un PC o estación de trabajo para enviar solicitudes a un servidor que ejecutará operaciones complejas.

En el caso de la aplicación desarrollada, en el cliente se debe ejecutar la aplicación mediante un *Browser* o Navegador (de preferencia, Firefox, Chrome o Safari), y como servidor de aplicaciones se utilizará Tomcat (de preferencia Tomcat6).

Como se mencionó anteriormente no se persistirá datos. Debido a esto no será necesario interactuar con una base de datos. Como sí es necesario almacenar los datos procesados, estos se guardarán en un archivo XML (Modelo de Contexto) y en un archivo XMI (Contexto Concreto), los cuales podrán ser descargados por el usuario una vez terminadas las tareas necesarias en los módulos correspondientes de la aplicación.

4.5. Diseño de archivos que almacenan la información procesada

Para guardar los datos procesados en el módulo donde se define el modelo de contexto, se decidió utilizar un archivo XML. Esta decisión se tomó en base al requisito de generar un archivo XMI como salida de la configuración del contexto concreto, ya que ambos archivos tienen una estructura básicamente igual, con la diferencia que los elementos de los archivos XMI poseen *ids* que permiten relacionar elementos dentro del mismo archivo. Los archivos XML y XMI, poseen la siguiente jerarquía:

- Context
- myDimensions
- myContextAttributes
- posibleValues
- Links (Si se han creado)
- myContextConfigurations (Estos elementos solo existen en el archivo XMI)

Todos estos elementos presentan sus propios atributos. En este caso los atributos para cada elemento son un nombre (“name”) y una descripción (“description”). Para el elemento posibleValue, además de los atributos mencionados, posee el atributo valor (“value”).

Los valores de estos atributos son asignados según la configuración que el usuario haya realizado en cualquiera de los módulos de la aplicación.

Además de estos elementos, también estarán presentes los elementos que representen los *links* de prioridad entre los atributos. Estos sólo estarán dentro del elemento contexto, además de tener su propia jerarquía, la cual se presenta a continuación:

- Links
- PriorityAttribute

Estos elementos poseen un único atributo, cuyo nombre es atributo (“attribute”), que representa

a un “myContextAttributes” del contexto. El elemento Link tendrá como atributo, algún atributo “myContextAttributes” del modelo de contexto, el cual tendrá prioridad sobre todos los atributos de los elementos PriorityAttribute. Por Ejemplo:

- Links attribute = atributo 1
- PriorityAttribute attribute = atributo 2
- PriorityAttribute attribute = atributo 3

Esta jerarquía significa que el “myContextAttributes” cuyo nombre es atributo 1, tiene prioridad sobre atributo 2 y atributo 3.

Para los archivos XMI, además de tener los elementos anteriormente descritos, tendrán los elementos que representen la configuración concreta del contexto. Estos elementos se encontrarán dentro de la jerarquía contexto, además de tener su propia jerarquía. Esta jerarquía es la siguiente:

- myContextConfigurations
- contextAttributeConfiguration

Estos elementos también poseen los atributos name y description, pero además poseen el atributo id, el cual es un identificador único para el elemento dentro del contexto. El elemento contextAttributeConfiguration además de los atributos mencionados, posee el atributo ”myContextAttributeValue”, cuyo valor es el id del posibleValues al cual esté asociado y el atributo ”myContextElement”, cuyo valor también es una id. Esta id corresponde al myContextAttribute al que corresponda este elemento.

En los anexos del presente informe es posible visualizar dos imágenes que representan un archivo XML y un archivo XMI. La Figura A1 es un archivo XML que representa un modelo de contexto definido, la Figura B1 es un archivo XMI, que representa un contexto concreto configurado. Cabe destacar que ambos archivos son muestras obtenidas con la aplicación realizada en el presente trabajo de título.

4.6. Diseño de clases

Todas las clases java de la aplicación se encuentran divididas en tres packages distintos. Estos packages y sus respectivas clases son:

- Controller
 - ConcreteContextController
 - ContextModelController
- Domain
 - Context
 - ContextAttribute

- ContextAttributeValue
- ContextDimension
- ContextElement
- Link
- ProjectSpecificAttribute
- ProjectSpecificContext
- ProjectSpecificElement
- Utils
 - ContextToXMI
 - ContextToXML
 - DocumentUtils
 - ElementUtils
 - FileUtils
 - IdGenerator
 - Utils
 - XMLToContext
 - XMLToLinks

4.6.1. Package Controller

El package Controller, como su nombre lo dice, almacena los controladores que participan en la aplicación. Estas clases Controller existen, ya que son necesarias para poder seguir la arquitectura MVC propuesta como la arquitectura de software base para la aplicación. El rol de estas clases es básicamente responder a las peticiones o acciones que se realizan desde la interfaz por parte del usuario. Estas peticiones son interpretadas por el controlador y enviadas al servidor de aplicaciones, en donde se llevará a cabo la tarea solicitada en la petición. Una vez resuelta la petición en el servidor, el servidor se comunica con el controlador entregando el resultado de lo solicitado. Luego el controlador devuelve los resultados en una nueva interfaz.

Son dos los controladores presentes en la aplicación, ConcreteContextModel y ContextModel-Controller. El primero es quién controla e intercambia las peticiones entre las interfaces y el servidor del módulo de creación de contextos concretos. El segundo controlador es para el módulo de la definición de modelos de contextos.

Dentro de las clases Controller se pueden destacar elementos que facilitaron la asociación entre elementos del formulario con los objetos del sistema. Estos elementos son:

1. `@ModelAttribute` (`org.springframework.web.bind.annotation.ModelAttribute`):
Elemento que permite el correcto “binding” o enlace entre los elementos del formulario (vista) y los atributos de las entidades (modelo). Si el binding es correcto, es posible trabajar con el `ModelAttribute` como un objeto propiamente tal.
2. `MultipartFile` (`org.springframework.web.multipart.MultipartFile`):
Elemento que permite el manejo de archivos en el servidor, permitiendo crear directorios temporales, para poder trabajar con los archivos sin necesidad de subirlos permanentemente al servidor. Esto resultó muy útil para la creación de los archivos XML y XMI, ya que por seguridad, las aplicaciones web no permiten obtener la ruta absoluta de un archivo subido desde un computador, lo cual hacía imposible tratar de abrir el archivo directamente. Entonces gracias a `MultipartFile`, se crean directorios temporales, en donde se almacena un archivo con el contenido del archivo subido y luego se accede al archivo del directorio temporal. Luego, obteniendo la información del archivo es posible construir los objetos de la aplicación los cuales son mostrados mediante una interfaz. Para la descarga de archivos se realiza el mismo ejercicio.
3. `HttpServletRequest` (`javax.servlet.http.HttpServletRequest`):
Elemento que permite el envío de todo tipo de dato dentro del servidor. Este elemento es de mucha ayuda al momento de enviar objetos completos desde una interfaz a otra.
4. `@RequestMapping` (`org.springframework.web.bind.annotation.RequestMapping`):
Anotación que permite “mapear” las acciones de un controlador para ser llamadas desde las vistas.
5. `ModelAndView`:
Objeto que permite hacer “rendering” o pintar una vista en base a una página JSP dada. Al objeto `ModelAndView` es posible agregarle cuantos parámetros sean necesarios mediante el método `addObject(“parametro”, parametro)`. Este método se utilizó para enviar objetos desde el modelo hacia las vistas.

4.6.2. Package Domain

El package Domain contiene todas las clases que representan a las entidades de la aplicación. La mayoría de estas clases sólo poseen los métodos getters y setters. La clase Context posee un método que es muy importante para la definición del modelo de contexto; este método lo que hace es crear un contexto con todos los elementos del modelo de contexto descrito en 4.2.

Es posible visualizar las clases y su relación mediante el diagrama de clases de la Figura 10

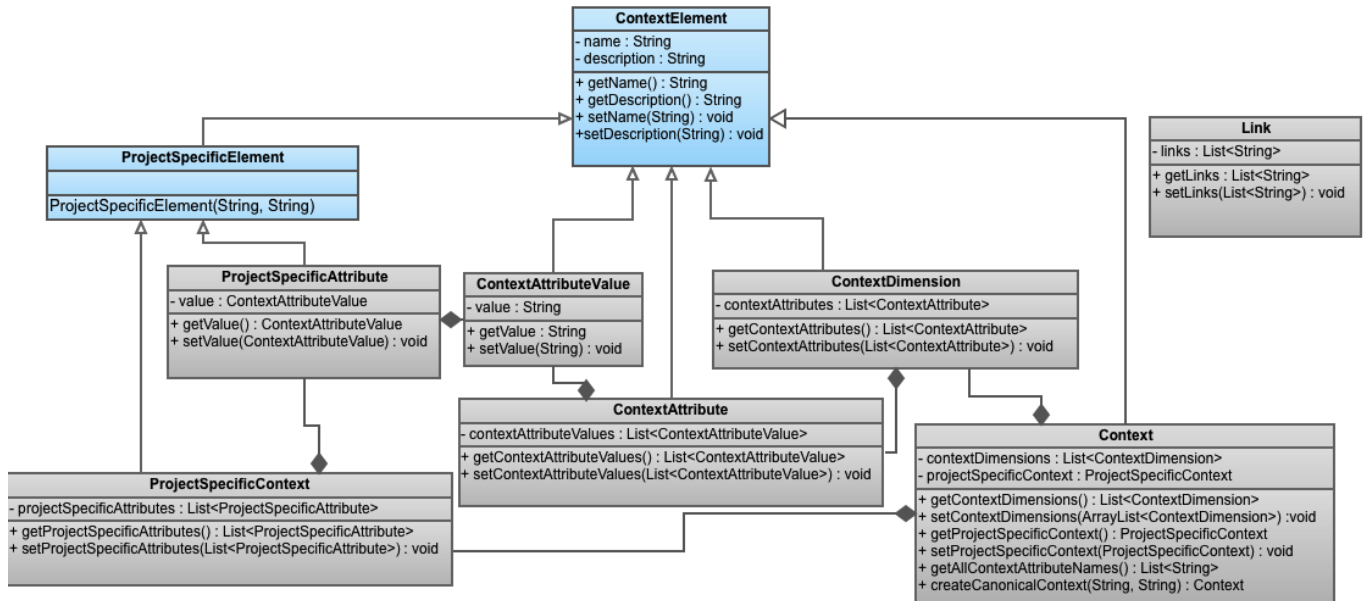


Figura 10: Diagrama de clases del package Domain

4.6.3. Package Utils

Dentro del package Utils, se encuentran clases muy importantes, las cuales permiten la creación de los archivos XML y XMI descritos en 4.5. Antes de describir las clases y sus métodos más importantes, se describen dos objetos que resultaron ser de mucha ayuda para el manejo de archivos XML y XMI.

Elements y Documents: Estos dos objetos fueron fundamentales para facilitar el manejo de los archivos XML y XMI. Ambos objetos serán descritos además de las principales funciones utilizadas.

- **DocumentBuilder** (`javax.xml.parsers.DocumentBuilder`):

Este elemento hace “parsing” (internamente) del archivo XML y lo transforma en un elemento Document. Para crear un objeto DocumentBuilder, es necesario tener un DocumentBuilderFactory, que crea el objeto con `DocumentBuilder documentBuilder = documentBuilderFactory.newDocumentBuilder()`. Para crear un nuevo objeto Document, a partir de un documento (puede ser su ruta como String, o un objeto `java.io.File`), se usa `Document document = documentBuilder.parse(“file.xml”)`.
- **Document** (`org.w3c.dom.Document`):

Esta clase representa el documento XML (o XMI) en sí. Se utiliza principalmente para obtener el objeto Element, mediante la llamada `Element element = document.getDocumentElement()`, que permite crear un element con el primer hijo o hijo raíz del documento. También se necesita el documento para obtener un nuevo elemento, que no necesariamente será una nueva

raíz del documento, ya que también puede ser un hijo de otro elemento. Para esto es necesario entregarle un nombre al nuevo elemento (nodo); esto se hace con `Element anElement = document.createElement(nodeName)`

- **Element** (`org.w3c.dom.Element`):

`Element` representa un nodo dentro del documento XML (o XMI), extiende de la clase `Node` (`org.w3c.dom.Node`) y contiene un nombre (`getNodeName()`), un valor (`getNodeValue()`), y contenido (`getTextContent()`). Además se puede acceder a los atributos del nodo en un objeto `NamedNodeMap` (`org.w3c.dom.NamedNodeMap`) mediante el método `NamedNodeMap attributes = element.getAttributes()`; el objeto `NodeList` (`org.w3c.dom.NodeList`) permite obtener todos los elementos hijos del nodo mediante `NodeList listOfChilds = element.getChildNodes()`. También se puede obtener un atributo por su nombre con `String name = element.getAttribute("name")`, y todos los elementos hijos de la jerarquía por su tag name, almacenándolos en el ya mencionado objeto `NodeList`, con `NodeList tagelems = element.getElementsByTagName("TagName")`. Finalmente se puede asignar un atributo mediante `element.setAttribute("name", name)`, y agregar un nuevo elemento hijo a este elemento, con `element.appendChild(childElement)`.

- **Transformer** (`javax.xml.transform.Transformer`):

Esta clase sirve en el momento de generar el archivo de salida, es decir, transformar nuevamente el `Document` en un documento XML (o XMI). Para tener un transformer es necesario primero inicializar un `TransformerFactory` (`javax.xml.transform.TransformerFactory`). Es en este objeto donde se agregan atributos como el *encoding* y la indentación. Para la transformación final es necesario tener un `Source` (`javax.xml.transform.dom.Source`), el cual para inicializarse necesita un `Element`, `Source src = new DOMSource(element)`, y un `Result` (`javax.xml.transform.Result`), que para inicializarse necesita un `Writer` (`java.io.Writer`), `Result rst = new StreamResult(writer)`. Finalmente la transformación se realiza con `transformer.transform(src, rst)`.

1. `ElementUtils`:

Esta clase consta de un solo método, el cual permite manejar de mejor manera los problemas relacionados a un objeto `Element`.

```
public static List<Element>getChildsByTagName(Element elem, String tagName)
```

Este método permite obtener la lista de todos los elementos hijos con `tagName = tagName` del elemento `elem`. Este método se utilizó en la clase `XMLToContext` donde se transforma un documento XML en un objeto `Context`.

2. `DocumentUtils`:

Esta clase consta de un solo método, el cual permite crear un objeto `Document` desde un archivo que puede ser su ruta como `String`, o un objeto `java.io.File`.

3. Utils:

Clase cuyos métodos realizan acciones básicas, como obtener el día de hoy en un formato definido y el quitar los caracteres conflictivos de un string, como por ejemplo, los caracteres con tildes y ñ.

4. idGenerator:

Se utiliza esta clase para generar un String aleatorio que sirva como id para los elementos de los archivos XMI generados.

5. XMLToContext:

Esta clase consta de un solo método, el cual sirve para crear un objeto Contexto desde un archivo XML. Este método es utilizado en ambos módulos de la aplicación. Para el módulo Modelo de Contexto, se utiliza cuando el usuario carga un archivo XML en la pantalla de inicio con un modelo de contexto ya definido. Este método permite crear el objeto Contexto desde este archivo, para poder modificar este objeto en la siguiente interfaz. Para el módulo Contexto Concreto se utiliza siempre, ya que en la pantalla de inicio el usuario debe cargar un archivo XML con el modelo de contexto definido de forma obligatoria. El método se usa de la misma forma que para el módulo anteriormente descrito.

```
public static Context createContextFromXML(String fileSrc)
```

Este método recibe como parámetro la ruta del archivo cargado por el usuario (XML). Este archivo lo transforma en un objeto Document, para luego poder obtener todos sus Element. Teniendo Todos los objetos Element es posible construir un objeto Contexto que luego será modificado por el usuario.

