

**CONSTRUCCIÓN DEL MÓDULO DE TRAZABILIDAD ENTRE ELEMENTOS
UML PARA LA HERRAMIENTA AR2CA**

JOHN KELLY VILLOTA PISMAG

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2009**

**CONSTRUCCIÓN DEL MÓDULO DE TRAZABILIDAD ENTRE ELEMENTOS
UML PARA LA HERRAMIENTA AR2CA**

JOHN KELLY VILLOTA PISMAG

**Trabajo de Grado presentado como requisito parcial para optar al título de
Ingeniero de Sistemas**

**Dra. Raquel Anaya de Páez
Directora**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2009**

“Las ideas y conclusiones aportadas en el trabajo de grado son responsabilidad exclusiva de sus autores”.

Artículo 1º. Del acuerdo No. 324 del 11 de Octubre de 1966 emanado del honorable Consejo Directivo de la Universidad de Nariño.

NOTA DE ACEPTACIÓN

Asesora: Dra. Raquel Anaya de Páez

Jurado 1: Esp. Alexander Barón Salazar

Jurado 2: Esp. Jesús Insuasty

*To my Dad,
my Mom,
my Bro,
and my sweetheart Patricia,
because they made this possible*

AGRADECIMIENTOS

A la Doctora Raquel, por haberme abierto las puertas del grupo de investigación y por abrir las puertas de mi mente a una gran cantidad de nuevo conocimiento.

Al profe Alex, a Dianita Naspiran, a Gustavo Blanco, a Santiago Builes y a Diana Torres por su amistad, por su apoyo en las situaciones difíciles y por todos los buenos momentos durante todo este tiempo.

A mi papá y mi mamá por cada momento en que estuvieron conmigo, por sus consejos, por las alegrías y tristezas, porque ellos son los artífices de todo esto.

A mi hermanito por todo su inmenso apoyo, por todos los sueños, por las charlas inspiradoras y por estar siempre cuando lo necesitaba.

A Patricia por su amor, por estar a mi lado en cada pequeño logro, y por mostrarme una perspectiva diferente de la vida.

TABLA DE CONTENIDO

Pág.

GLOSARIO	14
RESUMEN.....	16
ABSTRACT.....	17
INTRODUCCIÓN.....	18
1 DESCRIPCIÓN DEL PROBLEMA	20
1.1 PLANTEAMIENTO DEL PROBLEMA	20
1.2 FORMULACIÓN DEL PROBLEMA	21
1.3 SISTEMATIZACIÓN DEL PROBLEMA.....	21
1.4 ALCANCE.....	22
1.5 JUSTIFICACIÓN.....	22
2 OBJETIVOS.....	24
2.1 OBJETIVO GENERAL.....	24
2.2 OBJETIVOS ESPECÍFICOS.....	24
3 MARCO TEÓRICO.....	25
3.1 TRAZABILIDAD	25
3.1.1 <i>Gestión de la trazabilidad.....</i>	<i>27</i>
3.1.2 <i>Automatización y herramientas de apoyo.....</i>	<i>29</i>
3.2 ANÁLISIS DE IMPACTO	30
3.2.1 <i>Proceso para realizar el Análisis de Impacto.....</i>	<i>31</i>
3.3 VISUALIZACIÓN DE LA INFORMACIÓN.....	32
3.4 EL MODELO DE BENJAMÍN FRY	36
3.4.1 <i>Adquisición de Datos.....</i>	<i>38</i>
3.4.2 <i>Análisis de Datos.....</i>	<i>38</i>
3.4.3 <i>Filtrado.....</i>	<i>40</i>
3.4.4 <i>Minería de datos.....</i>	<i>40</i>
3.4.5 <i>Representación de datos.....</i>	<i>44</i>
3.4.6 <i>Refinar.....</i>	<i>56</i>
3.4.7 <i>Interacción.....</i>	<i>58</i>
4 MANEJO DE TRAZABILIDAD EN LAS HERRAMIENTAS ACTUALES	60
4.1 ENTERPRISE ARCHITECT.....	60
4.1.1 <i>Trazabilidad en Enterprise Architect.....</i>	<i>60</i>
4.2 OBJECTEERING UML	63
4.2.1 <i>Trazabilidad en Objecteering UML.....</i>	<i>64</i>

4.3	BORLAND CALIBERRM	67
4.3.1	<i>Trazabilidad en Borland CaliberRM</i>	68
4.4	AR2CA	77
4.4.1	<i>Trazabilidad en AR2CA</i>	78
4.5	RESULTADOS DEL ANÁLISIS	79
5	ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO DENTRO DE AR2CA	80
5.1	ANÁLISIS DE LA FUNCIONALIDAD	80
5.2	DISEÑO DE LA INTERFACE GRÁFICA	84
5.3	DISEÑO DETALLADO	90
5.3.1	<i>Arquitectura de AR2CA</i>	90
5.3.2	<i>Uso de servicios existentes</i>	95
5.3.3	<i>Creación de Servicios</i>	98
5.4	TRAZABILIDAD EN AR2CA POSTERIOR A LA IMPLEMENTACIÓN	104
6	CONCLUSIONES	113
7	RECOMENDACIONES	115
8	REFERENCIAS BIBLIOGRÁFICAS	116

LISTA DE FIGURAS

Pág.

Figura 1. Representación de Trazabilidad entre los elementos de los diferentes modelos resultantes del análisis de un sistema.....	26
Figura 2. Tabla de Trazabilidad. Los requisitos en las filas dependen de los requisitos marcados en las columnas	28
Figura 3. Proceso para realizar el Análisis de Impacto. Tomado de [28].....	31
Figura 4. Tabla Periódica de métodos de visualización. Tomado de [21].....	35
Figura 5. ZipCode Aplicación desarrollada por Benjamín Fry en el marco de su tesis doctoral [28].....	35
Figura 6. Visualización de Redes Sociales – Vizster [19].....	36
Figura 7. Separación por áreas de las etapas del proceso de visualización de la información. Tomado de [29].....	37
Figura 8. Proceso para el Diseño de una herramienta para la visualización de Información Computacional. Tomado de [29].....	37
Figura 9. Tabla. Tomado de [31].....	45
Figura 10. Diagrama de Dispersión. Tomado de [32].....	45
Figura 11. Gráfico de líneas. Tomado de [33].....	46
Figura 12. Gráfico de Barras. Tomado de [10].....	47
Figura 13. Gráfico de Caja. Tomado de [12].....	48
Figura 14. Mapa de Calor. Tomado de [33].....	49
Figura 15. Árbol. Tomado de [14].....	49
Figura 16. Grafo. Tomado de [35].....	50
Figura 17. Histograma. Tomados de [36].....	52
Figura 18. Dendrograma. Tomado de [15].....	53
Figura 19. Gráfico de Coordenadas Paralelas. Tomado de [37].....	54
Figura 20. Gráfico de Radar. Tomado de [38].....	54
Figura 21. Treemap. Tomado de [39].....	55

Figura 22. Creación de un artefacto UML nuevo a partir de una relación Realización	61
Figura 23. Creación de una relación Trace a partir de dos artefactos UML previamente creados.....	61
Figura 24. Creación de una relación Realización haciendo uso de la matriz de trazabilidad	61
Figura 25. Enterprise Architect - Matriz de Trazabilidad con relaciones horizontales	62
Figura 26. Enterprise Architect – Matriz de Trazabilidad con relaciones verticales	62
Figura 27. Objecteering Modeler – Fast Link Creator.....	64
Figura 28. Objecteering Modeler – Relaciones de Fast Link Creator, la ultima Trazabilidad	65
Figura 29. Objecteering Modeler – Fast Link Creator en estado de Edición.....	65
Figura 30. Objecteering Modeler - Fast Link Creator mostrando una relación de Trazabilidad	66
Figura 31. Objecteering Modeler – Auxiliary View.....	66
Figura 32. Objecteering Modeler – Auxiliary View Borrado de relación.....	67
Figura 33. CaliberRM – Pantalla de edición de requisitos	69
Figura 34. CaliberRM – Pantalla de edición de trazabilidad de requisitos	70
Figura 35. CaliberRM – Selección de requisito con el cual se desea realizar una traza.....	70
Figura 36. CaliberRM – Botón que traza desde el requisito	71
Figura 37. CaliberRM – Botón que traza hacia el requisito	71
Figura 38. CaliberRM – Comentar el cambio realizado.....	72
Figura 39. CaliberRM – Pantalla de edición de trazabilidad de requisitos listando relaciones de traza.....	72
Figura 40. Together – Menú de despliegue de caso de uso, seleccionando Manage Traces	73

Figura 41. Together – Selección de requisitos para crear relación de traza con un proyecto CaliberRM	74
Figura 42. CaliberRM – Menú Traceability Matrix	75
Figura 43. CaliberRM – Ventana de filtro de parámetros de la matriz de trazabilidad	75
Figura 44. CaliberRM – Matriz de trazabilidad	76
Figura 45. CaliberRM – Menú Traceability Diagram.....	76
Figura 46. CaliberRM – Grafo de trazabilidad	77
Figura 47. Casos de Uso relacionados con la Matriz de Trazabilidad	82
Figura 48. Casos de uso relacionados con el Grafo de Trazabilidad	83
Figura 49. AR2CA - Imagen del Menú Contextual de Trazabilidad	85
Figura 50. Grafo por niveles completo	87
Figura 51. Grafo de Trazabilidad Completo	87
Figura 52. Grafo de Trazabilidad Interactivo	88
Figura 53. AR2CA – Elemento de modelado mostrado en la Visualización de Trazabilidad	88
Figura 54. AR2CA – Relaciones de modelado mostrado en la Visualización de Trazabilidad	89
Figura 55. Diagrama de Paquetes de AR2CA [22].....	91
Figura 56. Arquitectura de AR2CA modificada con el módulo de trazabilidad.....	98
Figura 57. AR2CA – Dialogo de selección de Artefactos para la matriz de trazabilidad	101
Figura 58. AR2CA – Dialogo de selección de Relaciones para la matriz de trazabilidad	102
Figura 59. AR2CA – Matriz de Trazabilidad	102
Figura 60. AR2CA – Grafo de Trazabilidad.....	103
Figura 61. AR2CA – Menú Mapeo para un diagrama de requisitos	104
Figura 62. AR2CA – Wizard Mapeo para un diagrama de requisitos hacia un diagrama de Casos de uso	105