6. XMLToLink:

Esta clase consta de un solo método, el cual sirve para crear un objeto Link desde un archivo XML. Este método es utilizado en ambos módulos de la aplicación. Para el módulo Modelo de Contexto, se utiliza cuando el usuario carga un archivo XML en la pantalla de inicio con un modelo de contexto ya definido. Este método permite crear el objeto Link desde este archivo (siempre y cuando existan Links). Para el módulo Contexto Concreto se utiliza siempre, ya que en la pantalla de inicio el usuario debe cargar un archivo XML con el modelo de contexto definido de forma obligatoria. El método se usa de la misma forma que para el módulo anteriormente descrito.

```
public static Link createLinkFromXML(String fileSrc)
```

Este método recibe como parámetro la ruta del archivo cargado por el usuario (XML). Este archivo lo transforma en un objeto Document, para luego poder obtener todos los Element cuyo tagName sea "Link". Con esto es posible crear el objeto Link que contendrá una lista de Strings con todos los links de prioridad creados por el usuario.

7. FileUtils:

Esta clase se creó por el hecho de que los archivos XML y XMI son prácticamente iguales, con la única diferencia de que los archivos XMI todos sus elementos poseen una id. Entonces para crear un archivo XML o XMI desde un objeto Context, los métodos de las clases ContextToXML y ContextToXMI eran prácticamente iguales. Por lo tanto esta clase, se creó a modo de parametrizar los métodos de las clases antes mencionadas, para así factorizar código.

Todos los métodos que son utilizados en ambas clases poseen un parámetro booleano (Boolean isXMI) que permite identificar si el mensaje viene desde la clase ContextToXML o ContextToXMI.

```
public static Element createRootElement(Document doc, Context context, Boolean isXMI)
```

Este método crea la raíz del documento doc, la cual dependerá de si isXMI es verdadero o falso, ya que si es un archivo XMI (isXMI = verdadero), al elemento raíz se le asignan más atributos además del nombre y la descripción. Estos son:

- xmi:id, cuyo valor es un String aleatorio generado por una instancia de idGenerator.
- xmi:version, es la versión del documento xmi, la cual se asigna a 2.0
- xmlns:xmi, sitio web <http://www.omg.org/XMI>
- xmlns:spcm, sitio web <http://contextmetamodel/1.0>

```
public static void createDocumentElements(Context context, Document doc, Element rootElement, Boolean isXMI, Map<String, String>attributesAndIds, Map<String, String>attributesValuesAndIds)
```

Método que crea todos los objetos Element que serán hijos de la raíz rootElement del documento doc. Estos objetos son: myDimensions, myContextAttributes y posibleValues. Si el parámetro isXMI es verdadero, los atributos attributesAndIds y attributesValuesAndIds serán utilizados, de ser false, estos parámetros vendrán null. Los parámetros recién mencionados se utilizan para asociar los atributos myContextElement y myContextAttributeValue de los objetos Element “contextAttributeConfiguration” a las ids de los objetos Element “myContextAttributes” y “posibleValues” respectivamente.

```
public static void createLinksElements(Document doc, Element rootElement, Link links, Boolean isXMI)
```

Método que crea todos los objetos Element “Links” de los archivos XML y XMI. Los objetos Element Links son creados siempre y cuando El objeto links tenga una lista de links no vacía. De ser vacía, los archivos carecerá n de este tipo de Element. La única diferencia que se genera entre los archivos dependiendo del valor del parámetro isXMI, es que si isXMI es verdadero, los Element Links tendrán una id, si es falso, no la tendrán.

```
public static void createProjectSpecificElements(Context context, Document doc,
Element rootElement, Map<String, String>attributesAndIds,
Map<String, String>attributesValuesAndIds)
```

Método exclusivo de la clase ContextToXML, ya que permite crear el objeto Element “myContextConfigurations”, el cual tendrá como hijos a todos los objetos Element “contextAttributeConfiguration”. Estos últimos son todos los atributos con su valor asignado en el módulo Contexto Concreto. Los atributos que se le asignan a los objetos “contextAttributeConfiguration” son:

- xmi:id, cuyo valor es un String aleatorio generado por una instancia de idGenerator.
- name, valor del nombre del atributo.
- description, valor de la descripción del atributo.
- myContextElement, corresponde al xmi:id del atributo del modelo de contexto al que representa.
- myContextAttributeValue, corresponde al xmi:id del posible valor de atributo del modelo de contexto que representa.

8. ContextToXML y ContextToXMI:

Clases cuya función es crear un archivo XML y un archivo XMI respectivamente desde un objeto Context. Esta labor la realizan en base a un método, el cual usa principalmente todos los métodos descritos en la clase FileUtils.

4.6.4. Formularios Dinámicos en las Interfaces

En el módulo Modelo de Contexto fue necesario crear formularios dinámicos para permitir que el usuario pudiera agregar nuevas dimensiones, atributos y posibles valores de atributos. Para lograr este dinamismo, se crearon dos *scripts JavaScript: contextModel.js y createLinks.js*. Ambos archivos utilizan la librería jQuery 2.8. Estos *scripts* poseen funciones que permiten que la aplicación sea robusta, ya que aquí se definen todas las validaciones que mantienen la consistencia y coherencia de los datos en la aplicación.

1. contextModel.js:

Este *script* contiene la totalidad de funciones que permiten el dinamismo en la segunda interfaz del módulo Modelo de Contexto, además de todas sus validaciones y efectos visuales. En esta interfaz, es donde el usuario puede: agregar, editar y eliminar; dimensiones, atributos y posibles valores de atributos. Gracias a las funciones implementadas en este archivo, fue posible crear una interfaz usable e intuitiva para el usuario.

2. createLinks.js

Script exclusivo de la tercera interfaz del módulo Modelo de Contexto; contiene sólo dos funciones que se detallan a continuación.

- `$("#linksForm").submit(function()`

Función cuya labor es mostrar un diálogo que permita al usuario confirmar la creación de los *links* de prioridad.

- `$(".checkLinks").click(function()`

Función que evita la creación de *links* inconsistentes, es decir, no permite crear *links* en ambas direcciones. Por ejemplo: Si el atributo uno tiene prioridad sobre el atributo dos, esta función no permitirá crear el *link* tal que el atributo dos tenga prioridad sobre el atributo uno.

Esta función se creó para aumentar la robustez en la aplicación, evitando inconsistencias e incoherencia en los datos generados.

Finalmente, la interacción de las vistas, con los controladores, y estos con las entidades dentro de la aplicación, se ilustran en la Figura 11

4.7. Cómo la solución planteada resuelve el problema

Como se mencionó en la especificación del problema, en el capítulo 3, el gran problema que existe al momento de configurar un contexto es la herramienta que se utiliza hoy en día, siendo una herramienta muy específica (EMF), la cual requiere de conocimientos previos para poder operar con ella.

La solución propuesta resuelve el problema, ya que se realizó pensando en una herramienta fácil de usar por parte de los usuarios finales.

El primer usuario será un Ingeniero de Procesos quién deberá definir el modelo de contexto de la empresa utilizando el módulo Modelo de Contexto de la aplicación. El segundo usuario, será un Jefe de Proyecto que solamente sabe de las características del proyecto que va a dirigir, quien basado en la información que maneja, podrá configurar el contexto concreto del proyecto, utilizando el módulo Contexto Concreto de la aplicación.

La configuración del contexto concreto depende de la definición previa del modelo de contexto. Es decir, el trabajo en la aplicación por parte del Jefe de Proyecto depende del trabajo realizado por el Ingeniero de Procesos. Por esto, es importante destacar que ambos serán miembros de una misma empresa. De esta forma ambos manejarán la misma jerga, y el Jefe de Proyecto no tendrá problemas al momento de configurar el contexto concreto, debido a que no existirán problemas con los conceptos empleados por el Ingeniero de Procesos en la definición del modelo de contexto.

Finalmente el flujo que debe seguir la aplicación para poder crear el archivo XMI con el contexto concreto configurado se ilustra en la Figura 12

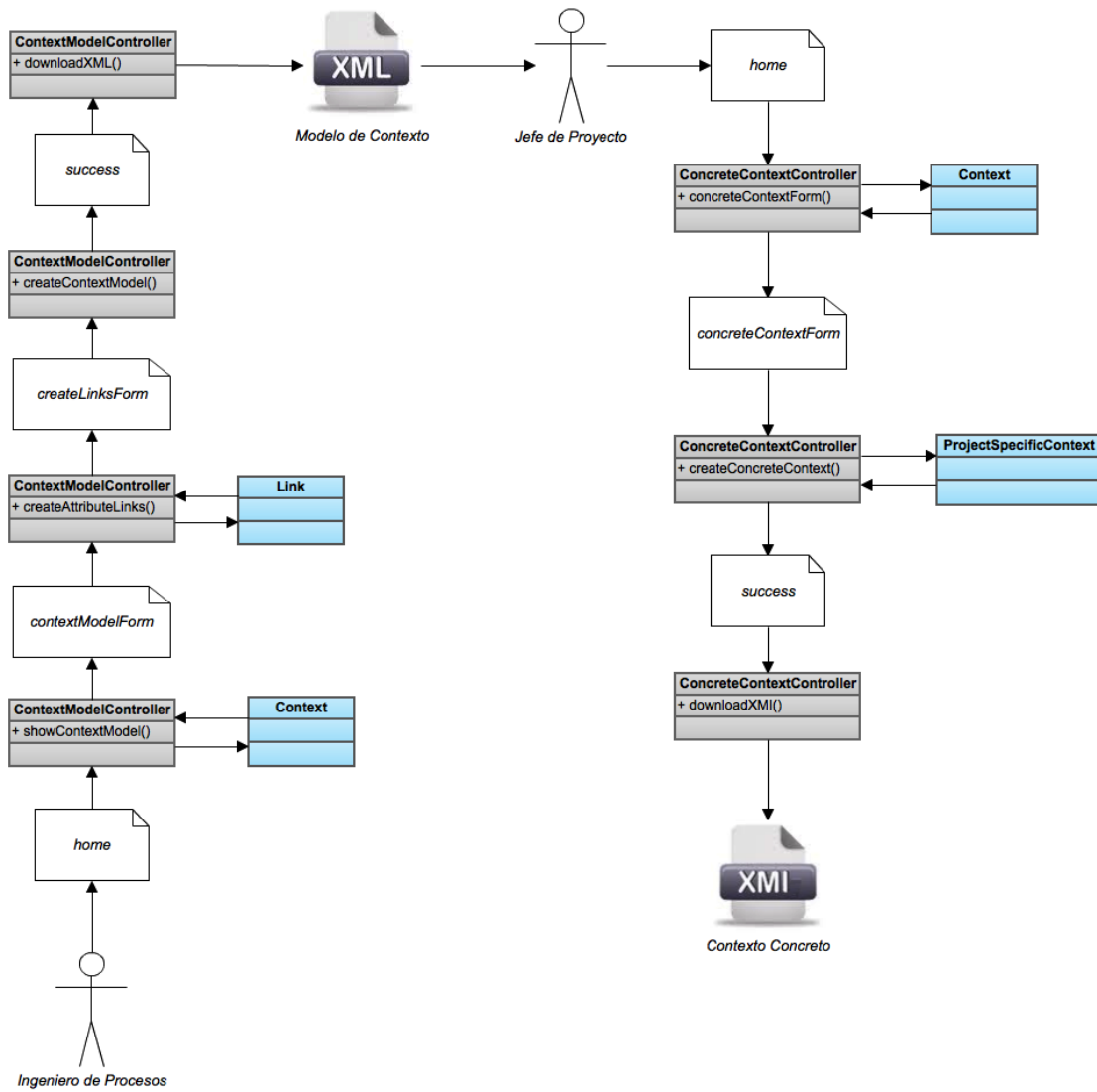


Figura 11: Diagrama que ilustra la interacción de todas las vistas, controladores y entidades de la aplicación.

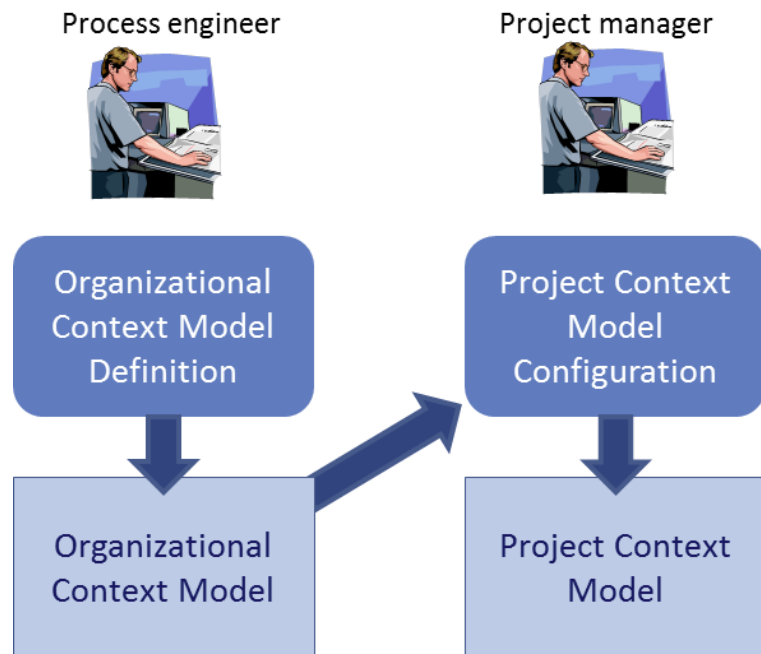


Figura 12: Diagrama del flujo que sigue la aplicación para crear el contexto concreto. Se comienza con la definición del modelo de contexto organizacional por parte de un Ingeniero de Procesos. Una vez definido, un Jefe de Proyecto configura el modelo de contexto para obtener finalmente el contexto concreto para un proyecto de software específico.

5. Funcionamiento de la aplicación paso a paso

5.1. Modelo de Contexto

5.1.1. Inicio

La interfaz de inicio, sirve para darle la bienvenida al usuario y explicarle cuál es el primer paso a realizar para poder iniciar la definición del modelo de contexto. Aquí el usuario tiene dos alternativas para cargar un modelo de contexto inicial, el cual servirá como guía para la definición del modelo de contexto definitivo. Una alternativa es cargar un archivo XML, con el formato detallado en la sección 4.5 “Diseño de archivos que almacenan la información procesada”. Esta opción se creó, ya que el Ingeniero de Procesos puede tener en poder un archivo XML previamente definido con un modelo de contexto, distinto al modelo canónico, que coincida de mejor manera con el modelo de contexto que se quiera definir, por lo tanto esto llevaría al usuario a realizar menos trabajo para lograr su objetivo.

La segunda opción es presionar directamente en el botón siguiente, cargandose automáticamente el modelo de contexto canónico definido en la sección 4.2.

A continuación se ilustra la interfaz de inicio del módulo Modelo de Contexto.



Figura 13: Interfaz inicio módulo Modelo de Contexto

5.1.2. Definición del modelo de contexto

La siguiente interfaz, es donde se define el modelo de contexto propiamente tal. Esta interfaz consta de 3 secciones principales:

- **Panel Modificación de Datos Modelo de Contexto:**

En este panel es posible modificar el nombre y la descripción del nuevo modelo de contexto que se definirá. Esta acción se realiza presionando el ícono de edición. Una vez hechos los

cambios, se presiona el ícono para guardar. La figura 14 ilustra el panel recién descrito.



Figura 14: Interfaz panel modificación datos modelo de contexto módulo Modelo de Contexto

- **Panel Administrador de Dimensiones:**

Aquí es donde el usuario puede seleccionar la dimensión que desee modificar. Además en este panel es donde el usuario puede agregar una nueva dimensión, haciendo click en el botón agregar. También es posible eliminar una dimensión existente presionando en el ícono de papelerita al lado del nombre de la dimensión que se desee eliminar. Además en este panel, se encuentra el botón que permitirá crear el modelo de contexto, con las nuevas definiciones realizadas por el usuario. La figura 15 ilustra el panel de dimensiones.

- **Panel Modificación Totalidad de datos Dimensión:**

Este panel es el más importante dentro de esta vista, ya que aquí es posible modificar todos los datos relacionados con una dimensión. Estos datos son: el nombre, la descripción, agregar, eliminar o modificar atributos, de los cuales a su vez es posible modificar sus nombres, descripciones y agregar, eliminar o modificar los posibles valores que puedan tener estos atributos. En este panel se utiliza la misma iconografía o una similar a la que presentan los paneles descritos anteriormente. Para la eliminación se tiene una papelerita, o una cruz roja. Para la edición y guardado de los datos se utilizaron los íconos más utilizados para estas acciones (libreta con lápiz para edición y un *diskette* para guardar). En este panel los valores existentes por defecto vienen inhabilitados, es decir, no es posible su edición. Si el usuario desea editar los valores de la dimensión seleccionada, deberá hacer click en el ícono de edición. Luego de haber editado los datos a gusto, estos se deben guardar. No será posible modificar otra dimensión si no se han guardado los cambios realizados en la dimensión actual. Todo lo descrito sobre este panel, se ilustra a través de la Figura 16.

Finalmente en la Figura 17 es posible visualizar la interfaz completa, captando todos los módulos recién descritos, en donde se definen el nombre y descripción del modelo de contexto, además de las dimensiones, atributos y posibles valores de atributos.



(a) Panel administrador de dimensiones



(b) Panel administrador de dimensiones al hacer click en agregar

Figura 15: Panel administrador de dimensiones módulo Modelo de Contexto

5.1.3. Creación de links de prioridad entre atributos

En la interfaz de creación de links o enlaces de prioridad entre los atributos del modelo de contexto definido en la vista anterior es posible identificar tres secciones definidas.

- **Recuadro de reglas:**

Esta sección se realizó para explicar al usuario cómo opera la grilla de checkboxes, qué representa y de qué forma se configuran los links de prioridad mediante esta. Si el texto no es lo suficientemente claro, el usuario puede hacer click en el link “Ejemplo” y este desplegará una imagen que ilustra un ejemplo concreto de la utilización de la grilla y su significado (Figura 18).

- **Lista de atributos existentes en el modelo de contexto:**

Los números que rodean la grilla, tanto verticales (rojos), como horizontales (negros), repre-

Dimensión Proyecto

Nombre: 

Descripción:

Atributos **Agregar**

Nombre:

Descripción:

Posibles valores **Agregar**

Valor: 

Valor: 

Valor: 



Nombre:

Descripción:

Posibles valores **Agregar**

Valor: 


Valor: 

Valor: 



(a) Panel administrador de todos los valores de la dimensión seleccionada

Dimensión Proyecto

Nombre: 

Descripción:

Atributos **Agregar**

Nombre:


Descripción:


[Agregar](#) [Cerrar](#)


Nombre:


Descripción:

Posibles valores **Agregar**

Valor: 

Valor: 

Valor: 



(b) Campos para agregar un nuevo atributo a la dimensión al hacer click en agregar

Nombre:

Descripción:

Posibles valores **Agregar**

Valor:

[Agregar](#) [Cerrar](#)

Valor: 

Valor: 

Valor: 



(c) Campo para agregar un nuevo posible valor de un atributo de la dimensión seleccionada

Figura 16: Panel administrador de todos los valores de la dimensión seleccionada módulo Modelo de Contexto



Figura 17: Interfaz definición modelo de contexto módulo Modelo de Contexto

Links de prioridad

Reglas para configuración de links

La grilla consta de una columna de números rojos que representa a los atributos que tendrán la prioridad sobre la columna de números negros, que serán los atributos sobre los cuales se tendrá prioridad.

Pincha en el siguiente enlace para ver un ejemplo: [Ejemplo](#)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Dada la configuración de la imagen, el atributo (1) tiene prioridad sobre los atributos: (2), (4), (6), (7), (8)

Figura 18: Ejemplo de cómo opera la grilla

sentan los atributos del modelo de contexto. La lista de atributos del panel de la izquierda permite al usuario identificar cuál es la relación entre el número y el atributo.

- **Grilla de checkboxes:**

Esta sección es la más importante de la presente vista, ya que aquí es donde se le permite al usuario crear los links o enlaces de prioridad entre atributos. El uso de esta grilla es explicado en la sección del recuadro de reglas. La grilla permite relacionar todos los atributos entre sí, menos consigo mismos. Además cabe destacar, que la grilla no permite crear links en ambas direcciones para dos atributos. Es decir, si se tiene que el atributo (1), tiene prioridad sobre el atributo (2), automáticamente se deshabilita la opción de seleccionar la caja que me permita crear el link inverso, es decir, que el atributo (2) tenga prioridad sobre el atributo (1). Esto se ilustra en la Figura 19.

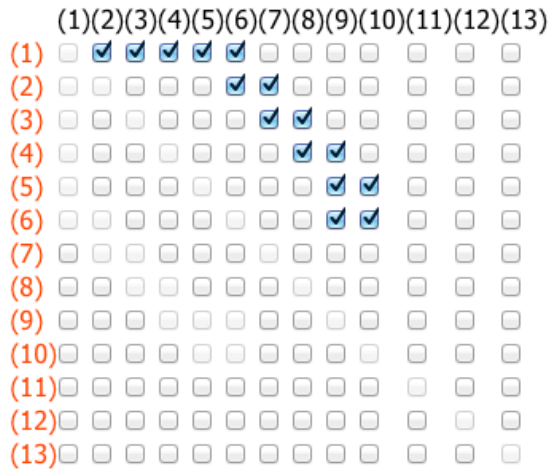


Figura 19: Grilla con checkboxes seleccionados

Finalmente en la Figura 20 se muestra en su totalidad la interfaz que permite la creación de los links o de prioridad.

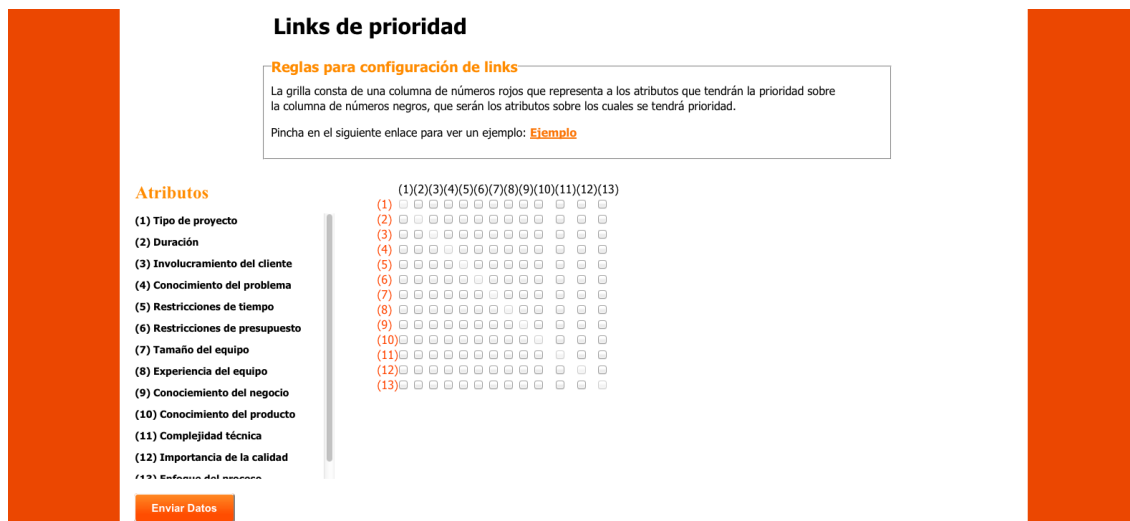


Figura 20: Interfaz creación de links entre atributos módulo Modelo de Contexto

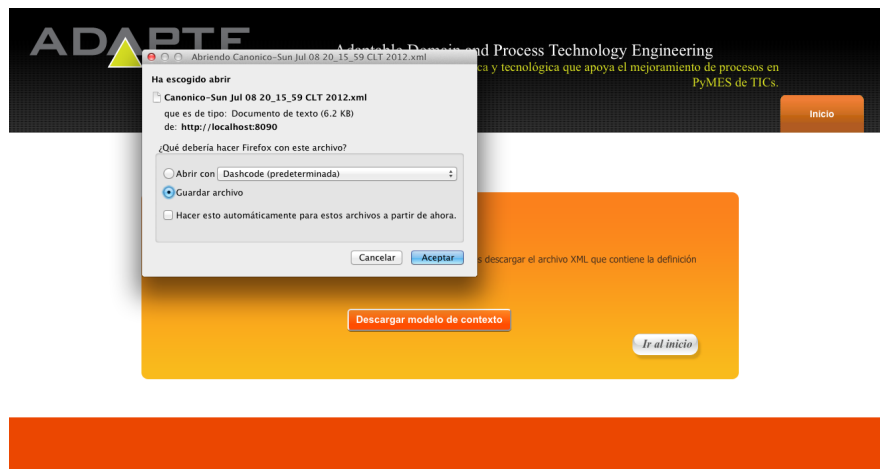
5.1.4. Creación exitosa y descarga de los resultados

Esta interfaz es la última vista del módulo Modelo de Contexto. Esta vista es muy importante, ya que además de informar al usuario de la creación exitosa del nuevo modelo de contexto que definió, aquí es donde se descarga el archivo XML con el modelo de contexto. Es importante destacar que esta es la única instancia que el usuario tiene para obtener el archivo XML, vale decir, si por alguna razón se cerró el navegador, o se refrescó la página, el modelo de contexto se pierde y se debe realizar todo el proceso de nuevo. También es bueno mencionar que el usuario al momento de llegar aquí,

puede descargar el archivo XML cuantas veces desee. Este archivo será creado con el nombre del modelo de contexto que se asignó y la fecha del día en que se definió el modelo de contexto. Lo descrito anteriormente, se puede visualizar en la Figura 21.



(a) Interfaz final



(b) Interfaz final con diálogo de descarga del archivo XML

Figura 21: Interfaz final módulo Modelo de Contexto

5.2. Contexto Concreto

5.2.1. Inicio

La interfaz de inicio sirve para dar la bienvenida al usuario al módulo Contexto Concreto y explicarle cuál es el primer paso a realizar para poder iniciar la configuración del contexto concreto. Aquí el usuario tiene sólo una alternativa para continuar con el siguiente paso. El usuario debe cargar un modelo de contexto definido en un archivo XML. Este archivo es el que se descargará al finalizar el módulo Modelo de Contexto descrito en la sección 5.1. Una vez cargado el archivo, se

debe presionar el botón siguiente, para avanzar a la siguiente interfaz, la cual permitirá configurar el contexto concreto.

A continuación en la Figura 22 se ilustra la interfaz de inicio del módulo Contexto Concreto.

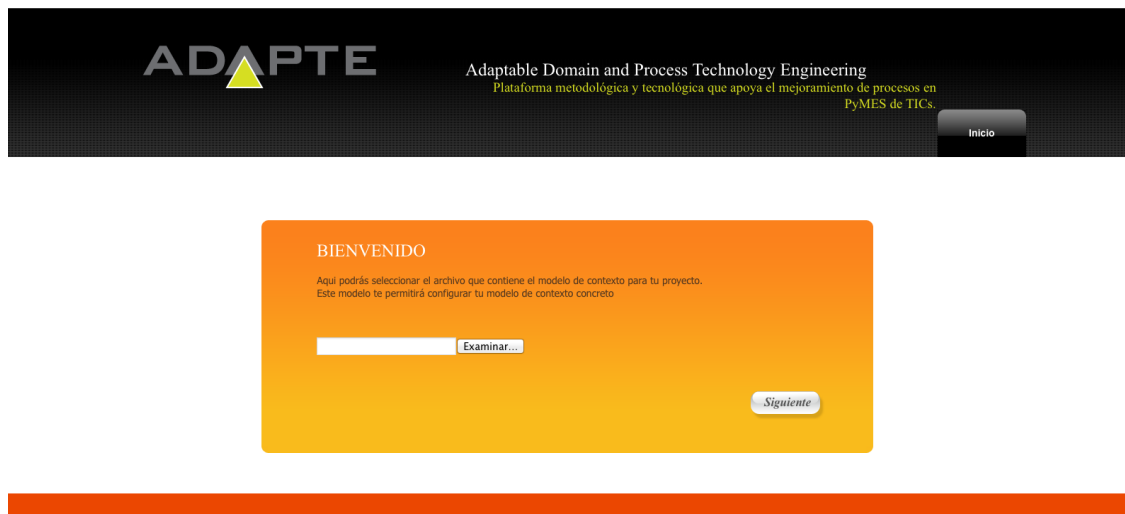


Figura 22: Interfaz inicio módulo Contexto Concreto

5.2.2. Configuración del contexto concreto

En esta interfaz, el usuario debe realizar dos acciones para poder avanzar a la última interfaz de este módulo y poder descargar el archivo XMI, que contenga el contexto concreto configurado.

El usuario, debe ingresar obligatoriamente un nombre que identifique al nuevo contexto concreto que se configurará. Agregar una descripción de este contexto es opcional.

Luego el usuario pasa a realizar la configuración del contexto propiamente tal. Se despliegan todos los atributos del modelo de contexto categorizados por su dimensión. Al lado de cada atributo hay un selector que le permite al usuario elegir un valor entre todos los posibles valores definidos en el modelo de contexto.

Una vez asignados todos los valores a sus respectivos atributos, el usuario debe presionar el botón crear contexto concreto para poder avanzar a la última interfaz. La Figura 23 ilustra la interfaz que se acaba de describir.

5.2.3. Creación exitosa y descarga de los resultados

Esta interfaz, es la última vista del módulo Contexto Concreto. Esta vista es muy importante, ya que además de informar al usuario de la creación exitosa del nuevo contexto concreto que se acaba de configurar, aquí es donde se descarga el archivo XMI con el contexto concreto. Es importante destacar que esta es la única instancia que el usuario tiene para obtener el archivo XMI, vale decir, si por alguna razón se cerró el navegador, o se refrescó la página, el contexto concreto se pierde y se debe realizar todo el proceso de nuevo. También es bueno mencionar que el usuario al momento

ADAPTE Adaptable Domain and Process Technology Engineering
 Plataforma metodológica y tecnológica que apoya el mejoramiento de procesos en PyMES de TICs.