Figura 63. AR2CA – Wizard Mapeo para un diagrama de requisitos hacia un diagrama de Casos de uso configurado	105
Figura 64. AR2CA – Wizard Mapeo – Mensaje de confirmación de aplicación de reglas	106
Figura 65. AR2CA – Menú Matriz de Trazabilidad	106
Figura 66. AR2CA – Matriz de Trazabilidad	107
Figura 67. AR2CA – Menú Desplegable Perfiles	107
Figura 68. AR2CA – Matriz de Trazabilidad Llena	108
Figura 69. AR2CA – Botones de configuración de la matriz.....	108
Figura 70. AR2CA – Pantalla de selección de Elementos.....	109
Figura 71. AR2CA – Pantalla de selección de Relaciones	109
Figura 72. AR2CA – Menú Adicionar Perfil	110
Figura 73. AR2CA – Menú Grafo de Trazabilidad Browser	111
Figura 74. AR2CA – Menú Grafo de Trazabilidad Área de trabajo.....	111
Figura 75. AR2CA – Menú Grafo de Trazabilidad Área de trabajo.....	111
Figura 76. AR2CA – Menú Grafo de Trazabilidad Área de trabajo.....	112

LISTA DE TABLAS

	Pág.
Tabla 1. Requisitos recolectados para la gestión de trazabilidad en AR2CA	80
Tabla 2. Descripción de los paquetes que componen la arquitectura de AR2CA..	95
Tabla 3. Servicios utilizados en la matriz de trazabilidad existentes en AR2CA....	97
Tabla 4. Servicios creados en las clases existentes en AR2CA.....	99

GLOSARIO

ANÁLISIS DE IMPACTO: La evaluación de los riesgos asociados al cambio, incluyendo estimados en recursos, esfuerzo y tiempo programado.

CASE: Computer Aided Software Engineering

DISEÑO DE INFORMACIÓN: Arte de la representación de información relevante de forma que se optimice la posibilidad de comprensión del receptor. Se considera parte de la Visualización de Información.

GRAFO: Conjunto de objetos llamados vértices o nodos unidos por enlaces llamados aristas o arcos, que permiten representar relaciones binarias entre elementos de un conjunto.

HCI (HUMAN-COMPUTER INTERACTION): O Interacción Persona-Computador en español, es la disciplina que estudia el intercambio de información entre las personas y los ordenadores. Su objetivo es que este intercambio sea más eficiente: minimiza los errores, incrementa la satisfacción, disminuye la frustración y en definitiva, hace más productivas las tareas que envuelven a las personas y los ordenadores.

INFOVIS: Visualización de la información.

MATRIZ: Arreglo rectangular de números o bien de elementos de un álgebra. Los elementos de las matrices están divididos en columnas y filas

QA (QUALITY ASSURANCE): O aseguramiento de la calidad en español, se puede definir como el esfuerzo total para plantear, organizar, dirigir y controlar la

calidad en un sistema de producción con el objetivo de dar al cliente productos con la calidad adecuada

SÍMBOLO: Imagen, figura u objeto con que se representa un concepto abstracto, moral o intelectual. El estudio de los símbolos se denomina simbología.

STAKEHOLDER: Cualquier persona u organización que sea afectada de forma directa por las actividades desarrolladas por otra organización o con interés sobre los procesos ejecutados en otra organización.

TRAZABILIDAD: La habilidad para mostrar las relaciones existentes entre los elementos de un modelo de sistema y la dependencia que existe entre estos

UML: Unified Modeling Language. Lenguaje de modelado visual que se usa para especificar, visualizar y documentar artefactos de un sistema software.

USABILIDAD: Capacidad de un sistema de ser usado fácil o eficientemente.

VISUALIZACIÓN: Generación de una imagen mental o una imagen real de algo abstracto o invisible.

VISUALIZACIÓN DE LA INFORMACIÓN: Rama de la computación gráfica que se ocupa de representar gráficamente datos no gráficos y que acostumbran presentarse como valores numéricos independientes

WIKI: Sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web. Los usuarios pueden crear, modificar o borrar un mismo texto que comparten.

RESUMEN

CON EL AUGE DE LOS PROYECTOS SOFTWARE EN TODOS LOS NIVELES EMPRESARIALES Y CON LA APLICACIÓN DE PROCESOS DE INGENIERÍA DEL SOFTWARE EN DICHO PROYECTOS, CADA DÍA SE BUSCA LA FORMA DE CÓMO MEJORAR EL ASEGURAMIENTO DE LA CALIDAD DE ESTOS.

UNA DE LAS TÉCNICAS QUE SE HAN ESTABLECIDO Y ACOGIDO POR LA INDUSTRIA ES EL SEGUIMIENTO EVOLUTIVO DE LOS MODELOS DE ANÁLISIS Y DE SUS ELEMENTOS. PARA ELLO SE HA VISTO LA UTILIDAD QUE TIENE LA MATRIZ DE TRAZABILIDAD, HERRAMIENTA DE MAYOR DIFUSIÓN ENTRE LAS HERRAMIENTAS CASE COMERCIALES, AUNQUE SE RECONOCE SUS DEBILIDADES INHERENTES. OTRA HERRAMIENTA, LA CUAL ESTÁ TOMANDO AUGE, ES EL GRAFO DE TRAZABILIDAD EN LA CUAL BUSCA RESOLVER LOS PROBLEMAS QUE EXISTEN EN LA MATRIZ DE TRAZABILIDAD, EL GRAFO DE TRAZABILIDAD MUESTRA DE FORMA GRÁFICA COMO UN ELEMENTO EVOLUCIONA A TRAVÉS DE TODO EL CICLO DE VIDA.

ACTUALMENTE MUCHOS PROYECTOS BUSCAR APORTAR EN EL DESARROLLO DE SOFTWARE DE MAYOR CALIDAD, ESTE ES EL CASO DE MMEDUSA “MARCO METODOLÓGICO PARA EL DESARROLLO ORIENTADO A ASPECTOS” DONDE SE PRESENTA UNA METODOLOGÍA COMPLETA PARA EL DESARROLLO DE SOFTWARE Y DENTRO DE MMEDUSA EL PROYECTO AR2CA “ARQUITECTURA DE REFINAMIENTO Y REUTILIZACIÓN DE COMPONENTES DE ANÁLISIS” QUE DESARROLLA UNA HERRAMIENTA CASE DONDE ÉSTE TRABAJO DE INVESTIGACIÓN CONSTRUYE LAS HERRAMIENTAS DE TRAZABILIDAD: MATRIZ DE TRAZABILIDAD Y GRAFO DE TRAZABILIDAD.

PALABRAS CLAVE: TRAZABILIDAD, VISUALIZACIÓN, GRAFO, CALIDAD, INGENIERÍA, SOFTWARE

ABSTRACT

WITH THE RISE OF SOFTWARE PROJECTS IN ALL LEVELS OF BUSINESS AND THE APPLICATION OF SOFTWARE ENGINEERING PROCESSES IN THIS KIND OF PROJECTS, EACH DAY THE RESEARCHERS ARE LOOKING FOR WAYS OF IMPROVING THE QUALITY ASSURANCE.

ONE OF THE TECHNIQUES THAT HAVE BEEN ESTABLISHED AND USED BY THE INDUSTRY IS THE EVOLUTIONARY TRACK OF ANALYSIS MODELS AND ITS ELEMENTS. THE INDUSTRY HAS SEEN THE UTILITY OF THE TRACEABILITY MATRIX, TOOL WITH WIDER DISSEMINATION AMONG THE COMMERCIAL CASE TOOL, ALTHOUGH RECOGNIZING THEIR INHERENT WEAKNESSES. ANOTHER TOOL, WHICH IS TAKING BOOM, IS THE TRACEABILITY GRAPH WHICH SEEKS TO SOLVE THE PROBLEMS IN THE TRACEABILITY MATRIX, TRACING THE GRAPH SHOWN IN GRAPHICAL FORM AS AN EVOLVING THROUGHOUT THE LIFE CYCLE.

MANY PROJECTS SEEK TO MAKE SOFTWARE DEVELOPMENT BETTER, THIS IS THE CASE MMEDUSA "MARCO METODOLÓGICO PARA EL DESARROLLO ORIENTADO A ASPECTOS" WHICH PRESENTS A COMPREHENSIVE METHODOLOGY FOR SOFTWARE DEVELOPMENT AND PROJECT MMEDUSA WITHIN AR2CA "ARQUITECTURA DE REFINAMIENTO Y REUTILIZACIÓN DE COMPONENTES DE ANÁLISIS" DEVELOPED A CASE TOOL IN WHICH THIS RESEARCH BUILDS TOOLS TRACEABILITY: TRACEABILITY MATRIX AND TRACEABILITY GRAPH.

KEYWORDS: TRACEABILITY, VISUALIZATION, GRAPH, QUALITY, ENGINEERING, SOFTWARE

INTRODUCCIÓN

La ingeniería del software tiene entre sus objetivos el proporcionar metodologías de control que mejoren la calidad de los proyectos que se desarrollen, bien sea por medio de metodologías de desarrollo, modelos arquitectónicos, o herramientas que hagan más ágil este proceso desarrollo.

Dentro de las diferentes opciones que se han creado para el control de calidad encontramos el análisis de impacto. Por medio de un análisis de impacto podemos definir si es o no válido un cambio dentro del proyecto e incluso saber que artefactos se verán afectados y en qué medida serán modificados. Un análisis de impacto tiene como insumo las relaciones evolutivas o relaciones de trazabilidad que tiene un proyecto.