Inicio Definir Contexto

Modelo de contexto: Canónico

Contexto Concreto

Nombre:

Descripción:

Proyecto

Tipo de proyecto: nuevo desarrollo

Duración: corto

Involucramiento del cliente: alto

Conocimiento del problema: claro

Restricciones de tiempo: muy restringido

Restricciones de presupuesto: muy restringido

(a) Interfaz configuración contexto concreto

Equipo

Tamaño del equipo: muy restringido

Experiencia del equipo: alta

Conocimiento del negocio: conocido

Conocimiento del producto: conocido

Producto

Complejidad técnica: alta

Importancia de la calidad: alta

Proceso

Enfoque del proceso: producto final

Configurar contexto concreto

(b) continuación interfaz configuración contexto concreto

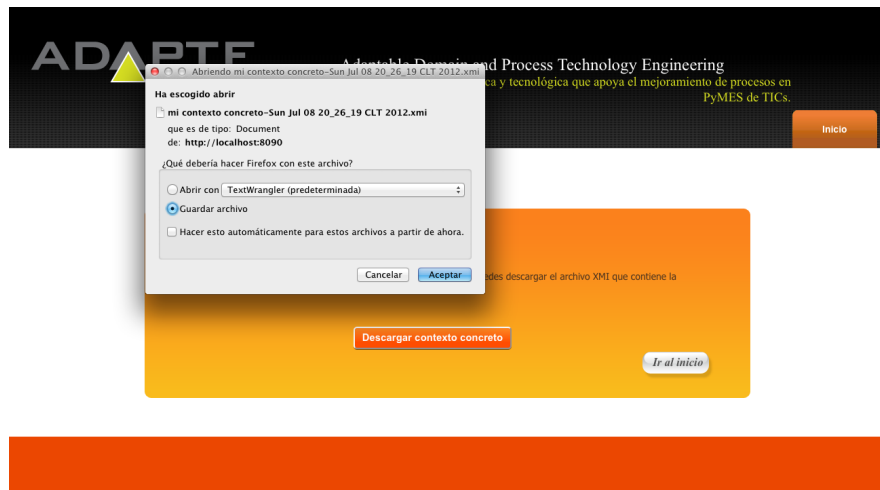
Figura 23: Interfaz configuración contexto concreto módulo Contexto Concreto

de llegar aquí, puede descargar el archivo XMI cuantas veces desee.

Este archivo será creado con el nombre del contexto concreto que se definió en la interfaz anterior y la fecha del día en que se configuró el contexto concreto. Lo descrito anteriormente se puede visualizar en la Figura 24.



(a) Interfaz final



(b) Interfaz final con diálogo de descarga del archivo XMI

Figura 24: Interfaz final módulo Contexto Concreto

6. Validación de la solución

En la sección anterior se explicó y se realizó una demostración de cómo funciona la aplicación, validando los requerimientos funcionales de la solución. Con esto es claro que la solución desarrollada cumple con lo necesario para afirmar que la aplicación funcionalmente está completa.

Para la verificación y la validación de los requisitos no funcionales de la solución se realizaron pruebas de usabilidad y de robustez. A continuación se presentan en detalle estos tópicos.

6.1. Usabilidad en la Aplicación

La usabilidad en la aplicación fue un punto fundamental para el desarrollo de la solución, ya que una aplicación que hace correctamente lo que debe hacer pero no es usable simplemente no sirve.

El análisis se enfoca en el usuario final, en especial en lo intuitivo del uso de la aplicación, permitiéndole encontrar lo que necesita y manejarse con todas las opciones de la aplicación. También es muy importante tener una interfaz que satisfaga al usuario. Por esto es que se midieron estos dos atributos generales: el Aprendizaje y la Satisfacción.

Para medir los atributos mencionados anteriormente se utilizaron los siguientes métodos de usabilidad:

1. Se emplea una entrevista, la cual se realiza al finalizar un plan de navegación por la aplicación, y tendrá una pauta semi-abierta (es decir, se tiene una lista de preguntas permitiendo ahondar en una de estas si es que la situación lo amerita). Esto permite obtener una impresión de la satisfacción del usuario con el uso de la aplicación y en general obtener una medida del otro atributo.
2. Un cuestionario usuario final que permite obtener información respecto a los atributos mencionados anteriormente, en especial la satisfacción.

6.1.1. Muestra

Como el perfil del usuario final es bastante específico, sólo se obtuvo una muestra de 4 personas por módulo de la aplicación. Se cree que la muestra es suficiente para lograr resultados confiables por el hecho de que la aplicación no es muy grande, y el universo de usuarios finales muy pequeño y específico.

6.1.2. Instrumentos

Para la entrevista se usó una pauta semi-abierta de entrevista, es decir, se llevan preguntas y se permite ahondar en un aspecto. La pauta de entrevista se encuentra en el Anexo C y permite obtener datos cualitativos (mediante preguntas abiertas) y datos cuantitativos, permitiéndole al usuario calificar de 1 a 7 según la escala Likert ciertos aspectos de la aplicación. Finalmente se utilizó un cuestionario de usuario final que está en el Anexo D y permite al usuario calificar ciertos

aspectos de usabilidad en la escala 1 a 5. Finalmente permite que el usuario califique en la misma escala la aplicación de manera global y justifique esta respuesta.

6.1.3. Tareas

1. Tareas a realizar por el usuario Ingeniero de Procesos:

- Acceder al formulario del modelo de contexto desde la pantalla de inicio.
- Modificar el nombre y descripción del modelo de contexto.
- Agregar una nueva dimensión.
- Eliminar una dimensión.
- Seleccionar una nueva dimensión.
- Modificar el nombre y descripción de una dimensión.
- Agregar un nuevo atributo.
- Eliminar un atributo de una dimensión.
- Agregar un nuevo posible valor de un atributo.
- Eliminar un posible valor de un atributo.
- Crear links entre atributos.
- Descargar el archivo con el modelo de contexto.

2. Tareas a realizar por el usuario Jefe de Proyectos:

- Cargar correctamente el archivo que contiene el modelo de contexto.
- Agregar nombre y descripción al contexto concreto.
- Asignar un valor para cada atributo del modelo de contexto.
- Crear el contexto concreto.
- Descargar el archivo con el contexto concreto.

6.1.4. Resultados Entrevista

La entrevista consta de una parte cuantitativa (Ver Anexo C) cuyos resultados son expuestos a continuación:

	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Promedio
¿Qué tan fácil fue para usted agregar una nueva dimensión?	7	7	7	7	7
¿Qué tan fácil fue para usted eliminar una dimensión ?	7	7	7	7	7
¿Qué tan fácil fue para usted agregar un nuevo atributo ?	7	5	6	5	5,75
¿Qué tan fácil fue para usted eliminar un atributo?	6	7	7	5	6,25
¿Qué tan fácil fue para usted agregar un nuevo posible valor a un atributo?	7	5	6	6	6
¿Qué tan fácil fue para usted eliminar un posible valor de un atributo?	7	7	7	7	7
¿Qué tan fácil fue para usted modificar los datos de una dimensión?	7	7	7	7	7
¿Qué tan fácil fue para usted crear nuevos links entre atributos del modelo de contexto?	4	7	7	6	6
¿Qué tan fácil fue para usted descargar el archivo del modelo de contexto?	7	7	7	7	7
En general ¿Está usted satisfecho con la aplicación?	6	7	7	6	6,5

(a) Entrevista módulo Modelo de Contexto

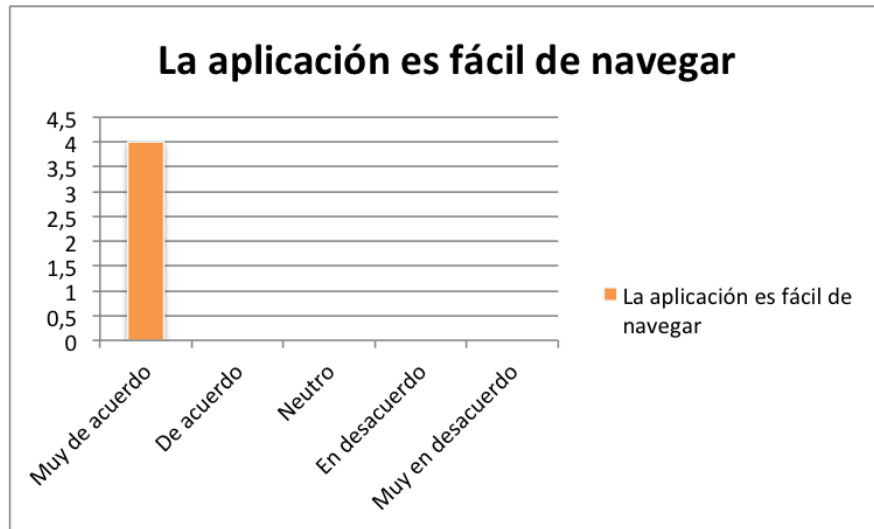
	Usuario 1	Usuario 2	Usuario 3	Usuario 4	Promedio
¿Qué tan fácil fue para usted definir un nombre y descripción para el contexto concreto?	7	7	7	7	7
¿Qué tan fácil fue para usted asignar los valores de los atributos ?	7	7	7	7	7
¿Qué tan fácil fue para usted descargar el archivo del contexto concreto ?	7	7	7	7	7
En general ¿Está usted satisfecho con la aplicación?	7	7	7	7	7

(b) Entrevista módulo Contexto Concreto

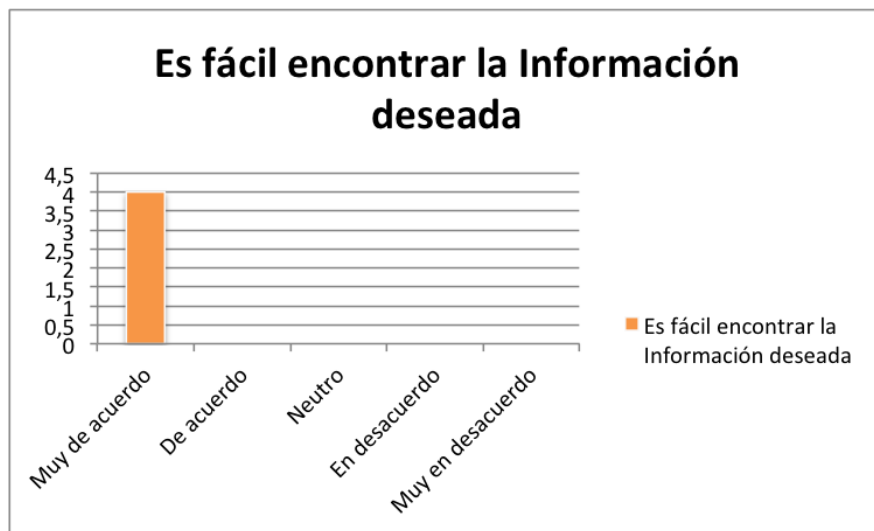
Figura 25: Resultados entrevista

6.1.5. Resultados Cuestionario del Usuario Final

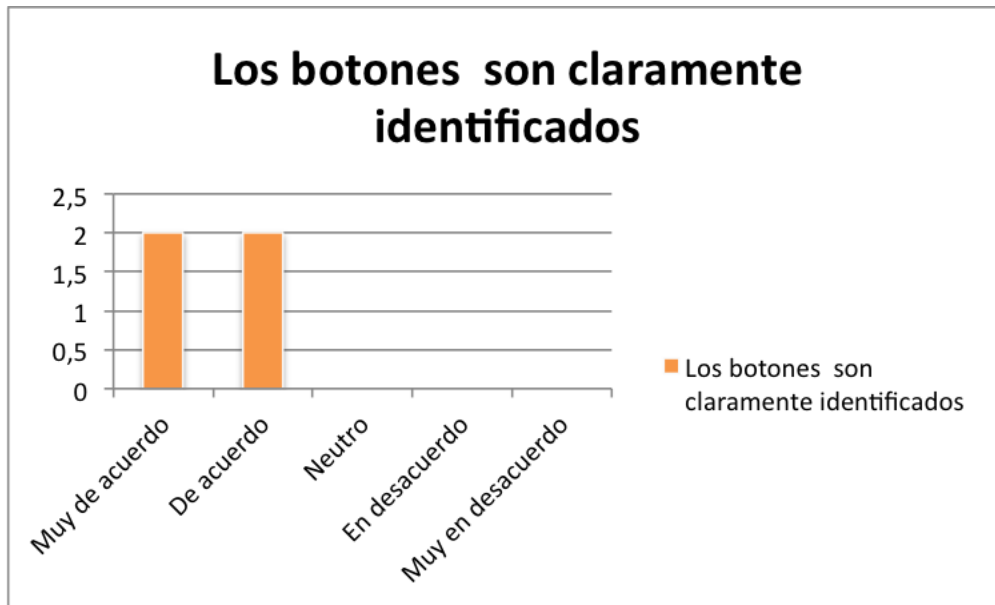
A continuación se mostrarán los resultados obtenidos por cada pregunta del cuestionario usuario final (Ver anexo D). Las respuestas de los usuarios se ilustran a través de los siguientes gráficos.



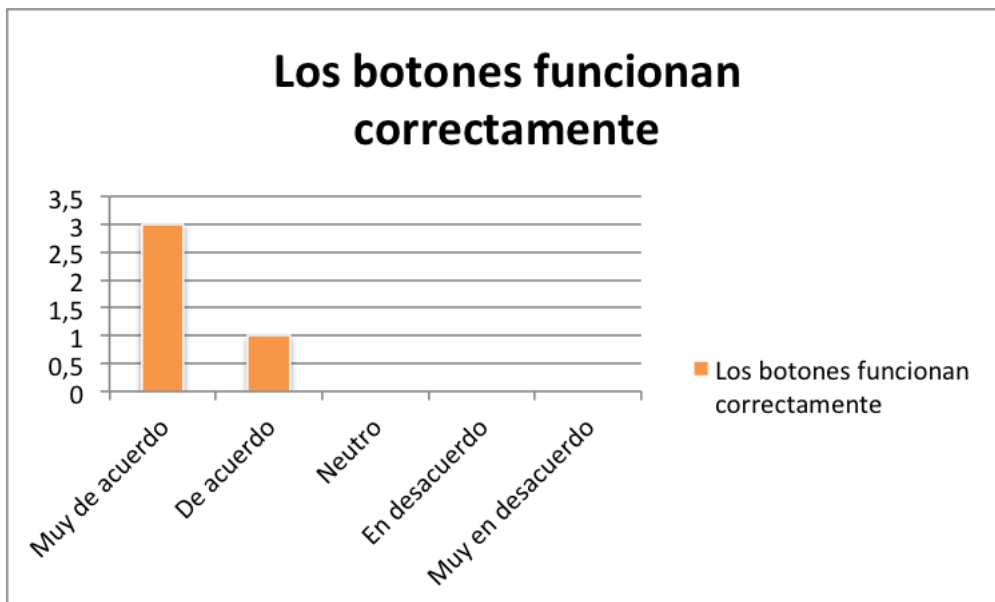
(a) La aplicación es fácil de navegar



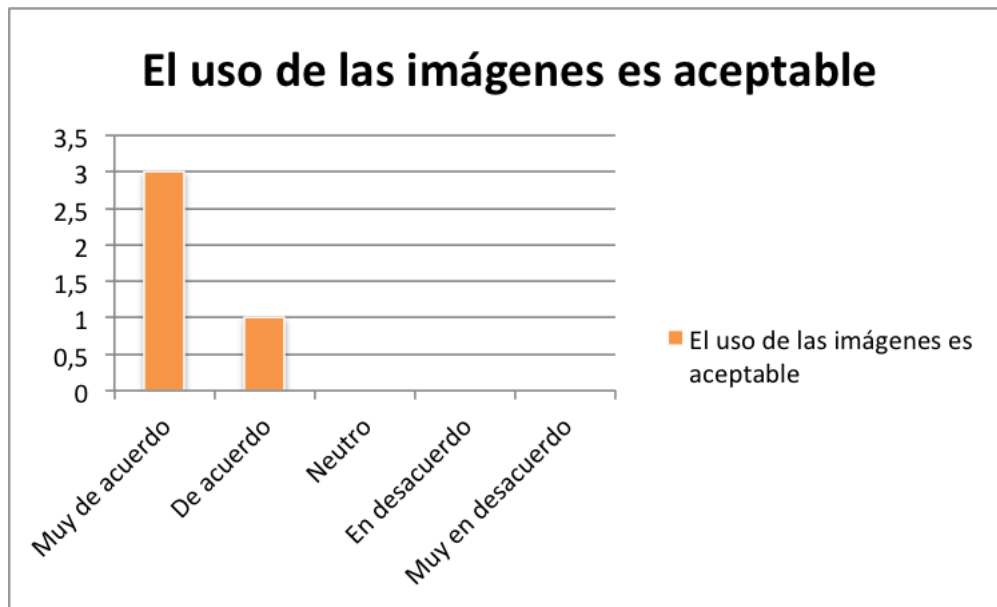
(b) Es fácil encontrar la información deseada



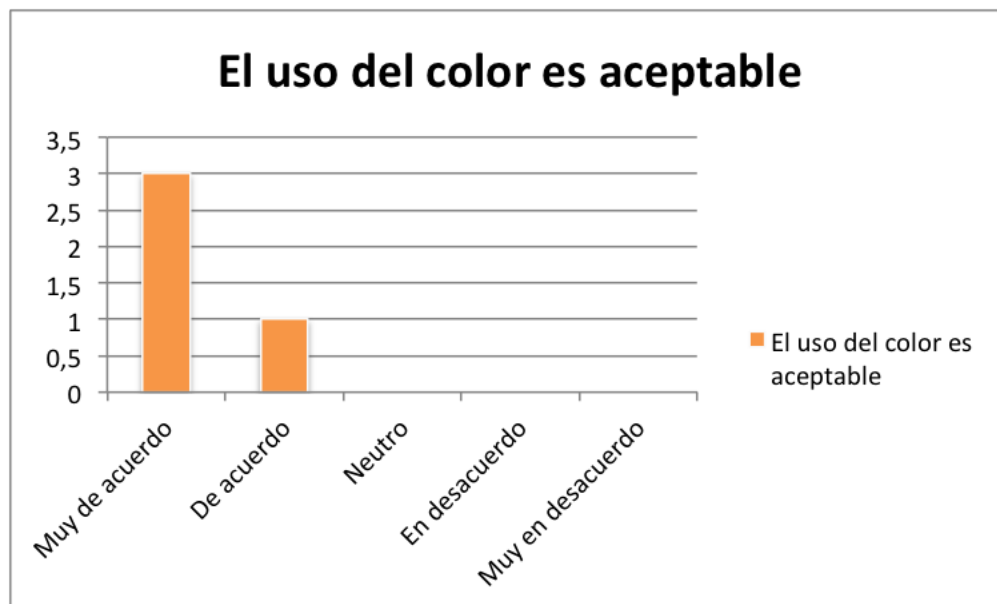
(c) Los botones son claramente identificados



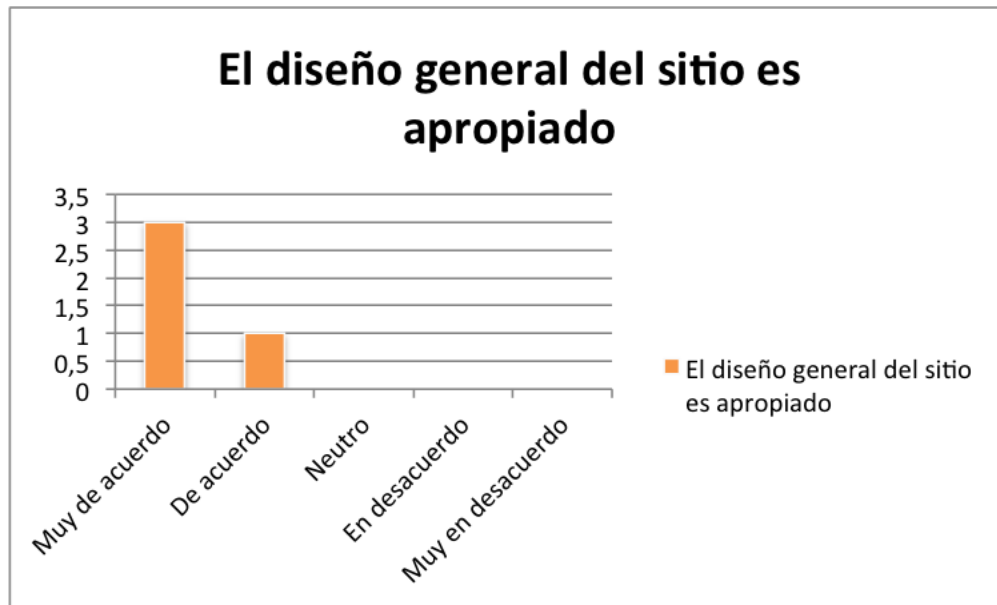
(d) Los botones funcionan correctamente



(e) El uso de las imágenes es aceptable



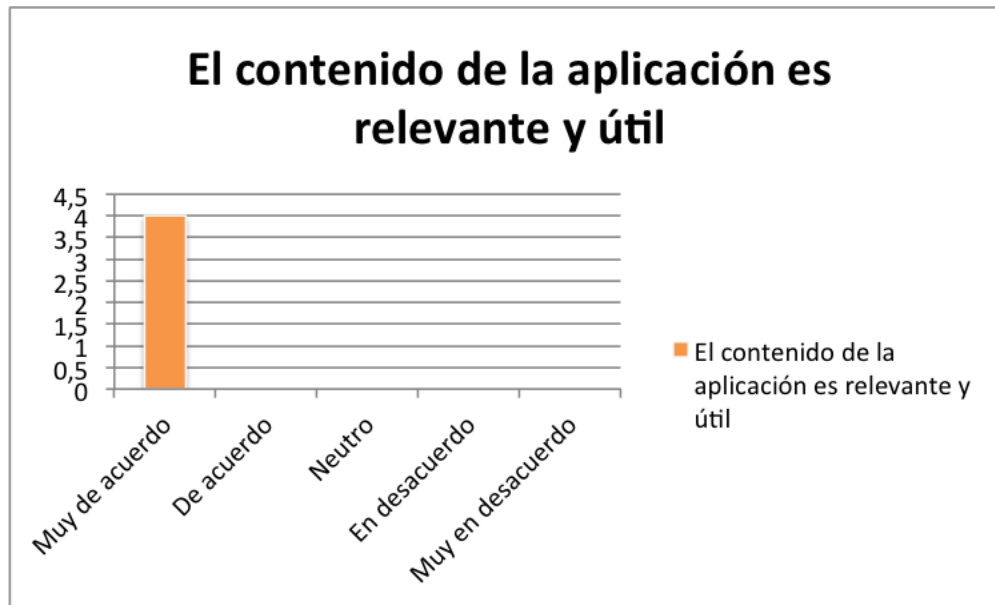
(f) El uso del color es aceptable



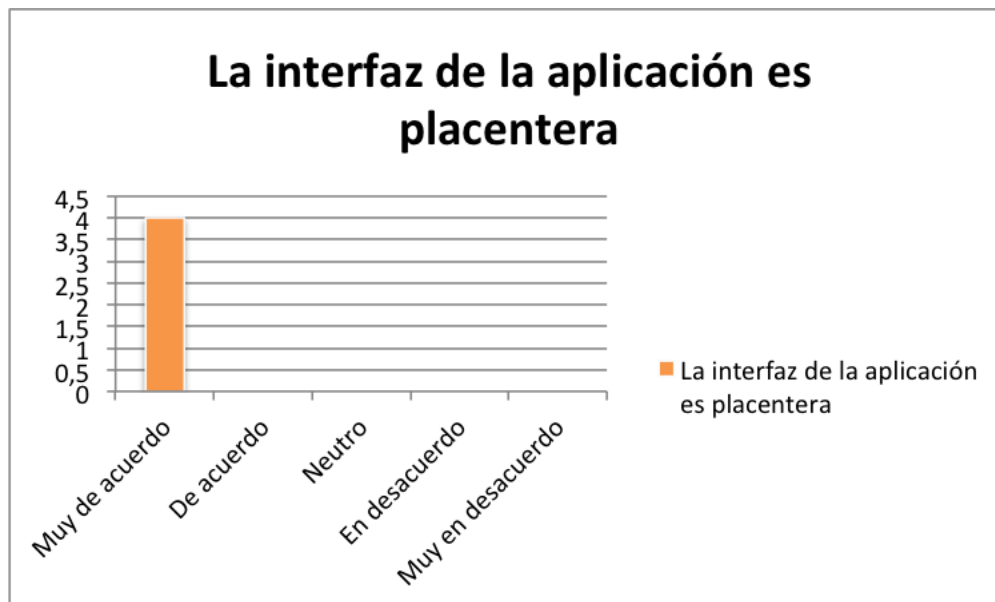
(g) El diseño general de la aplicación es apropiado



(h) La organización de la información de la aplicación es apropiada



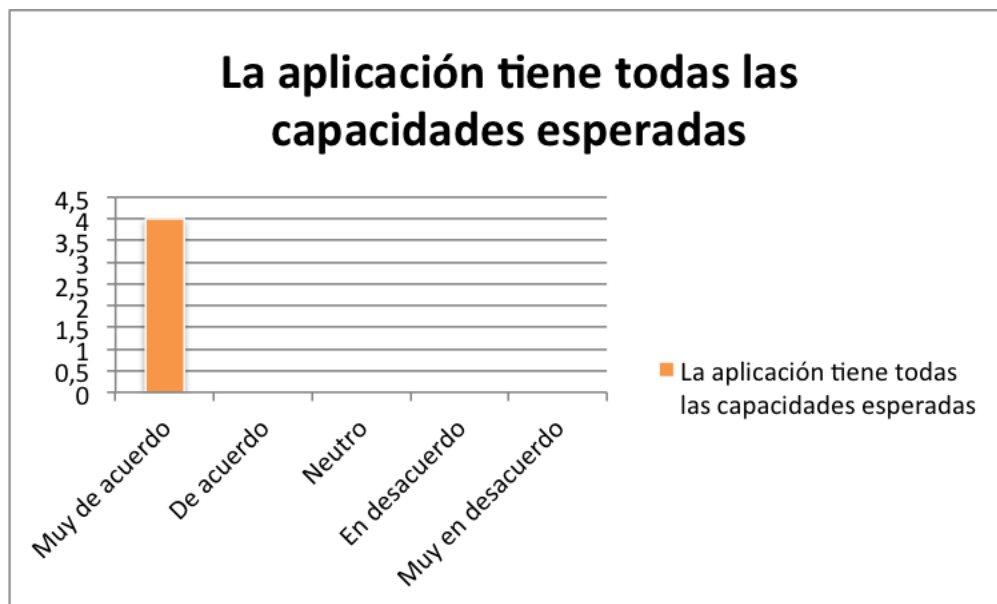
(i) El contenido de la aplicación es relevante y útil



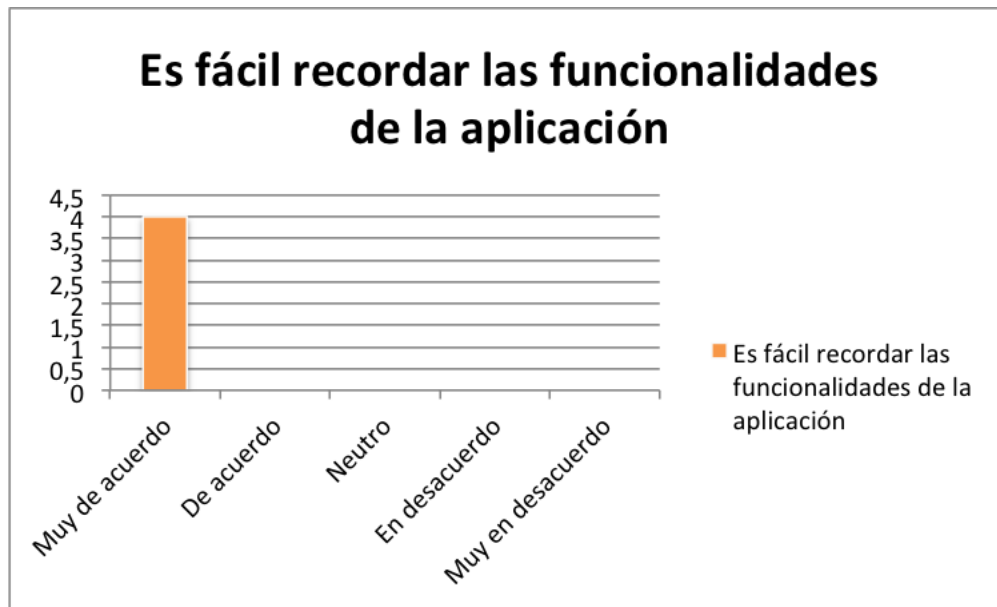
(j) La interfaz de la aplicación es placentera



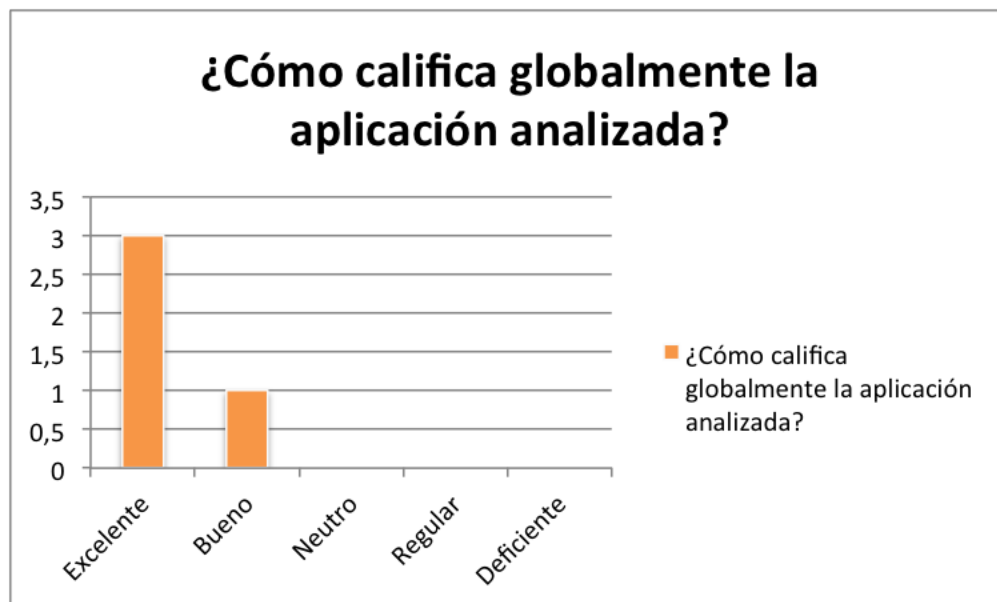
(k) La aplicación tiene todas las funcionalidades esperadas