Trazabilidad puede definirse como el grado en el cual una relación puede ser establecida entre dos o más elementos, relación que permite conocer la ubicación y la trayectoria de un artefacto o conjunto de artefactos a través de un proceso de desarrollo para su posterior análisis.

Con base en la premisa que un buen análisis de impacto, depende de un buen manejo de las relaciones de trazabilidad, y de una buena representación de estas; se plantea la construcción de un modelo gráfico que permita al analista manejar de una forma interactiva como se relacionan los diferentes artefactos de modelado. Adicional al método gráfico, se ve necesario construir un modelo matricial que permita interactuar en dos elementos haciendo más fácil el poder relacionar unos artefactos con otros.

La construcción de este módulo se realizará como extensión de la herramienta case AR2CA, permitiéndole a esta complementar su manejo de las reglas de refinamiento preestablecidas dentro del diseño inicial de la herramienta.

Este estudio hace parte del proyecto de investigación “Marco Metodológico Para El Desarrollo Orientado a Aspectos – MMEDUSA” desarrollado por el grupo de investigación en ingeniería de software de la universidad EAFIT y de la empresa de desarrollo de software AVANSOFT. S.A.

En este trabajo se presenta la implementación realizada para el soporte de trazabilidad dentro de la herramienta CASE AR2CA y se organiza así: En el primer capítulo se determinan los objetivos del proyecto, en el segundo capítulo el marco teórico en que se fundamente este trabajo, en el tercer capítulo se muestra el estudio realizado en diferentes herramienta de modelado acerca del soporte a trazabilidad que estas ofrecen, el cuarto capítulo explica puntualmente como se hizo la implementación del módulo en AR2CA y finalmente el en quinto capítulo se presentan algunas conclusiones y propuestas de trabajos futuros relacionados con la temática.

1 DESCRIPCIÓN DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Las herramientas CASE que se encuentran actualmente en el mercado tales como RequisitePro o ART's han implementado la trazabilidad como un módulo importante desde hace ya varios años. La forma en que se hizo desde un principio y hasta hoy en día es por medio de matrices de trazabilidad.

Las matrices de trazabilidad permiten ver y relacionar tan solo dos tipos de elementos al tiempo, los que se ubican en el eje x y los que se ubican en el eje y, esto no permite que se haga un análisis más profundo de cómo se relacionan los elementos en todos los niveles de análisis dentro del modelado de un sistema.

Otra característica a resaltar en las herramientas del mercado es que las relaciones creadas por medio de las matrices de trazabilidad no son caracterizadas, cosa que limita aun más el análisis correspondiente de cómo interactúan los diferentes artefactos de modelado.

En las herramientas más modernas, últimas versiones de productos como Enterprise Architect¹, se ha comenzado a implementar módulos gráficos de trazabilidad junto con las matrices de trazabilidad, este gráfico de trazabilidad es bastante útil para analizar de una manera más completa las relaciones y dependencias entre artefactos de modelado. El problema que presentan estas herramientas radica en que solo se tiene una única forma de visualización de relaciones, lo que limita al usuario de estas herramientas a basar sus interpretaciones a una vista bidimensional en un mundo con tres dimensiones.

¹ Enterprise Architect (Versión 7.0) producido por Sparx Systems

Dadas estas circunstancias, cuando se intente realizar un análisis de impacto este se verá limitado, no se podrá manejar de manera adecuada las dependencias entre un elemento y otro, no se alcanzará el objetivo principal de aplicación de la trazabilidad que es establecer de una forma clara si un cambio es justificado o el tiempo y costo de modificar un elemento del modelado de un sistema.

1.2 FORMULACIÓN DEL PROBLEMA

¿De qué manera se puede crear un mecanismo de trazabilidad que sea óptimo y que, a futuro, sirva como insumo suficiente para realizar un análisis de impacto del cambio bien fundamentado?

1.3 SISTEMATIZACIÓN DEL PROBLEMA

¿Las herramientas existentes están haciendo una correcta implementación de un mecanismo de trazabilidad en los proyectos que se modelan en ellas?

¿Cómo y cuales serian las características especiales que permitan tipificar las relaciones entre los artefactos de modelado?

¿Qué alcance tiene una matriz de trazabilidad si se desea ver todos los elementos que se desprenden o de los que depende un elemento de modelado?

¿Cómo la construcción de un grafo de trazabilidad puede mejorar los análisis que se hagan de las relaciones existentes entre elementos de modelado?

¿Cuáles son las bibliotecas gráficas de JAVA que permiten un mejor desarrollo de los grafos de trazabilidad?

1.4 ALCANCE

El módulo a construir en la herramienta AR2CA tiene como fin desarrollar dos tipos de vista de trazabilidad, la vista matricial, matriz de trazabilidad y la vista gráfica, grafo de trazabilidad.

La vista matricial, una vista que permitirá relacionar dos tipos de elementos y verificar que tipo de relación tiene cada uno con los demás. Esta vista, gracias a la forma en que se presentan los elementos, permite de una manera más organizada, la creación, modificación y eliminación de las relaciones existentes entre elementos además de permitir la tipificación de las relaciones.

La vista gráfica o el grafo de trazabilidad, vista que gracias a la interacción visual que tendrá el analista permitirá obtener, de forma mucho más rápida y de mejor calidad, información o conocimiento de cómo están interactuando los elementos del modelado y así poder definir con más propiedad cuál será el impacto de un cambio dentro de todo el proyecto.

Como es lógico los dos módulos se relacionaran de forma directa, cualquier cambio realizado en la matriz de trazabilidad se verá reflejado en el grafo de trazabilidad y viceversa.

1.5 JUSTIFICACIÓN

El impacto que pueden tener los cambios en un proyecto durante cualquier etapa de su ciclo de vida es muy grande, ya que no afecta solo los elementos involucrados directamente con ese momento en el tiempo, también afecta los artefactos que se usados con anterioridad y los que, a futuro, se vayan a usar. Si

no se logra controlar estos cambios puede causar contratiempos o dificultades en vez de mejorar y corregir errores en el proyecto.

La trazabilidad es la capacidad para relacionar artefactos de modelado unos con otros. Estas relaciones son dirigidas y además estas relaciones posibilitan la navegabilidad entre elementos.

Dado que la trazabilidad permite evaluar que artefactos se ven afectados por un cambio durante cualquier etapa del ciclo de vida del proyecto podemos asegurar la integridad entre los resultados obtenidos entre una etapa y otra del ciclo de vida. Por ejemplo un analista puede asegurar que todos los requisitos solicitados por un cliente sean contemplados tanto en el análisis como en el diseño y desarrollo de todo el proyecto, y de forma inversa verificando que todo módulo creado en un sistema tiene una imagen como requerimiento por parte de un cliente.

Cuando el sistema comienza su evolución natural, la trazabilidad dejara de ser un actor principal para convertirse en insumo, insumo que permitirá realizar un análisis de impacto. Por medio de un análisis de impacto podemos definir con mayor certeza la cantidad de recursos a usar, el tiempo necesario para el cambio y la justificación costo/beneficio. Por ejemplo un análisis de impacto permitirá ver si el cambio en un requisito o en un módulo del sistema estudiado será muy costoso o incluso si vale la pena realizar el nombrado cambio.

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Implementar dentro de la herramienta case AR2CA, un módulo que permita crear, ver, modificar y eliminar tanto gráficamente como conceptualmente las relaciones de trazabilidad existentes entre los diferentes elementos de modelado UML.

2.2 OBJETIVOS ESPECÍFICOS

Hacer un estudio de las diferentes opciones que ofrecen las herramientas CASE actuales respecto a las estrategias de trazabilidad y la manera en que son manejadas por cada una.

Definir la representación de los diferentes tipos de trazabilidad y como se hará su implementación dentro de la herramienta AR2CA, para que permitan definir a ciencia cierta la forma en que están interactuando los elementos de modelado.

Diseñar una matriz de trazabilidad que permita mostrar de manera clara las relaciones existentes entre los diferentes artefactos de modelado en un determinado nivel de abstracción.

Diseñar un grafo de trazabilidad que permita navegar entre los elementos de modelado mostrando que artefactos serán afectados por un cambio en el modelo del sistema.

Hacer un estudio del framework *Prefuse* para la implementación de los grafos de trazabilidad dentro de la herramienta AR2CA

3 MARCO TEÓRICO

3.1 TRAZABILIDAD

La trazabilidad se puede definir de muchas formas, Alexander Radev, Sven Breuner lo definen como: la habilidad para mostrar las relaciones existentes entre los elementos de un modelo de sistema y la dependencia que existe entre estos [2]. Los analistas de UNISYS lo definen como: la habilidad de relacionar características de los artefactos existentes en un proyecto. Las relaciones existentes entre los elementos tienen una dirección definida y pueden ser usadas como guía para navegar a través de los elementos de forma significativa [3]. La ISO 9000.2000 define la trazabilidad como: la habilidad para recuperar historia, uso y localización de ítem o actividades similares por medio de una identificación registrada [4]. Se define también como “la habilidad para determinar realmente cómo una pieza de un artefacto de software (requisito, diseño, código) afecta a otros [1].

La ingeniería de software estandariza la trazabilidad como un atributo de calidad que permite validar y controlar la transformación de los elementos y artefactos de software [1]. La trazabilidad hace posible buscar un cambio a un requisito y encontrar sus partes en los detalles del diseño y el código que son afectados por el cambio.

Con el uso de relaciones de traza entre los elementos de modelado, un experto puede asegurar que todas las reglas de negocio son implementadas al menos por un caso de uso y que todos los requerimientos considerados son el resultado de las necesidades de los stakeholders. Estas relaciones ayudan a prevenir algunos problemas típicos que pueden aparecer durante el desarrollo de un proyecto

software. Problemas como alcance desproporcionado o la implementación incompleta de los requisitos.

El uso de la trazabilidad además permite capturar conocimiento. Cuando un miembro del equipo deja el proyecto, no se desea que su conocimiento acerca del sistema en desarrollo se vaya con él. Por medio del uso continuo de trazabilidad durante todo el ciclo de vida del proyecto se tiene un mejor entendimiento de los requisitos del sistema y los factores que motivaron su definición. Llevar las relaciones de traza desde los requisitos hasta los casos de prueba asegura que se cubran todas las funcionalidades o necesidades existentes en el sistema.