(l) La aplicación tiene todas las capacidades esperadas



(m) Es fácil recordar las funcionalidades de la aplicación



(n) Calificación global de la aplicación

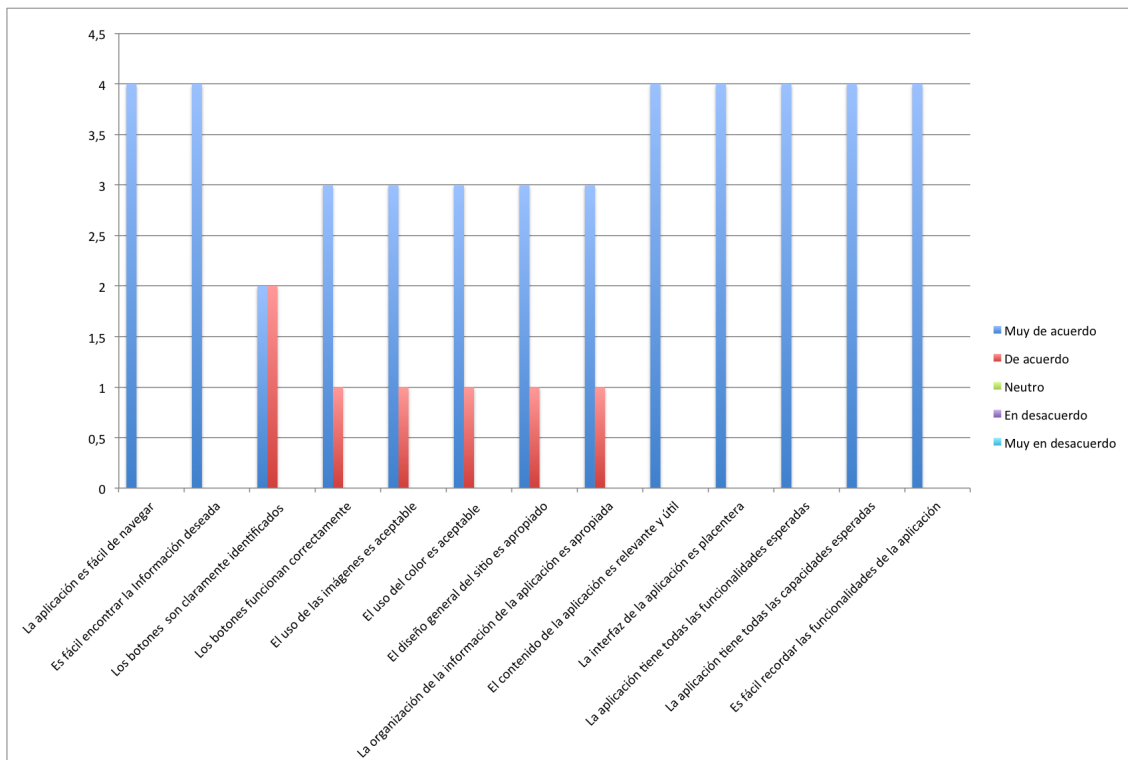


Figura 26: Respuestas a todas las preguntas del cuestionario usuario final

6.1.6. Análisis de los resultados obtenidos

1. Entrevista:

La información cuantitativa que se obtuvo en la entrevista aparece en la Figura 25. Entre toda la información obtenida destaca el alto nivel de aceptación de la aplicación por parte del usuario final. Ambos módulos de la aplicación tuvieron muy alta calificación, destacando la calificación del módulo Contexto Concreto, en donde se obtuvo la calificación máxima en todas las preguntas de la entrevista. No así el módulo Modelo de Contexto, que si bien tuvo muy buenas calificaciones, existen puntos en los que hay que mejorar. Hay que destacar que el módulo Contexto Concreto tiene un grado de complejidad menor, tanto en cantidad de etapas por las que hay que pasar para llegar al resultado final, como a la información que se debe mostrar y solicitar ingresar al usuario.

En general los usuarios quedaron muy satisfechos con la aplicación, y esto se ve reflejado en la última pregunta de la entrevista en cada módulo, donde se pregunta por el nivel de satisfacción por parte del usuario hacia la aplicación, la cual para la aplicación Modulo de Contexto fue un 6.5 y en la aplicación Contexto Concreto un 7. Ambas en una escala de 1 a 7.

La información cualitativa permite ampliar más en detalle las virtudes y defectos de la aplicación.

- Respecto a la organización de la información, los usuarios se mostraron satisfechos:
 - *“Me parece adecuada y eficiente, intentando maximizar el tiempo que una persona debe invertir”*
 - *“Me parece bastante buena, ya que permite al usuario tener toda la información necesaria sin abrir nuevas ventanas ni cambiarse a nuevas páginas”*
- Respecto a lo que le gustaría que tuviese la aplicación en este momento, existieron opiniones variadas, algunas complejizando un poco más la solución:
 - *“Tal vez sería útil reorganizar los elementos (en particular la sección “dimensiones” de “atributos”, ya que “atributos” está bien con respecto a “valores”) en la aplicación del ingeniero de procesos, debido a que, como están todos los elementos en la misma sección, uno podría pensar que están en el mismo orden jerárquico.”*
 - *“Tal vez organizar las dimensiones en tablas”*

2. Cuestionario Usuario Final:

Con respecto al cuestionario usuario final, es posible visualizar un alto número de respuestas positivas (Ver Figura 26). Esto concuerda con los resultados obtenidos en la entrevista de la sección anterior.

Se visualiza que las respuestas de los usuarios varían entre “muy de acuerdo” y “de acuerdo”, lo cual muestra el alto grado de satisfacción por parte de los usuarios hacia la aplicación. Esto permite alcanzar uno de los objetivos más importante al momento de medir la usabilidad de una aplicación, ya que un usuario que no está a gusto y no se siente cómodo navegando o realizando tareas en una aplicación web lo más probable es que no la use o falle cuando la utilice.

Otros de los puntos importante en temas de usabilidad que era deseable lograr, era el crear una aplicación intuitiva, fácil de manipular y muy importante, fácil de aprender. El aprendizaje en la usabilidad es muy importante, ya que si una aplicación tiene un alto nivel de aprendizaje, las tareas que se deban desempeñar en esta, cada vez se harán de forma más rápida y eficiente. Es posible inferir el alto nivel de aprendizaje en la aplicación por las buenas calificaciones obtenidas en preguntas como:

- La aplicación es fácil de navegar
- Es fácil encontrar la información deseada
- Es fácil recordar las funcionalidades de la aplicación

Una de las preguntas más importante, que resume la usabilidad global de la aplicación y la aceptación por parte del usuario final, es la pregunta: ¿Cómo califica globalmente la aplicación analizada?, donde 3 de los 4 usuarios la encontraron excelente y uno la encontró buena.

Algunas justificaciones de las calificaciones obtenidas por los usuarios que demuestran el éxito de la aplicación en temas de usabilidad son:

- *“La aplicación en líneas generales funciona correctamente, tiene una interfaz grata y amigable. Es eficiente y, salvo el problema del botón de edición (del cual se puede llegar a cuestionar su existencia), es bastante fluido y limpio. En general el aprendizaje suele ser instantáneo, basta hacer una dimensión para saber manera por completo la aplicación.”*
- *“La aplicación es bastante amigable, con iconografía reconocible y funciones intuitivas de realizar, como el editar, guardar, eliminar. Su aprendizaje es muy rápido, ya que siempre tiene un mismo comportamiento. Interfaz amena. Buen uso de los colores.”*

6.2. Robustez en la Aplicación

Al igual que la usabilidad, la robustez en la aplicación resultó ser un tema importante al momento de desarrollar la solución. Con la robustez es posible asegurar que la aplicación sabrá como responder de la mejor manera ante situaciones críticas durante el funcionamiento de esta. Una aplicación robusta permite obtener los resultados esperados al momento de finalizar las tareas realizadas en la aplicación, es decir, entregar al usuario un resultado correcto y consistente con lo que él esperaba.

El mayor esfuerzo en lograr una aplicación robusta se realizó en uno de los módulos de la aplicación, el módulo Modelo de Contexto. Debido a que en este módulo el usuario tiene la libertad de realizar muchas tareas del estilo “eliminar” y “editar”. Si no existe una correcta respuesta por parte de la aplicación al momento de ejecutar una de estas tareas, se pueden generar situaciones no esperadas por parte del usuario.

Las acciones realizadas para entregar mayor robustez a la aplicación se detallan a continuación.

6.2.1. Diálogos de confirmación al momento de eliminar

Al momento de intentar eliminar cualquier elemento dentro de la definición del modelo de contexto, la aplicación envía un mensaje de confirmación, permitiendo al usuario confirmar su decisión, o recular si presionó un botón de eliminación por equivocación, o simplemente si se arrepintió.

Esta confirmación es muy importante, ya que acciones como las de eliminar no pueden realizarse sin una previa confirmación, ya que puede generar inconsistencia en el resultado para el usuario si es que algo se eliminó por equivocación sin el consentimiento del usuario.

A continuación, en la Figura 27 se ilustran los diálogos para la situación recién descrita.

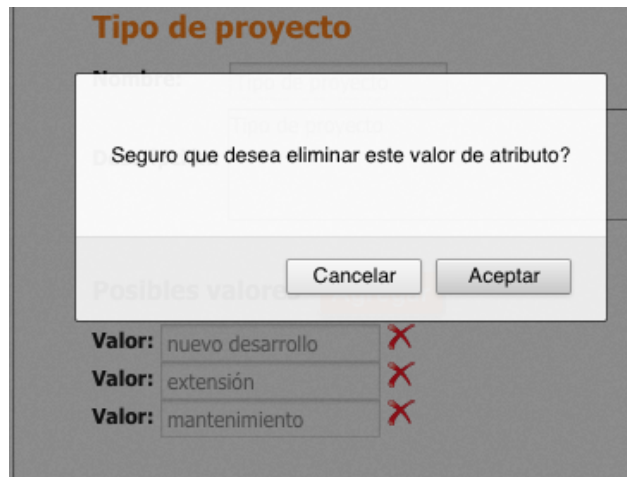
6.2.2. Diálogo de confirmación al momento de crear el modelo de contexto

Cuando el usuario presiona el botón para crear el modelo de contexto, antes de crearlo recibe un mensaje de confirmación. Esto es muy importante, ya que este tipo de botones, los cuales envían los datos de un formulario al servidor, como es el caso del botón que crea el modelo de contexto, con el simple hecho de presionar “Enter” ejecutan la acción de enviar el formulario. Entoces puede ocurrir que por accidente el usuario presione la tecla “Enter” o simplemente haya apretado el botón sin la intención de crear el formulario, generando molestia en el usuario, ya que de volver atrás deberá empezar la definición del modelo de contexto de nuevo.



(a) Mensaje de confirmación al momento de eliminar una dimensión

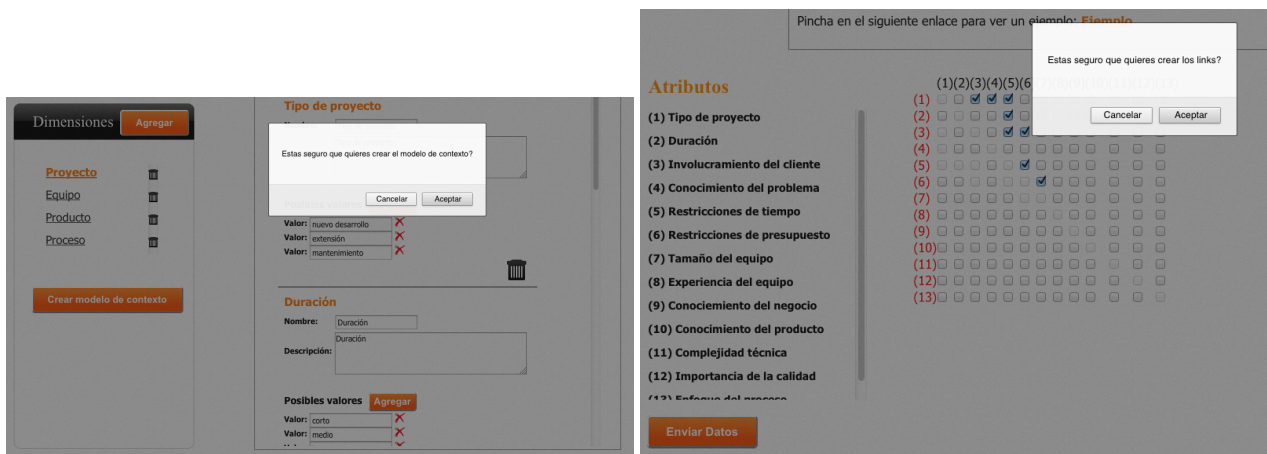
(b) Mensaje de confirmación al momento de eliminar un atributo de una dimensión



(c) Mensaje de confirmación al momento de eliminar un posible valor de un atributo

Figura 27: Mensajes de confirmación antes de eliminar una Dimensión, Atributo o un Posible Valor

Esta misma confirmación se aplica en la sección de creación de links de prioridad. A continuación, en la Figura 28 se ilustran los diálogos de confirmación para la situación recién descrita.



(a) Mensaje de confirmación al momento de crear el Modelo de Contexto

(b) Mensaje de confirmación al momento de crear los Links de Prioridad

Figura 28: Mensajes de confirmación antes de enviar los formularios al servidor

6.2.3. Diálogo informativo que no permite la eliminación de todas las dimensiones, atributos o posibles valores

Para la correcta creación de un modelo de contexto es necesario que exista al menos una dimensión, que a su vez tenga al menos un atributo, y de la misma forma este atributo debe tener al menos un posible valor.

Para mantener la correctitud y coherencia del modelo de contexto que se creará, al momento en que el usuario intente eliminar en su totalidad cualquiera de los elementos anteriormente mencionados, éste recibirá un diálogo de alerta explicándole que debe existir al menos un elemento del que intenta eliminar.

A continuación, en la Figura 29 se ilustran los diálogos de confirmación para la situación recién descrita.

6.2.4. Diálogo informativo que no permite tener valores repetidos en las dimensiones, atributos o posibles valores

Finalmente, la última medida que garantiza la robustez de la aplicación, es evitar la repetición de algún elemento que pueda generar inconsistencias o incoherencias en el modelo de contexto. Lo que no se permite repetir es:

- No pueden existir 2 dimensiones con el mismo nombre.
- No pueden existir 2 atributos con el mismo nombre dentro de una misma dimensión.
- No pueden existir 2 posibles valores con el mismo valor para un mismo atributo.

Esta validación se realiza al momento de presionar el botón guardar dentro de la modificación de una dimensión. Al existir cualquiera de los casos descritos anteriormente, se arrojará un diálogo

de alerta que informe al usuario del elemento repetido. Estos mismos casos se vuelven a validar, al momento de presionar el botón que permite crear el modelo de contexto. Esto se hizo de esta manera, ya que puede darse el caso en que el usuario en la última dimensión modificada se le olvidó presionar el botón guardar y aún así presionó el botón de creación del modelo de contexto.

A continuación en la Figura 30 se ilustran estos tres casos, mostrando sus respectivos diálogos.