Es esencial mantener actualizada la documentación relacionada con el proyecto, registrando los diferentes avances que se hagan en todas las etapas del desarrollo de software realizando el cruce entre los elementos involucrados. En la figura 1 se muestra una representación de las trazas existentes entre los elementos de los diferentes modelos resultantes del análisis del sistema.

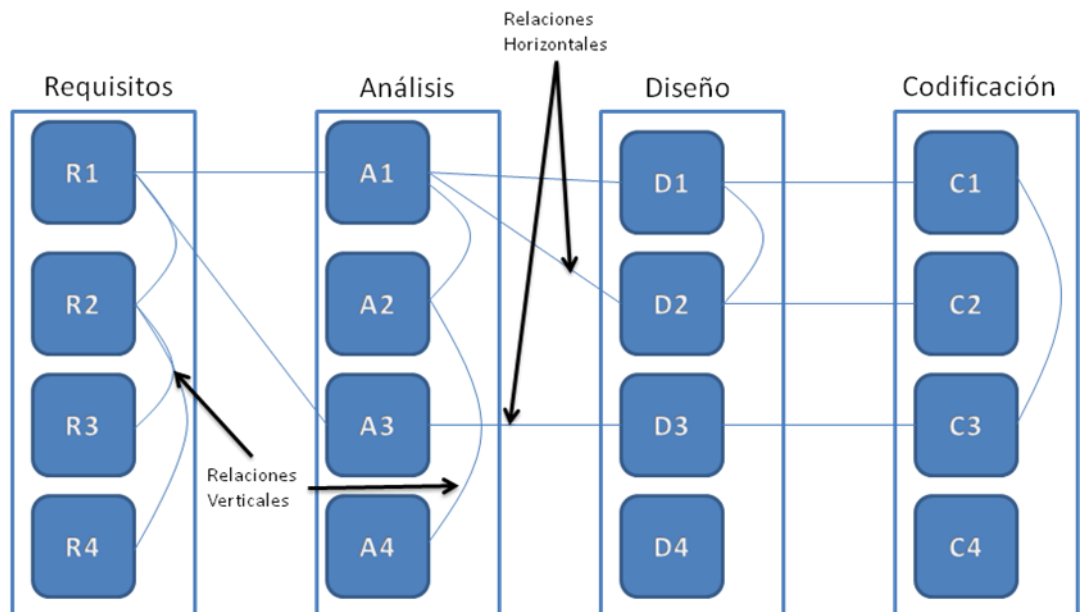


Figura 1. Representación de Trazabilidad entre los elementos de los diferentes modelos resultantes del análisis de un sistema

A pesar de que esto podría parecer un esfuerzo adicional, el proveer trazabilidad en el proyecto ayudará a prevenir un mayor esfuerzo cuando el sistema necesite ser modificado, o durante el desarrollo de una fase más adelante. La trazabilidad asistirá la justificación y toma de decisiones manteniendo la consistencia de la implementación los requerimientos y necesidades de los usuarios.

Además, esto proveerá la habilidad para que el impacto de análisis sea más efectivo, para descubrir en que forma un cambio deseado afectaría la totalidad del sistema y cuanto esfuerzo involucraría la implementación de dicho cambio. En particular, esto también ayuda a determinar si un cambio vale la pena ser implementado, o si el costo resultante será muy alto.

Los métodos y modelos de trazabilidad se basan en el proceso de desarrollo de software aplicado en un determinado sistema y de allí se desprenden tareas como determinar los elementos por trazar, trazar los elementos de un modelo a otro modelo, analizar los resultados de la trazabilidad, discutir incertidumbres con los desarrolladores y adaptar los modelos para la trazabilidad.

La trazabilidad está determinada por las diferentes dependencias de traza (relaciones entre elementos trazados) que son valoradas por los participantes (stakeholders) del sistema de software. Ellos determinarán la validez, flujo, evolución, metas y prioridades de los requisitos. Pero aun así hay que tener cuidado, puesto que la información vista desde los participantes puede ser muy confusa y dispersa [1].

3.1.1 Gestión de la trazabilidad

Un ejemplo sencillo de gestión de trazabilidad de requisitos es una Tabla de trazabilidad. En esta tabla, las dependencias entre los diferentes requisitos pueden hacerse explícitos enumerando cada requisito en la fila y marcando en las

columnas los requisitos que dependen del primero, como se muestra en la siguiente figura.

Depende de	R1	R2	R3
R1		X	
R2	X		X
R3			

Figura 2. Tabla de Trazabilidad. Los requisitos en las filas dependen de los requisitos marcados en las columnas

Esta versión básica de trazabilidad en un modelo, o también llamada trazabilidad vertical, es comúnmente complementada por la introducción de diferentes tipos de relaciones entre los requisitos. Una extensión común es el uso de diferentes símbolos para las relaciones: “Rx detalla a Ry” o “Rx es restringido por Ry”, y otras. Puesto que cada tabla puede convertirse en una gran lista de requisitos, también puede ser útil crear grupos de requisitos.

Una aproximación común para visualizar trazabilidad entre diferentes modelos, también llamada trazabilidad horizontal, es una red de trazabilidad. La red está basada en la enumeración de los elementos en cada uno de los documentos relevantes y en el establecer las relaciones entre los correspondientes elementos. Lo que significa que también se puede incluir trazabilidad vertical, como se muestra en la Figura 1.

Con las herramientas adecuadas, la creación de las relaciones horizontales de la red podrían ser parcialmente automáticas, en caso de aplicar un mismo esquema de para la asignación de nombres a través de los diferentes modelos. Con una herramienta, esto es fácilmente manejable y permite adicionar una dimensión más a la trazabilidad: la trazabilidad en el tiempo.

3.1.2 Automatización y herramientas de apoyo

Hoy en día existen un gran número de proyectos de investigación que pretenden implementar trazabilidad de requisitos en diferentes herramientas CASE.

La mayoría de ellos producen matrices de trazabilidad y resultados adicionales o avanzados cuando el sistema tiene la capacidad de ser ejecutado. Cada proyecto tiene un esfuerzo menor en esta dirección.

En el mercado existen algunas herramientas que pueden ayudar en este proceso. Las más importantes y populares son:

- Borland - CaliberRM
- IBM Rational - RequisitePro / AnalystStudio
- Vitech Corporation - CORE
- Telelogic - DOORS
- CA – Advantage

Una forma fácil y económica para la aplicación de trazabilidad dentro de un proyecto software también sería el uso de una Wiki, que este conectada al sistema de control de versiones y trace todo a través de sí. Una buena candidata es Edgewall Trac, basada en Python y MySQL. Como sea, cualquier buen gestor de Wiki puede ser utilizado para este propósito con un poco de creatividad, orientación y disciplina.

En [5] se encuentra información relacionada con las herramientas que permiten el manejo de trazabilidad y documentos comparativos de las características más relevantes de estas herramientas.

3.2 ANÁLISIS DE IMPACTO

Aproximadamente el 40% del mantenimiento radica en reprocesos, y si no son bien definidos estos acarrearán problemas que finalmente terminan prematuramente con la vida útil del sistema.

Las actividades más complejas del proceso de mantenimiento son: entender los cambios necesarios para cumplir con los nuevos requerimientos y controlar el efecto que tendrán esos cambios en todo el sistema. Entender como los cambios afectan la totalidad del sistema es llamado *análisis de impacto del cambio*.

Varios autores ha definido de diferente forma el análisis de impacto: En [25] Pfleeger y Bohner lo definen como: la evaluación de los riesgos asociados al cambio, incluyendo estimados en recursos, esfuerzo y tiempo programado. En [26] Turver y Munro dicen que: es evaluación de un cambio, al código fuente de un módulo, sobre los demás módulos del sistema. Determinando el alcance de un cambio y midiendo su complejidad. Finalmente en [27] Arnold y Bohner definen el análisis de impacto como: identificar las potenciales consecuencias de un cambio, o estimar que se necesita para llevar a buen término el cambio propuesto.

Este análisis de impacto del cambio permite que los procesos necesarios para cumplir con los nuevos requisitos se cumplan de una mejor forma, más fácilmente y dentro de un estimado de tiempo y dinero mucho más real.

3.2.1 Proceso para realizar el Análisis de Impacto

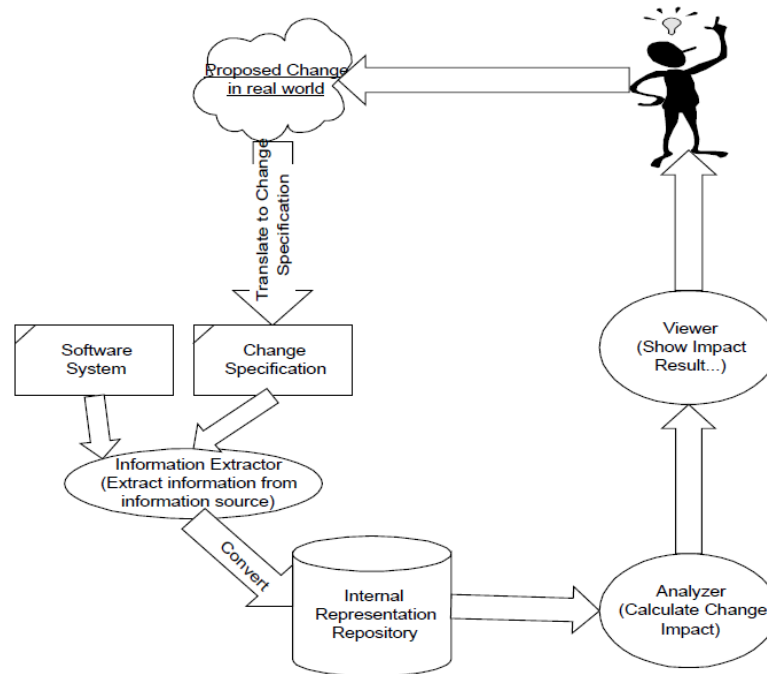


Figura 3. Proceso para realizar el Análisis de Impacto. Tomado de [28]

Para realizar un análisis de impacto se ha definido las siguientes tareas o actividades:

1. Convertir el cambio propuesto en una especificación de cambio del sistema
2. Extraer información de una fuente de información y convertirla en una representación interna de información
3. Calcular el impacto del cambio para el cambio propuesto. Realizar los paso 1 a 3 para cada cambio propuesto.
4. Hacer un estimado de los recursos necesarios, teniendo en cuenta aspectos como el tamaño o complejidad del sistema
5. Analizar el costo y los beneficios de la solicitud de cambio, de la misma forma que se evalúa una aplicación nueva.