6.3. Contexto Concreto Generado (XMI)

Para que la solución desarrollada sea realmente útil para el proyecto ADAPTE, es necesario que el archivo de salida de la aplicación, específicamente del módulo Contexto Concreto, se genere correctamente, ya que como se explicó en secciones anteriores del presente informe, este archivo será el que en conjunto con otro archivo XMI, serán sometidos al proceso de adaptación.

El archivo XMI debe tener un formato específico, el cual es explicado en la sección 4.5.

Si el archivo XMI está bien generado, éste podrá ser visualizado mediante una estructura de árbol a través de la herramienta EMF 2.3.

6.3.1. Demostrando el correcto formato del archivo XMI mediante la herramienta EMF

Para la siguiente demostración, primero se creó un modelo de contexto; este modelo está basado en el modelo de contexto canónico al cual se le eliminaron 2 dimensiones, dejando sólo las dimensiones Proyecto y Equipo.

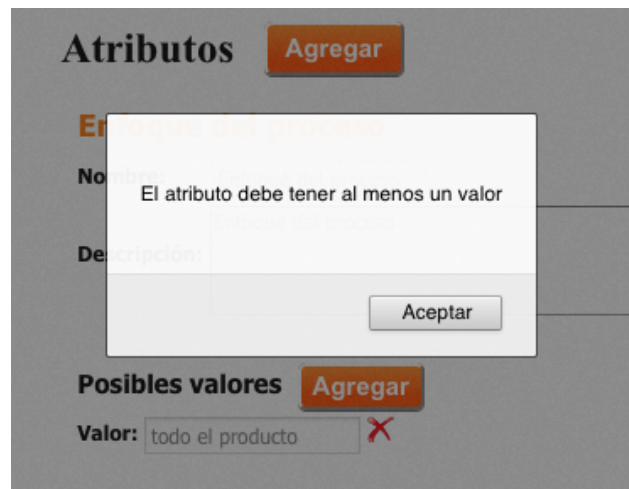
Una vez generado el archivo con el modelo de contexto, éste se utilizó para configurar el contexto concreto. Como resultado se obtuvo el archivo XMI con el contexto concreto llamado ContextoConcretoEjemplo.

A continuación en la Figura 31 se ilustra un archivo XMI que contiene un contexto concreto generado con la aplicación, visualizado mediante una estructura de árbol.



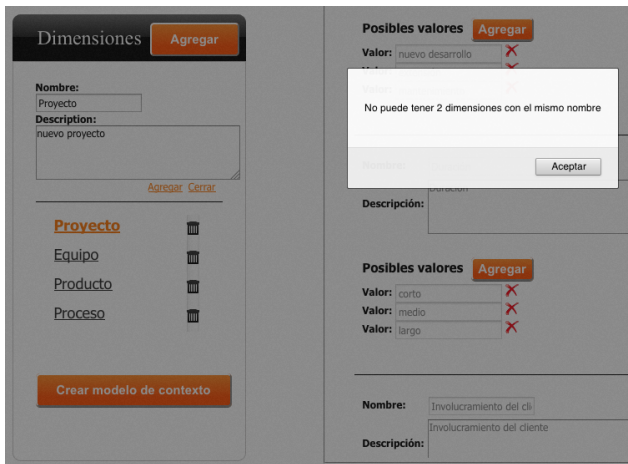
(a) Mensaje informativo de que al menos debe existir una dimensión

(b) Mensaje informativo de que al menos debe existir un atributo en todas las dimensiones

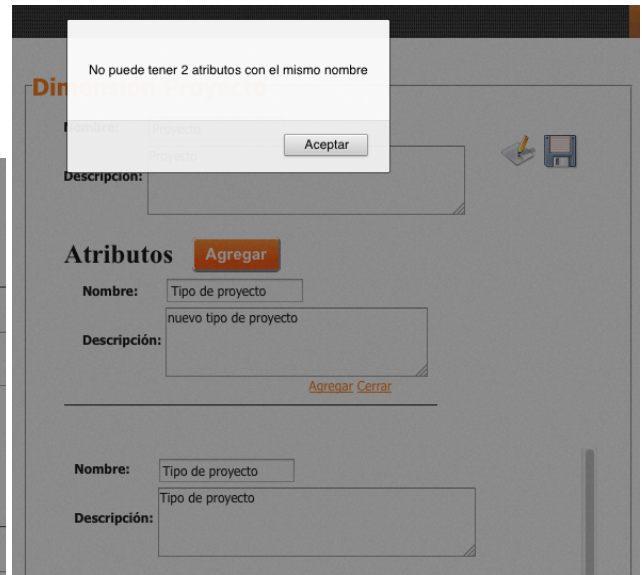


(c) Mensaje informativo de que al menos debe existir un posible valor para todos los atributos

Figura 29: Mensajes informativos, indicando que deben existir al menos una Dimensión, un Atributo y un Posible Valor



(a) Mensaje que informa al usuario que existen 2 dimensiones con el mismo nombre.

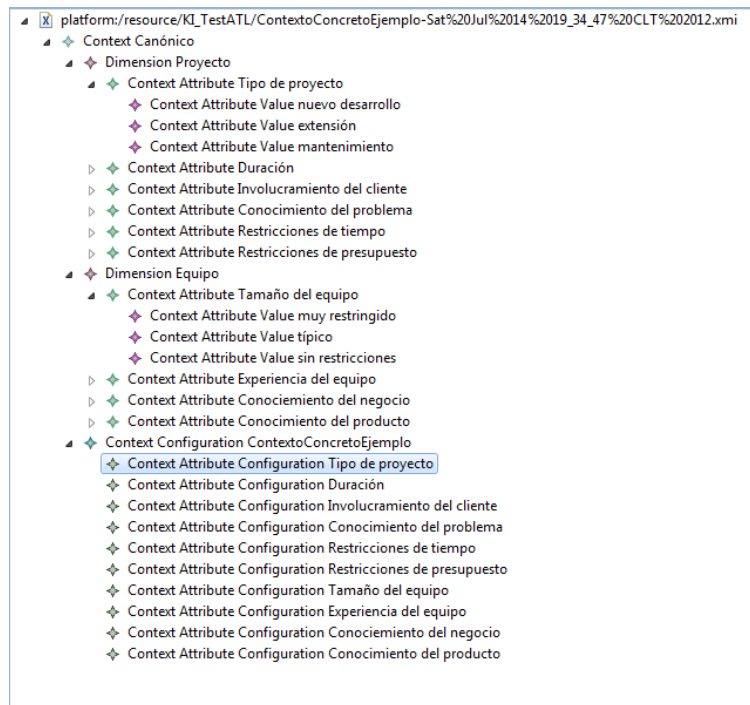


(b) Mensaje que informa al usuario que existen 2 atributos dentro de una misma dimensión con el mismo nombre.

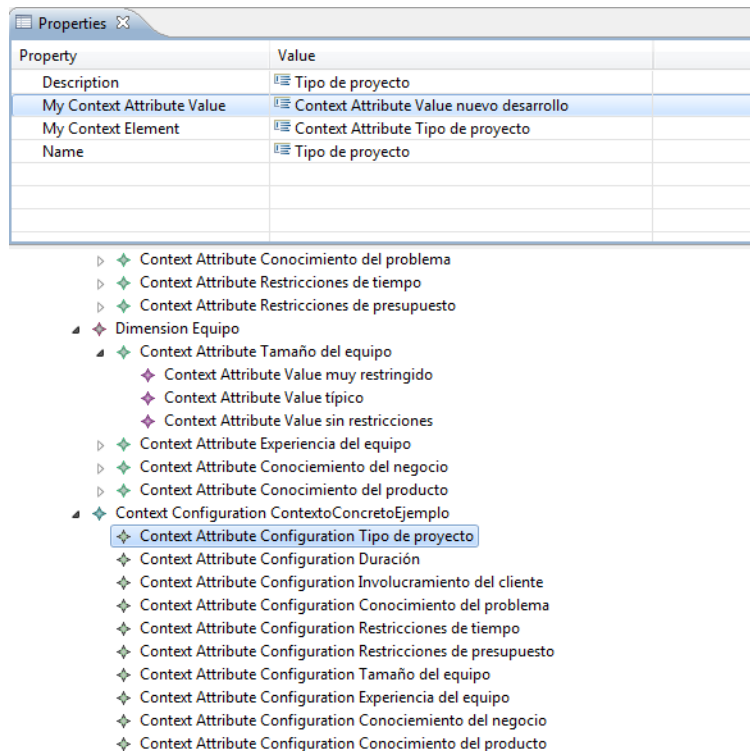


(c) Mensaje que informa al usuario que existen 2 posibles valores dentro de un mismo atributo con el mismo valor.

Figura 30: Mensajes informativos, indicando al usuario la existencia de elementos con el mismo nombre.



(a) Contexto concreto en archivo XMI representado mediante una estructura de árbol



(b) Datos que demuestran la correcta asociación entre los elementos del archivo XMI

Figura 31: Demostración de la correcta generación de los archivos XMI en el módulo Contexto Concreto

7. Análisis y Discusión

El trabajo realizado resultó ser bastante exitoso, ya que se logró cumplir con todos los objetivos propuestos. Se tomaron buenas decisiones durante todo el proceso de implementación, las cuales resultaron ser fundamentales al momento de desarrollar la herramienta, ya que la elección de una aplicación web, fue sin duda la mejor para poder obtener resultados favorables en usabilidad.

El manejo de las interfaces en aplicaciones de escritorio generalmente resulta ser bastante rígido, sin muchas librerías que permitan generar un dinamismo y usabilidad, requisitos fundamentales para la aplicación.

Se cree que la programación del modelo de la aplicación posee una buena estructura de clases, permitiendo a la aplicación ser mantenible y extensible. Por parte de la visualización, quizás hubiese sido bueno trabajar en conjunto o haber recibido asesoramiento por parte de algún diseñador para el desarrollo de las interfaces. Esto hubiese permitido hacer un mejor uso de los colores en las visualizaciones, logrando entregar una mejor estética de la aplicación al usuario.

La herramienta cubre todos los objetivos y requerimientos que permiten resolver el problema de la configuración de contextos de proyectos de software específicos. Dado que la problemática a resolver estaba bien definida y acotada, la solución propuesta en términos de implementación no requiere de una pronta evolución. Lo más probable es que sí exista una continua evolución en el tema visual de la aplicación, ya que la retroalimentación obtenida desde el usuario final puede ser de mucha ayuda para mejorar la estética y usabilidad de la aplicación.

Como la herramienta desde un principio se pensó como una aplicación con un alto nivel de usabilidad, el uso de ésta puede ir más allá de los dos roles especificados para el uso de la aplicación. Para que usuarios con otros roles puedan utilizar la aplicación, más que aprender a usar la aplicación, deben entender qué son los contextos, y qué es lo que se quiere lograr con la aplicación, más que aprender a usar la herramienta. Con el conocimiento de los contextos, el uso de la herramienta resulta ser bastante intuitivo.

8. Conclusión

El trabajo realizado, comenzó con la investigación de varios temas desconocidos. Algunas de las incógnitas que se presentaron fueron: “¿qué es el contexto en un proyecto de software?”, “¿cómo se modela un contexto?”, “¿cómo definir los atributos que formarán parte del contexto?”, “¿cómo representar un contexto en un archivo XML/XMI?”. Además de los temas relacionados con el contexto de un proyecto de software, se investigó y estudió el tema de la usabilidad en el desarrollo de software, más específicamente la usabilidad en las interfaces de aplicaciones web. También se estudió la herramienta EMF; ésta se utilizó para validar que los archivos XMI obtenidos con la aplicación fueran correctos. Finalmente se investigaron distintos tipos de lenguajes para poder decidir cuál sería la mejor alternativa para desarrollar la solución al problema planteado.

Del desarrollo de la solución, se obtuvieron dos resultados importantes. La aplicación propiamente tal, que consta de dos módulos, los cuales permiten definir un modelo de contexto, y configurar un contexto concreto para un proyecto de software, que serán utilizados por un Ingeniero de Procesos y por un Jefe de Proyecto respectivamente.

El segundo resultado importante, fue el poder crear el archivo XMI requerido que almacena el contexto concreto configurado.

Para el problema en cuestión, fue posible definir objetivos acotados y realistas, lo cual no significó que estuvieran exentos de complejidad. Esto permitió que se fijaran metas definidas y no muy extensas, logrando el cumplimiento de los objetivos en su totalidad, estos eran:

- Desarrollar una interfaz para la definición del modelo de contexto.
- Desarrollar una interfaz para la configuración de contextos concretos.
- Definir el formato de salida del modelo de contexto y luego implementarlo.
- Crear un archivo XMI correcto con el contexto concreto configurado.
- Altos niveles de usabilidad.
- Alto nivel de robustez.

Los objetivos se lograron satisfactoriamente, ya que hoy en día la herramienta funciona y operan correctamente. El formato correcto del archivo XMI, se comprobó con la herramienta EMF, logrando visualizar el contexto mediante una estructura de árbol, además de la correcta asociación entre los atributos del contexto concreto con los atributos del modelo de contexto.

Los objetivos más “subjetivos”, que dependían de la opinión del usuario final se lograron, y esto se demostró en la sección 6 donde se validó la solución. En esta sección se expuso al usuario a una serie de pruebas y encuestas de usabilidad, que al analizar sus resultados, permitió llegar a la conclusión de que la aplicación era suficientemente usable.

Para medir la robustez, se sometió a la aplicación a condiciones que permitieran observar el comportamiento de esta en las condiciones de borde. Los resultados de estas pruebas también se

encuentran en la sección de validación de la solución, las cuales resultaron ser bastante positivas, demostrando que la aplicación quedó con un alto nivel de robustez, asegurando correctos resultados para el usuario.

Se cree que los buenos resultados obtenidos se debieron en gran parte a lo interesante del tema y al impacto que puede tener el proyecto ADAPTE en el desarrollo de software en las PyMEs chilenas. Esto se convirtió en un gran incentivo, ya que el trabajar en un proyecto social, en pro de mejorar la calidad y competitividad del software desarrollado en Chile siempre fue una de las motivaciones por las cuales se ingresó a la carrera de Ingeniería Civil en Computación en la Universidad de Chile.

También se cree que el buen resultado obtenido en el presente trabajo, fue producto de las buenas decisiones tomadas al momento de implementar la solución propuesta, ya que se decidió por un lenguaje conocido, lo cual permitió utilizar de la mejor manera todas sus cualidades y particularidades y además por la decisión de realizar una aplicación web, permitiendo desarrollar las interfaces con toda libertad, logrando una buena estética y altos niveles de usabilidad.

El desarrollo de la herramienta, es un gran aporte para el proyecto ADAPTE, ya que permitirá a usuarios sin conocimientos en el modelamientos de contextos, realizar una labor en el proceso de adaptación, que actualmente es bastante compleja y costosa. Ahora con la herramienta la definición de un modelo de contexto para un proyecto de software y la configuración de un contexto concreto resultarán tareas fáciles y rápidas de realizar, agilizando enormemente el proceso de adaptación.