6. El administrador de mantenimiento del proyecto muestra a los usuarios las implicaciones que tiene el cambio solicitado, esto en términos de negocio no en términos técnicos, para que estos decidan si autorizan o no el cambio

Como se ve, el análisis de impacto es una tarea difícil y tediosa para ejecutar un cambio dentro de un proyecto software. La gestión manual es una labor muy intensiva y propensa a errores y las herramientas CASE no cuentan con un sistema avanzado para la gestión del análisis impacto ya que aun necesitan de la interacción humana directa.

No obstante, al realizar un análisis de impacto, a pesar de ser un esfuerzo adicional es una gran ayuda en el desarrollo de un proyecto, ya que permite reducir los costos necesarios para realizar el cambio con éxito dado que permite estimar con mayor exactitud los costos y la cantidad de recursos necesarios para realizar el cambio con éxito. También permite prever otros cambios necesarios, dando pie a organizar de forma más adecuada el trabajo con los recursos involucrados. Permite mantener un nivel de calidad alto gracias a que asegura que las restricciones externas sobre otros módulos no se vean afectadas con el cambio realizado, o proveer una alternativa al cambio propuesto y crea casos de prueba específicos al los módulos afectados por el cambio realizado ahorrando tiempo y esfuerzo en esta tarea.

3.3 VISUALIZACIÓN DE LA INFORMACIÓN

La visualización es la representación perceptual adecuada tanto de los datos con parámetros múltiples como de las tendencias y las relaciones subyacentes que existen entre ellos. Su propósito no es la creación de las imágenes en sí mismas sino el *insigth*, es decir, la asimilación rápida de información o monitoreo de grandes cantidades de datos. La Visualización es parte de los nuevos medios

hechos posible debido al desarrollo de la visualización en computadoras en tiempo real. [20]

Dentro de la computación gráfica se ha subdividido la visualización en diferentes áreas de aplicación, a partir de un trabajo realizado por Ralph Lengler y Martin J. Eppler en la Universidad de Lugano (Suiza) [21] encontramos una “*tabla de métodos de visualización*” (Figura 4) donde se crean seis áreas de aplicación:

- Visualización de Datos: incluye formatos estándar cuantitativos, tales como gráficos circulares, áreas o Línea de Gráficos. Se trata de representaciones visuales de datos cuantitativos en forma esquemática (ya sea con o sin ejes), todos con el mismo propósito, principalmente para obtener una visión de conjunto de datos. Una correspondencia podría ser la combinación entre la visualización de datos (responder "cuánto" preguntas) y metáforas visuales (cómo responder a las preguntas y por qué). [21]
- Visualización de la información: tales como redes semánticas o Treemaps, se define como el uso interactivo de representaciones visuales de datos para completar la cognición. Esto significa que los datos se transforma en una imagen, dentro de un espacio asignado. La imagen puede ser cambiado por los usuarios, ya que estos trabajan con él. [21]
- Visualización de conceptos: como un mapa conceptual o un diagrama de Gantt; estos son los métodos para elaborar conceptos cualitativos, ideas, planes y análisis a través de la ayuda de procedimientos de mapeado. La visualización de conocimientos a través de conceptos suele presentarse en un gráfico 2-D, donde los conceptos (por lo general representados en cajas o círculos), están conectados por arcos dirigidos codificando las relaciones entre pares de conceptos. Estas relaciones consisten normalmente de los verbos, formando proposiciones o frases para cada par de conceptos. [21]
- Visualización de Metáforas: se trata de plantillas simples que sirven para transmitir ideas complejas. Las metáforas visuales cumplen una doble

función, en primer lugar ubicar la información de manera que sea organizada y estructurada. En segundo lugar que transmitir una idea acerca de la información representada a través de las características fundamentales de la metáfora que se emplea. [21]

- Visualización estratégica: Se define "como el uso sistemático de representaciones visuales complementarias para mejorar el análisis, desarrollo, formulación, comunicación, y la aplicación de estrategias en las organizaciones." Éste es el más específico de todos los grupos, como que ha logrado gran relevancia en la gestión. [21]
- Visualización compuesta: Consiste en el uso de varios de los formatos mencionados. Pueden ser complejos mapas de conocimiento que contienen diagramas y elementos metafóricos, mapas conceptuales con gráficos cuantitativos, o del tamaño de una pared (infomurales). Este grupo normalmente se designa para, el uso complementario de los diferentes formatos de representación gráfica en un único esquema o marco. Según Tufte es el resultado del uso de dos (o más) representaciones de datos espacialmente distintas, cada una de las cuales puede funcionar de manera independiente, pero que pueden ser utilizados juntos para correlacionar la información en una representación con que en otro. [21]

También, de manera personal, se puede definir a la visualización de la información como la rama de la computación gráfica que se ocupa de representar gráficamente datos no gráficos y que acostumbra presentarse como valores numéricos independientes lo que hace difícil conocer su interacción y leer u obtener resultados significativos para un determinado grupo de personas.

Existen varios trabajos respecto a cómo se puede optimizar la muestra de la información existente en bases de datos, repositorios u otros volúmenes de datos de diferentes dimensiones, pero la gran mayoría de ellos se dirigen hacia la

extracción de información en grandes volúmenes de datos como los existentes en redes sociales o almacenes de datos.

A PERIODIC TABLE OF VISUALIZATION METHODS

C continuum		Data Visualization Visual representations of quantitative data in schematic form (either with or without axes)										Strategy Visualization The systematic use of complementary visual representations in the analysis, development, formulation, communication, and implementation of strategies in organizations.										G graphic facilitation													
Tb table		Ca cartesian coordinates		Information Visualization The use of interactive visual representations of data to amplify cognition. This means that the data is transformed into an image, it is mapped to screen space. The image can be changed by users as they proceed working with it.										Metaphor Visualization Visual Metaphors position information graphically to organize and structure information. They also convey an insight about the represented information through the key characteristics of the metaphor that is employed.										Ct cartoon											
Pi pie chart		L line chart		Concept Visualization Methods to elaborate (mostly) qualitative concepts, ideas, plans, and analyses.										Compound Visualization The complementary use of different graphic representation formats in one single schema or frame.										Ri rich picture											
B bar chart		Ar area chart		Sa sasky diagram		Pa parallel coordinates		Hy hyperbolic tree		Cy cyclic diagram		T timeline		Ve venn/euler diagram		Mi mindmap		Sq square of opposites		Co concentric circles		Ar argument slide		Co communication diagram		Gc gant chart		Pe perspectives diagram		D dilemma diagram		Pr parameter ruler		Kn knowledge map	
Hi histogram		Sc scatterplot		R radar chart cobweb		Ch chernoff faces		E entity relationship diagram		Fb feedback cycle diagram		N nassi shneiderman diagram		Cl clustering		L layer chart		Py pyramid technique		Ca cause-effect chains		Tl toulmin map		Dt decision tree		Cp cpm critical path method		Cf concept fan		Co concept map		Ic iceberg		Gm cognitive mapping	
Tk tukey box plot		Sp spectrogram		Pa pareto chart		Tr treemaps		Te tensor diagram		Sy system dyn./loop diagrams		Fl flow chart		Se semantic network		So soft system modeling		Sm synergy map		Fo force field diagram		Ib ibi argumentation map		Pr process event chain		Pe PERT chart		Ey evocative knowledge maps		V Vee diagram		Hh hours 'n' bell chart		I informal	

Cy Process Visualization

Hy Structure Visualization

☀ Overview
☐ Detail
☉ Detail AND Overview
<> Divergent thinking
>> Convergent thinking

Note: Depending on your location and connection speed it can take some time to load a pop-up picture.

Sd supply demand chain		Pr performance charting		St strategy map		Oc organisation chart		Ho house of quality		Fd feedback diagram		Ft failure tree		Mq magic quadrant		Ld life-cycle diagram		Po porter's five forces		S s-cycle		Sm stakeholder map		Is ishikawa diagram		Tc technology roadmap	
Ed edgeworth box		Pf portfolio diagram		Sg strategic game board		Mz mitsberg's organigraph		Z zwickly's morphological box		Ad affinity diagram		De decision discovery diagram		Bm bcg matrix		Stc strategy canvas		Vc value chain		Hy hype-cycle		Sr stakeholder rating map		Ta taps		Sd spray diagram	

Figura 4. Tabla Periódica de métodos de visualización. Tomado de [21]

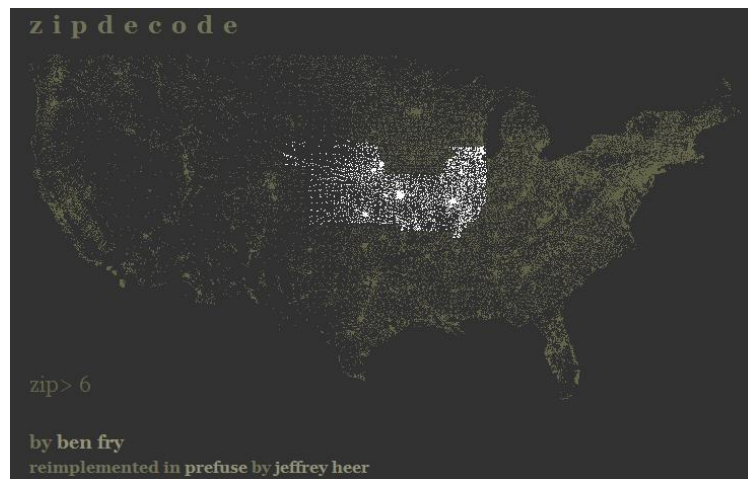


Figura 5. ZipCode Aplicación desarrollada por Benjamín Fry en el marco de su tesis doctoral [28]