8.1. Trabajo Futuro

Como trabajos futuros, se considera lo siguiente:

1. Trabajar en conjunto con algún diseñador para mejorar aún más la estética y la usabilidad de las interfaces de la aplicación.
2. Internacionalizar la herramienta, es decir, permitir al usuario seleccionar al inicio de la aplicación si desea trabajar en inglés o español, y quizás agregar algún otro idioma que sea necesario.
3. Implementar los links o enlaces faltantes descritos en la sección 4.2

9. Referencias

- [1] Proyecto ADAPTE
dirección URL: <http://www.adapte.cl/>
- [2] A. Aharoni and I. Reinhartz-Berger. A Domain Engineering Approach for Situational Method Engineering. In Proceedings of the 27th International Conference on Conceptual Modeling, ER'08, pages 455-468. Springer-Verlag, 2008.
- [3] Aharoni, A. and Reinhartz-Berger, I. 2008. A Domain Engineering Approach for Situational Method Engineering. In Proceedings of the 27th international Conference on Conceptual Modeling (Barcelona, Spain, October 20 - 24, 2008). Q. Li, S. Spaccapietra, E. Yu, and A. Olivé, Eds. Lecture Notes In Computer Science, vol. 5231. Springer-Verlag, Berlin, Heidelberg, 455-468.
- [4] A. Cockburn. Agile Software Development. Addison-Wesley. 2002
- [5] Boehm, B.; Clark, B.; Horowitz, E.; Westland, C.; Madachy, R. & Selby, R. Cost Models for Future Software Life Cycle Processes: COCOMO 2.0. 1995.
- [6] Brinkkemper, S., Saeki, M., and Harmsen, F. 1998. Assembly Techniques for Method Engineering. In Proceedings of the 10th international Conference on Advanced information Systems Engineering (June 08 - 12, 1998). B. Pernici and C. Thanos, Eds. Lecture Notes In Computer Science, vol. 1413. Springer-Verlag, London, 381-400.
- [7] Dr. Jaime Sánchez Ilabaca. Usabilidad en Software/Web.
- [8] Eclipse Modeling Framework, Home page
dirección URL: www.eclipse.org/modeling/emf/?project=emf
- [9] GECHS AG y Escuela de Ingeniería de la Universidad de Viña del Mar. Sexta versión del Estudio Diagnóstico de la Industria Nacional de Software y Servicios. 2008.
- [10] I. Mirbel and J. Ralyte. Situational method engineering: combining assembly-based and roadmap-driven approaches. Requirements Engineering, 11(1):58-78, 2006.
- [11] Julio Ariel Hurtado, Sergio F. Ochoa, Alcides Quispe and Cecilia Bastarrica. A Context Modeling Language to Support Tailoring of Software Processes. Computer Science Department, FCFM, University of Chile. December 23, 2011
- [12] jQuery
dirección URL: <http://www.jquery.cl>

- [13] Killisperger, P.; Stumptner, M.; Peters, G.; Grossmann, G. & Stückl, T. Meta Model Based Architecture for Software Process Instantiation International Conference on Software Process, 2009, 63-74.
- [14] Mirbel, I. and Ralyté, J. 2005. Situational method engineering: combining assembly-based and roadmap-driven approaches. *Requir. Eng.* 11, 1 (Dec. 2005), 58-78.
- [15] Sánchez, F. y otros (1998). *Psicología social*. Madrid: McGraw-Hill.

Apéndice

A. Anexo I: Archivo XML. Resultado de la definición de un modelo de contexto en el modulo Modelo de Contexto

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Context description="Canónico" name="Canónico">
  <myDimensions description="Proyecto" name="Proyecto">
    <myContextAttributes description="Tipo de proyecto" name="Tipo de proyecto">
      <possibleValues description="nuevo desarrollo" name="nuevo desarrollo" value="nuevo desarrollo"/>
      <possibleValues description="extensión" name="extensión" value="extensión"/>
      <possibleValues description="mantenimiento" name="mantenimiento" value="mantenimiento"/>
    </myContextAttributes>
    <myContextAttributes description="Duración" name="Duración">
      <possibleValues description="corto" name="corto" value="corto"/>
      <possibleValues description="medio" name="medio" value="medio"/>
      <possibleValues description="largo" name="largo" value="largo"/>
    </myContextAttributes>
    <myContextAttributes description="Restricciones de presupuesto" name="Restricciones de presupuesto">
      <possibleValues description="muy restringido" name="muy restringido" value="muy restringido"/>
      <possibleValues description="típico" name="típico" value="típico"/>
      <possibleValues description="sin restricciones" name="sin restricciones" value="sin restricciones"/>
    </myContextAttributes>
  </myDimensions>
  <myDimensions description="Producto" name="Producto">
    <myContextAttributes description="Complejidad técnica" name="Complejidad técnica">
      <possibleValues description="alta" name="alta" value="alta"/>
      <possibleValues description="media" name="media" value="media"/>
      <possibleValues description="baja" name="baja" value="baja"/>
    </myContextAttributes>
    <myContextAttributes description="Importancia de la calidad" name="Importancia de la calidad">
      <possibleValues description="alta" name="alta" value="alta"/>
      <possibleValues description="regular" name="regular" value="regular"/>
      <possibleValues description="mínima" name="mínima" value="mínima"/>
    </myContextAttributes>
  </myDimensions>
  <Links attribute="Tipo de proyecto">
    <PriorityAttribute attribute="Duración"/>
    <PriorityAttribute attribute="Involucramiento del cliente"/>
    <PriorityAttribute attribute="Conocimiento del problema"/>
    <PriorityAttribute attribute="Restricciones de tiempo"/>
    <PriorityAttribute attribute="Restricciones de presupuesto"/>
  </Links>
  <Links attribute="Duración">
    <PriorityAttribute attribute="Restricciones de presupuesto"/>
    <PriorityAttribute attribute="Tamaño del equipo"/>
  </Links>
  <Links attribute="Involucramiento del cliente">
    <PriorityAttribute attribute="Tamaño del equipo"/>
    <PriorityAttribute attribute="Experiencia del equipo"/>
  </Links>
  <Links attribute="Conocimiento del problema">
    <PriorityAttribute attribute="Experiencia del equipo"/>
    <PriorityAttribute attribute="Conocimiento del negocio"/>
  </Links>
  <Links attribute="Restricciones de tiempo">
    <PriorityAttribute attribute="Conocimiento del negocio"/>
    <PriorityAttribute attribute="Conocimiento del producto"/>
  </Links>
  <Links attribute="Restricciones de presupuesto">
    <PriorityAttribute attribute="Conocimiento del negocio"/>
    <PriorityAttribute attribute="Conocimiento del producto"/>
  </Links>
</Context>
```

Figura A1: Archivo XML que representa un modelo de contexto

B. Anexo II: Archivo XMI. Resultado de la configuración de un contexto concreto en el modulo Contexto Concreto

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<spcm:Context xmlns:spcm="http://contextmetamodel/1.0" xmlns:xmi="http://www.omg.org/XMI" description="Modelo canónico"
  <myDimensions description="Proyecto" name="Proyecto" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <myContextAttributes description="Tipo de proyecto" name="Tipo de proyecto" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="nuevo desarrollo" name="nuevo desarrollo" value="nuevo desarrollo" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="extensión" name="extensión" value="extensión" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="mantenimiento" name="mantenimiento" value="mantenimiento" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Duración" name="Duración" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="corto" name="corto" value="corto" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="medio" name="medio" value="medio" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="largo" name="largo" value="largo" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Involucramiento del cliente" name="Involucramiento del cliente" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="alto" name="alto" value="alto" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="medio" name="medio" value="medio" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="bajo" name="bajo" value="bajo" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="conocido" name="conocido" value="conocido" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Conocimiento del problema" name="Conocimiento del problema" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="claro" name="claro" value="claro" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="ambiguo" name="ambiguo" value="ambiguo" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="no claro" name="no claro" value="no claro" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Restricciones de tiempo" name="Restricciones de tiempo" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="muy restringido" name="muy restringido" value="muy restringido" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="típico" name="típico" value="típico" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="sin restricciones" name="sin restricciones" value="sin restricciones" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Restricciones de presupuesto" name="Restricciones de presupuesto" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="muy restringido" name="muy restringido" value="muy restringido" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="típico" name="típico" value="típico" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
      <possibleValues description="sin restricciones" name="sin restricciones" value="sin restricciones" xmi:Id="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    </myContextAttributes>
  </myDimensions>
  <myDimensions description="Equipo" name="Equipo" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    <myContextAttributes description="Tamaño del equipo" name="Tamaño del equipo" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="muy restringido" name="muy restringido" value="muy restringido" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="típico" name="típico" value="típico" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="sin restricciones" name="sin restricciones" value="sin restricciones" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    </myContextAttributes>
    <myContextAttributes description="Experiencia del equipo" name="Experiencia del equipo" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="alta" name="alta" value="alta" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="regular" name="regular" value="regular" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="baja" name="baja" value="baja" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    </myContextAttributes>
    <myContextAttributes description="Conocimiento del negocio" name="Conocimiento del negocio" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="conocido" name="conocido" value="conocido" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="abordable" name="abordable" value="abordable" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="desconocido" name="desconocido" value="desconocido" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    </myContextAttributes>
    <myContextAttributes description="Conocimiento del producto" name="Conocimiento del producto" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="conocido" name="conocido" value="conocido" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="abordable" name="abordable" value="abordable" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
      <possibleValues description="desconocido" name="desconocido" value="desconocido" xmi:Id="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    </myContextAttributes>
  </myDimensions>
  <myDimensions description="Producto" name="Producto" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
    <myContextAttributes description="Complejidad técnica" name="Complejidad técnica" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="alta" name="alta" value="alta" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="media" name="media" value="media" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="baja" name="baja" value="baja" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
    </myContextAttributes>
    <myContextAttributes description="Importancia de la calidad" name="Importancia de la calidad" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="alta" name="alta" value="alta" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="regular" name="regular" value="regular" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
      <possibleValues description="mínima" name="mínima" value="mínima" xmi:Id="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
    </myContextAttributes>
  </myDimensions>
  <myDimensions description="Proceso" name="Proceso" xmi:Id="MTM0MTQ3MzIxMDE4NjY2Mzc0NjI0NTgwOTA=">
    <myContextAttributes description="Enfoque del proceso" name="Enfoque del proceso" xmi:Id="MTM0MTQ3MzIxMDE4NjY2Mzc0NjI0NTgwOTA=">
      <possibleValues description="producto final" name="producto final" value="producto final" xmi:Id="MTM0MTQ3MzIxMDE4NjY2Mzc0NjI0NTgwOTA=">
      <possibleValues description="todo el producto" name="todo el producto" value="todo el producto" xmi:Id="MTM0MTQ3MzIxMDE4NjY2Mzc0NjI0NTgwOTA=">
    </myContextAttributes>
  </myDimensions>
  <myContextConfigurations description="fsadfs" name="34sdfs" xmi:Id="MTM0MTQ3MzIxMDE4NzI5NjQwODQ2NDUzNDc5NzU2MTk=">
    <contextAttributeConfiguration description="Tipo de proyecto" myContextAttributeValue="nuevo desarrollo" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Duración" myContextAttributeValue="corto" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Involucramiento del cliente" myContextAttributeValue="alto" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Conocimiento del problema" myContextAttributeValue="claro" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Restricciones de tiempo" myContextAttributeValue="muy restringido" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Restricciones de presupuesto" myContextAttributeValue="muy restringido" myContextElement="MTM0MTQ3MzIxMDE4MTYzODc5NjA3NDQ3MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Tamaño del equipo" myContextAttributeValue="muy restringido" myContextElement="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    <contextAttributeConfiguration description="Experiencia del equipo" myContextAttributeValue="alta" myContextElement="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    <contextAttributeConfiguration description="Conocimiento del negocio" myContextAttributeValue="conocido" myContextElement="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    <contextAttributeConfiguration description="Conocimiento del producto" myContextAttributeValue="conocido" myContextElement="MTM0MTQ3MzIxMDE4NDQ0xMzE3MDI1OTA2MDE3NTUzMDU=">
    <contextAttributeConfiguration description="Complejidad técnica" myContextAttributeValue="alta" myContextElement="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Importancia de la calidad" myContextAttributeValue="alta" myContextElement="MTM0MTQ3MzIxMDE4NTg0NzI1MjgzNTgwOTA=">
    <contextAttributeConfiguration description="Enfoque del proceso" myContextAttributeValue="producto final" myContextElement="MTM0MTQ3MzIxMDE4NjY2Mzc0NjI0NTgwOTA=">
  </myContextConfigurations>
</spcm:Context>

```

Figura B1: Archivo XMI que representa un contexto concreto

C. Anexo III: Pauta de entrevista

Las siguientes preguntas son contextadas con una escala de 1 a 7, según la escala de Likert

Usuario Ingeniero de procesos

1. ¿Qué tan fácil/difícil fue para usted agregar una nueva dimensión? (1: muy difícil, 7:muy fácil)
2. ¿Qué tan fácil/difícil fue para usted eliminar una dimensión? (1: muy difícil, 7:muy fácil)
3. ¿Qué tan fácil/difícil fue para usted agregar un nuevo atributo? (1: muy difícil, 7:muy fácil)
4. ¿Qué tan fácil/difícil fue para usted eliminar un atributo? (1: muy difícil, 7:muy fácil)
5. ¿Qué tan fácil/difícil fue para usted agregar un nuevo posible valor a un atributo? (1: muy difícil, 7:muy fácil)
6. ¿Qué tan fácil/difícil fue para usted eliminar un posible valor de un atributo? (1: muy difícil, 7:muy fácil)
7. ¿Qué tan fácil/difícil fue para usted modificar los datos de una dimensión? (1: muy difícil, 7:muy fácil)
8. ¿Qué tan fácil/difícil fue para usted crear nuevos links entre atributos del modelo de contexto? (1: muy difícil, 7:muy fácil)
9. ¿Qué tan fácil/difícil fue para usted descargar el archivo del modelo de contexto? (1: muy difícil, 7:muy fácil)
10. En general ¿Está usted satisfecho con la aplicación? (1: muy insatisfecho, 7:muy satisfecho)

Usuario jefe de proyecto

1. ¿Qué tan fácil/difícil fue para usted definir un nombre y descripción para el contexto concreto? (1: muy difícil, 7:muy fácil)
2. ¿Qué tan fácil/difícil fue para usted asignar los valores de los atributos? (1: muy difícil, 7:muy fácil)
3. ¿Qué tan fácil/difícil fue para usted descargar el archivo del contexto concreto? (1: muy difícil, 7:muy fácil)
4. En general ¿Está usted satisfecho con la aplicación? (1: muy insatisfecho, 7:muy satisfecho)

Las siguientes preguntas son respondidas cualitativamente

1. ¿Le parece adecuada la organización de la información en la aplicación ?
2. ¿Qué le gustaría que tuviese la aplicación en este momento? (en orden de prioridad)

D. Anexo IV: Cuestionario usuario final

	Muy de acuerdo	De acuerdo	Neutro	En desacuerdo	Muy en desacuerdo
La aplicación es fácil de navegar					
Es fácil encontrar la Información deseada					
Los enlaces son claramente identificados					
Los enlaces funcionan correctamente					
Los contenidos se cargan rápidamente (¡30 segundos)					
El uso de las imágenes es aceptable					
El uso del color es aceptable					
El diseño general de la aplicación es apropiado					
La organización de la información de la aplicación es apropiada					
El contenido de la aplicación es relevante					
La interfaz del sitio es placentera					
La aplicación tiene todas las funcionalidades esperadas					
La aplicación tiene todas las capacidades esperadas					

	Excelente	Bueno	Neutro	Regular	Deficiente
¿Cómo califica globalmente la aplicación analizada?					

Justifique su elección