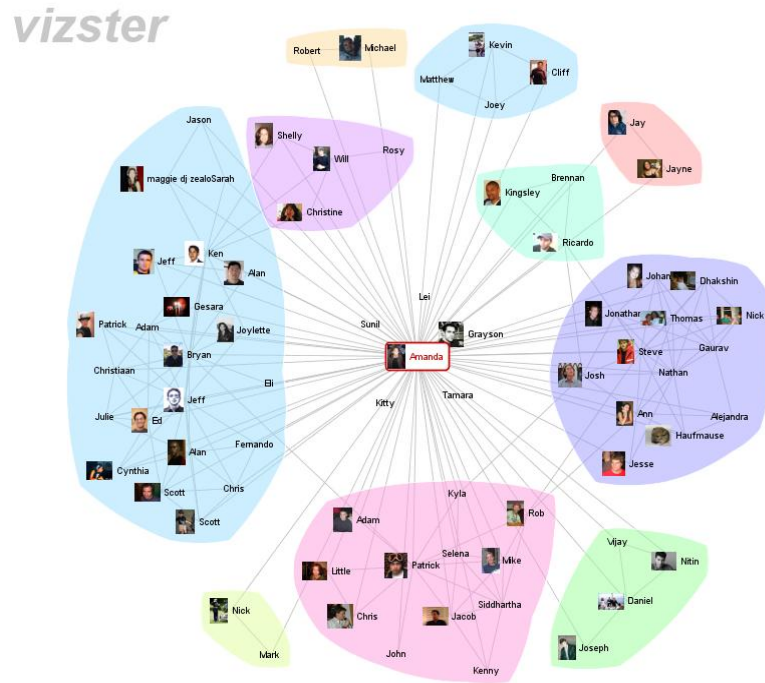


Figura 6. Visualización de Redes Sociales – Vizster [19]

3.4 EL MODELO DE BENJAMÍN FRY

Para realizar el diseño de la herramienta, se va a tomar el modelo propuesto por Benjamín Fry para el desarrollo de herramientas de visualización de información en su tesis doctoral [28].

Este proceso no quiere generar una nueva área interdisciplinaria, en vez de eso la idea es orientar diferentes áreas del conocimiento aplicadas hacia la solución o generación de una visualización de la información mucho más completa y precisa. En la siguiente figura se muestran las etapas definidas por Benjamín Fry en su modelo, y como estas encajan en diversas áreas.



Figura 7. Separación por áreas de las etapas del proceso de visualización de la información. Tomado de [29]

Un problema existente con el paso de la información entre un área y otra, a manera de “teléfono roto” perdiendo información valiosa entre paso y paso, hace que esta tarea sea aun más difícil y peor aun si tenemos en cuenta que cada experto dentro del área específica recibe un paquete de datos y entrega información sin tener en cuenta, para ambos casos, que es lo realmente útil para el siguiente experto. El objetivo de este desarrollo es permitir el paso de información de forma que la pérdida entre una etapa y otra sea mínima permitiendo obtener un resultado mucho más aproximado a la realidad.

El modelo plantea siete etapas para el diseño de una herramienta de visualización de información, estas etapas tiene una interacción definida que permite obtener un máximo rendimiento en el proceso. La siguiente figura muestra como las diferentes etapas planteadas por el modelo interactúan entre sí.

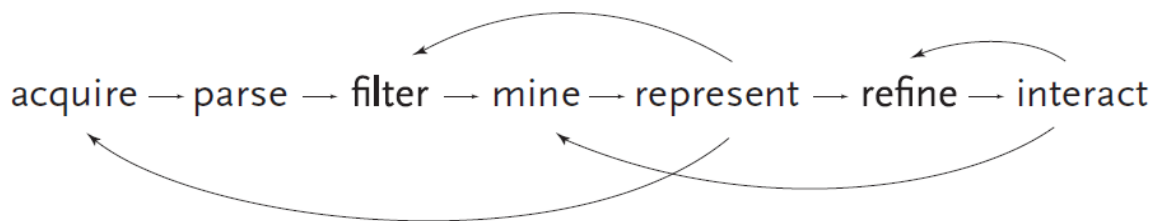


Figura 8. Proceso para el Diseño de una herramienta para la visualización de Información Computacional. Tomado de [29]

En primera instancia se observa un proceso lineal, pero en ciertas etapas existe la posibilidad de realizar la retroalimentación de una etapa anterior con la información resultante haciendo que los resultados obtenidos sean de mayor

calidad. A continuación se describe detalladamente cada etapa del modelo planteado por Benjamín Fray en su tesis doctoral.

3.4.1 Adquisición de Datos

Esta etapa del proceso está orientada hacia la selección del origen de los datos y el levantamiento que se hace de los mismos. Es importante conocer las diferentes fuentes de datos y sus posibles limitaciones para seleccionar el método que mejor se adecue al problema que daremos solución.

Algunos de los métodos más comunes para el levantamiento de datos son:

- Señal Análoga: Cargas de voltaje que al ser digitalizadas se convierten en una serie discreta de números.
- Archivo: Se trata de un conjunto organizado de datos relacionados o del mismo tipo, que pueden utilizarse en un mismo proceso.
- Base de Datos: Se define como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular [6]. Su principal deficiencia radica en una interacción complicada con el entorno.

3.4.2 Análisis de Datos

Esta etapa permite tomar los datos que fueron capturados anteriormente y por medio de diferentes procesos convertirlos en datos capaces de ser analizados por métodos tradicionales o haciendo uso de técnicas de visualización.

Para realizar un análisis completo de los datos, se debe adecuar y/o limpiar los datos para que los resultados de su futuro procesamiento sean mucho mejores.

Para que este proceso se lleve a cabo existen dos subprocesos que se deben cumplir que son: pre filtrado de datos y organización de datos.

3.4.2.1 Pre filtrado de datos

Hace referencia a los procesos más básicos sobre los datos recogidos en el primer anterior. Este proceso es tan sencillo como sincronizar y separar los datos que se usaran en el siguiente proceso.

Algunos métodos de pre filtrado son:

- Offset: Encontrar el conjunto de datos adecuado para la “minería” que se le realizara posteriormente.
- Filtrado a bajo nivel: A manera de limpiar los datos para que se puedan trabajar adecuadamente. Ej. Quitar los saltos a la señal de televisión haciéndola más pura.

3.4.2.2 Organización de datos

El objetivo de esta actividad es organizar los datos en estructuras conocidas que sean manipulables por una herramienta de software.

Algunos métodos comúnmente usados son:

- Bits periódicos o series de Bits: Cuando los datos están en cadenas periódicas o intervalos lineales donde una serie de estos representan o codifican un número o un valor.

- Caracteres de Texto: Este método puede tener una gran variedad de formas como separada por delimitadores, o interpretadas por un lenguaje o una gramática (gramáticas BNF) o marcas.
- Estructura de datos: Estructuras o jerarquías que permiten organizar los datos elementales de forma que se puedan manipular de una manera más adecuada y eficiente. Existen métodos que van desde una estructura (tabla) que codifica los datos almacenados hasta estructuras no uniformes que necesitan de complejos algoritmos para entender cómo se organizan los datos.

3.4.3 Filtrado

En la etapa de filtrado se separa el conjunto de datos que son realmente relevantes para el usuario de los demás por medio de técnicas de selección de datos. Esta etapa está relacionada directamente con la última actividad del proceso propuesto por Ben Fry (interacción) ya que los datos relevantes cambian fácilmente en este paso.

El filtrado puede entenderse de primera mano como eliminación de datos por métodos de selección, pero también existe la posibilidad de que esta etapa separe los datos en subgrupos de datos permitiendo que sean trabajados con mayor facilidad y velocidad.

3.4.4 Minería de datos

Esta categoría se encargar simplemente de cubrir las ramas de análisis de datos desde las matemáticas hasta la estadística, sin dejar de lado las operaciones avanzadas de minería de datos.

3.4.4.1 Matemática Básica y Estadística

John Tukey opina que la estadística es fundamental en el entendimiento del contenido y lo que representa el conjunto de valores existentes en los diferentes conjuntos de datos [30].

Entre los métodos utilizados están las diferentes operaciones de ordenamiento, distancia métrica, desviación estándar y normalización.

Otro aspecto importante de la estadística es la cantidad de datos existentes en un conjunto seleccionado, esto debe tenerse en cuenta cuando se desea hacer el conjunto de datos lo más compacto posible.

Algunos métodos recomendados son:

- Máximo y mínimo: Consiste simplemente en definir el valor más alto y el más bajo en el conjunto de datos.
- Media y Mediana: La media o promedio es la suma de los valores del conjunto dividido entre la cantidad de datos seleccionados. La mediana es el punto medio del conjunto de datos, sin importar cual sea su valor.
- Normalización: Este proceso hace referencia al remapeo que se hace de los valores existentes en el conjunto o otros en el rango de 0 a 1, logrando con esto que los datos sean mucho más fáciles de manejar en cálculos y operaciones posteriores.
- Este método es recomendado cuando se hace uso de la herramienta zoom en un proceso de visualización.
- Varianza y desviación estándar: La varianza provee una visión de cómo se distribuyen los valores dentro del conjunto de datos. La desviación estándar

(raíz cuadrada de la varianza) nos permite definir donde se encuentra un valor dentro de una distribución normal.

- Ordenar: Se puede decir que existen tres tipos básicos de ordenamiento: basado en el valor de los datos (ascendente o descendente), alfabéticamente o basado en funciones comparativas.
- Existen varios métodos de ordenamiento, el algoritmo más común es quicksort el cual, como su nombre lo indica, es el método más rápido para manejar conjuntos de datos aleatorios. Curiosamente es el peor método cuando los datos tienen cierto orden.
- Distancia métrica: se usa para calcular la distancia entre dos puntos, o en algunos casos dos vectores de datos (donde un punto puede tener múltiples valores).
- Conteo de únicos: si el conjunto de datos está conformado por una de n diferentes palabras o valores, una tarea común es hacer una lista de todas las palabras o valores diferentes en el conjunto de datos y contar cuantas ocurrencias de cada uno existen.

3.4.4.2 Medidas y transformación dimensional

Las medidas y transformaciones dimensionales incluyen métodos para el manejo de conjuntos de datos multidimensionales, como principle components analysis (PCA por sus siglas en ingles), los cuales son usados para determinar la dimensión más importante del conjunto de datos, o la transformada de Fourier, usada para separar una señal en sus frecuencias componentes.

- Principle components analysis: es un método del algebra lineal para descomponer matrices que tienen muchas dimensiones en matrices con menor número de dimensiones calculando cuales son las de mayor importancia.

- Escala multidimensional: es similar a PCA pero se usa cuando el conjunto de datos es como similar al conjunto Frank Gehry. Este método en vez de usar las dimensiones planares simples de PCA, escala multidimensional desenreda los datos mientras se intenta conservar las distancias relevantes entres los puntos.
- Transformada de Fourier: este método convierte señales al espacio de la frecuencia o el proceso contrario con la inversa de la transformada de Fourier.

3.4.4.3 Clasificación, marcado y búsqueda

Una vez que los datos han sido ajustados a un modelo y ha sido determinada su dimensión, la clasificación podría ser usada para agrupar elementos relacionados. Los métodos de agrupación (como k-means o self-organizing maps) existentes trata los conjuntos de datos como jerarquías de elementos relacionados.

Algunos métodos existentes son:

- Agrupación (clustering): Usado para agrupar datos similares para compararlas más fácilmente. Existen varios métodos de clustering entre los más importantes esta CAST y k-means
- Self-organizing maps: Este es un medio de clasificación basado en un proceso iterativo de selección.
- Métodos de marcado: Son una clase de métodos parte de la minería de datos que ayudan a determinar “que es importante” dentro de un conjunto de datos.

3.4.5 Representación de datos.

La etapa de representación busca la forma más simple de expresar todos los aspectos o características relevantes del conjunto de datos que se quiere visualizar. Esta representación trabaja la parte más básica de la visualización.

A continuación se presenta una lista de las representaciones más comunes o de mayor uso en la visualización de información.

3.4.5.1 Tabla

Es la forma donde se disponen y estructuran datos relacionados para que sea posible ver qué relaciones tienen con respecto a distintos conceptos, una tabla debe ser fácil de consultar y de leer, aunque cuando se trabaja una cantidad de datos excesiva el manejo de la información en una tabla se hace imposible. Las relaciones y jerarquías se deben ver al primer golpe de vista.

Las tablas pueden contener cualquier tipo de elemento visual y su estructuración y ajuste requiere bastante disciplina visual y tipográfica. [7] (Figura 9)

3.4.5.2 Diagrama de Dispersión

Un diagrama de dispersión es una representación gráfica de la relación entre dos variables, muy utilizada en las fases de Comprobación de teorías e identificación de causas raíz y en el Diseño de soluciones y mantenimiento de los resultados obtenidos. Tres conceptos especialmente destacables son que el descubrimiento de las verdaderas relaciones de causa-efecto es la clave de la resolución eficaz de un problema, que las relaciones de causa-efecto casi siempre muestran variaciones, y que es más fácil ver la relación en un diagrama de dispersión que en una simple tabla de números [8] (Figura 10)

LIPT- FRECUENCIA DE ESTRATEGIAS DE ACOSO			
ITEM	(LIPT dicotómico)	(LIPT escala)	
	n = 28	n = 79	
	% Positivos	% positivos	Media y (dt)
1. Superiores no dejan expresar	71,4	91,1	2,46 (1,31)
2. Interrumpen habla	75	83,5	2,06 (1,34)
3. Compañeros no dejan hablar	50	61,2	1,34 (1,36)
4. Gritan	39,3	77,2	2,02 (1,51)
5. Critican trabajo	85,7	82,3	2,15 (1,54)
6. Critican vida privada	28,6	53,2	1,09 (1,32)
7. Llamadas amenazantes	14,3	13,9	0,33 (0,90)
8. Amenaza verbal	60,7	70,9	1,77 (1,54)
9. Escritos amenazantes	28,6	21,5	0,47 (1,07)
10. - No miran, desprecio	78,6	77,2	1,95 (1,52)
11. Ignorarlo	60,7	82,2	2,02 (1,43)
12. No dirigirse a usted	64,3	62	1,54 (1,56)
13. Le evitan	35,7	49,4	1,11 (1,39)
14. Lugar trabajo aislado	32,1	50	1,42 (1,67)
15. Prohibir al resto hablarle	42,9	39	1,14 (1,60)
16. Trato como invisible	39,3	64,9	1,82 (1,66)
17. Calumnias, murmurar	71,4	72,1	1,99 (1,62)
18. Rumores falsos	67,9	75,3	2,05 (1,63)

Figura 9. Tabla. Tomado de [31]

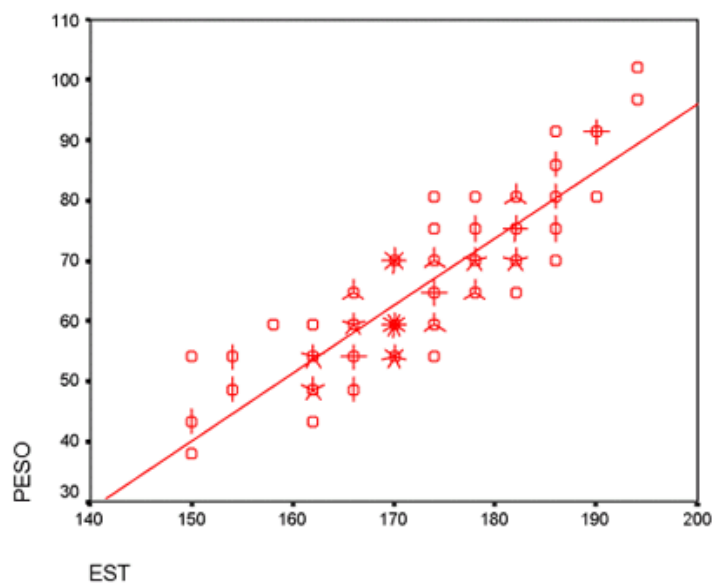


Figura 10. Diagrama de Dispersión. Tomado de [32]

3.4.5.3 Gráfico de Líneas

Los gráficos de líneas muestran una serie como un conjunto de puntos conectados mediante una línea. Los valores se representan por el alto de los puntos con relación al eje Y. Las etiquetas de las categorías se presentan en el eje X. Los gráficos de líneas suelen utilizarse para comparar valores a lo largo del tiempo. [9]

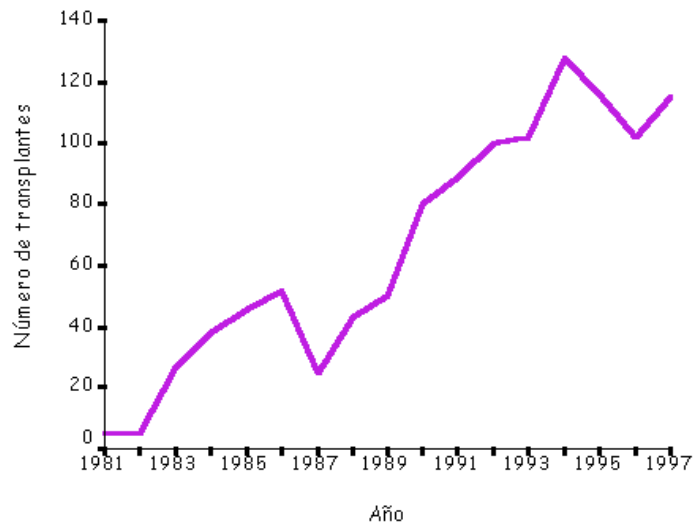


Figura 11. Gráfico de líneas. Tomado de [33]

3.4.5.4 Gráfico de Barras

Un gráfico de barras es aquella representación gráfica bidimensional en que los objetos gráficos elementales son un conjunto de rectángulos dispuestos paralelamente de manera que la extensión de los mismos es proporcional a la magnitud que se quiere representar.

Los rectángulos o barras pueden estar colocados horizontal o verticalmente. En éste último caso reciben también el nombre de gráficos de columnas. [10]

Gráfico de Columnas

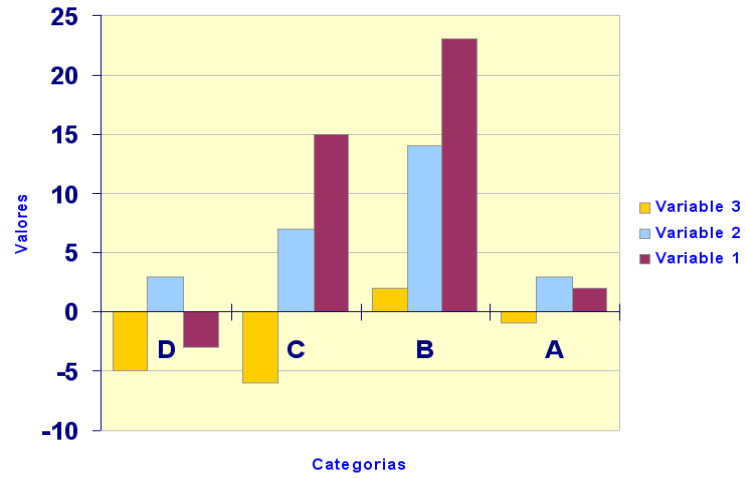


Gráfico de Barras

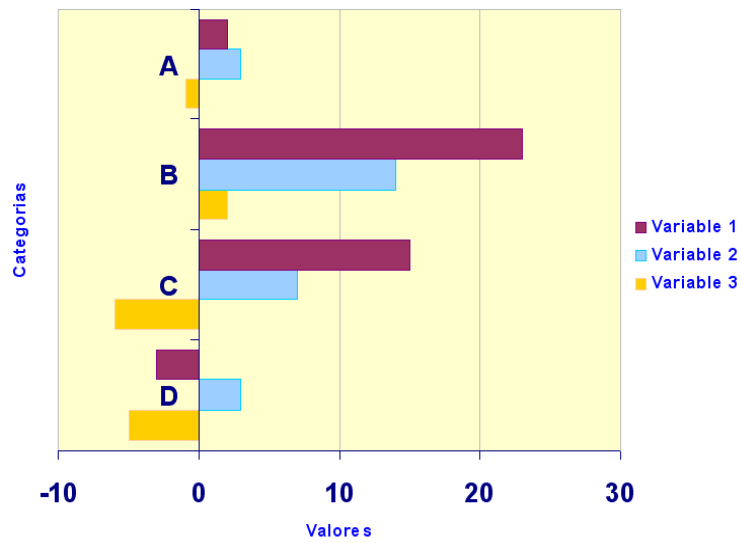


Figura 12. Gráfico de Barras. Tomado de [10]

3.4.5.5 Gráfico de Caja

Un diagrama de caja es un gráfico, basado en cuartiles, mediante el cual se visualiza un conjunto de datos.

En un gráfico que se suministra información sobre la mediana, El cuartil Q1 y Q3, sobre la existencia de atípicos y la simetría de la distribución. [12]

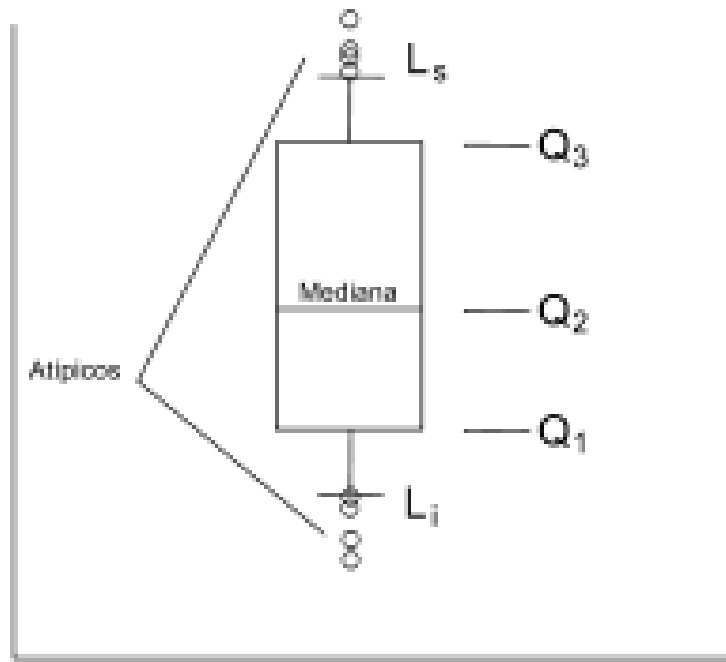


Figura 13. Gráfico de Caja. Tomado de [12]

3.4.5.6 Mapa de Calor

Un mapa de calor es una representación gráfica de datos donde los valores tomados por una variable en un mapa bidimensional son representados como colores. Esta representación se asemeja a Treemap. [13] (Figura 14)

3.4.5.7 Árbol

Un árbol es una estructura de datos ampliamente usada que emula la forma de un árbol (un conjunto de nodos conectados). Un nodo es la unidad sobre la que se construye el árbol y puede tener cero o más nodos hijos conectados a él. Se dice que un nodo a es padre de un nodo b si existe un enlace desde a hasta b (en ese

caso, también decimos que b es hijo de a). Sólo puede haber un único nodo sin padres, que llamaremos raíz. Un nodo que no tiene hijos se conoce como hoja. Los demás nodos (tienen padre y uno o varios hijos) se les conoce como rama. [14] (Figura 15)

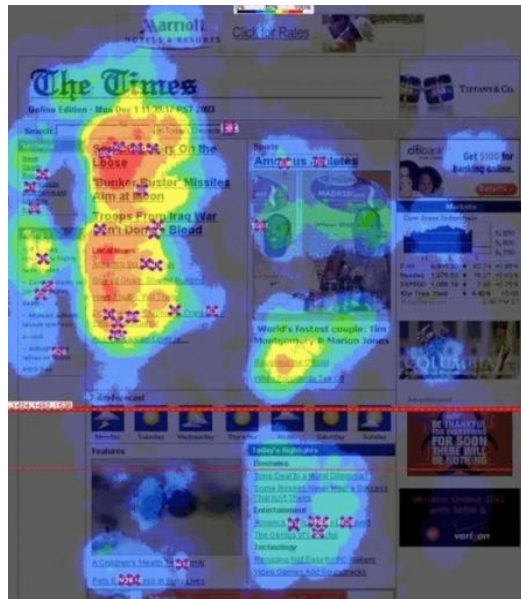


Figura 14. Mapa de Calor. Tomado de [33]

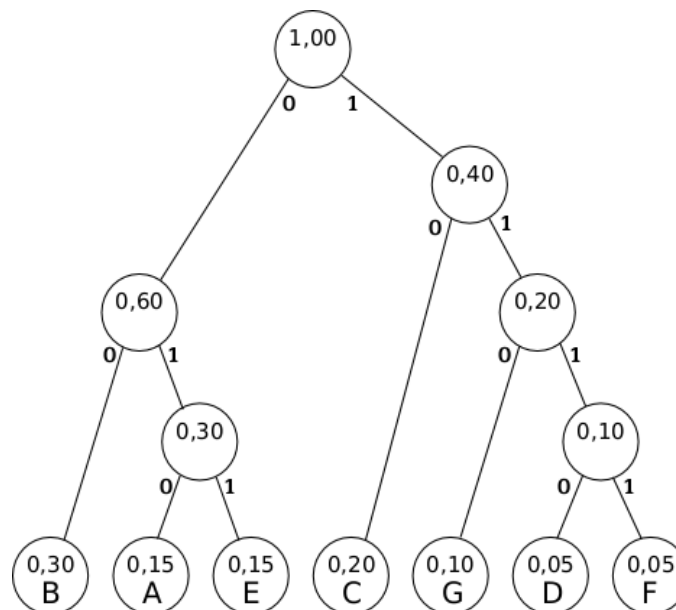


Figura 15. Árbol. Tomado de [14]

3.4.5.8 Grafo

Los grafos son artefactos matemáticos que permiten expresar de una forma visualmente muy sencilla y efectiva las relaciones que se dan entre elementos de muy diversa índole. Un grafo simple está formado por dos conjuntos:

- Un conjunto V de puntos llamados vértices o nodos.
- Un conjunto de pares de vértices que se llaman aristas o arcos y que indican qué nodos están relacionados.

De una manera más informal podemos decir que un grafo es un conjunto de nodos con enlaces entre ellos, denominados aristas o arcos.

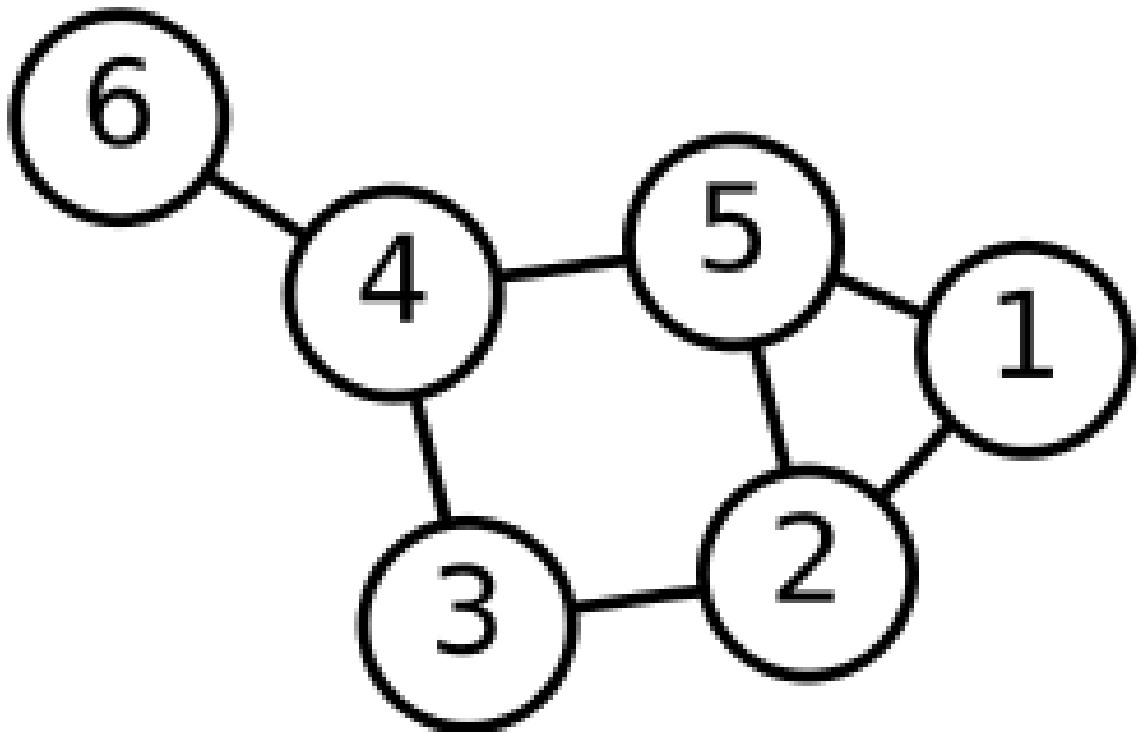
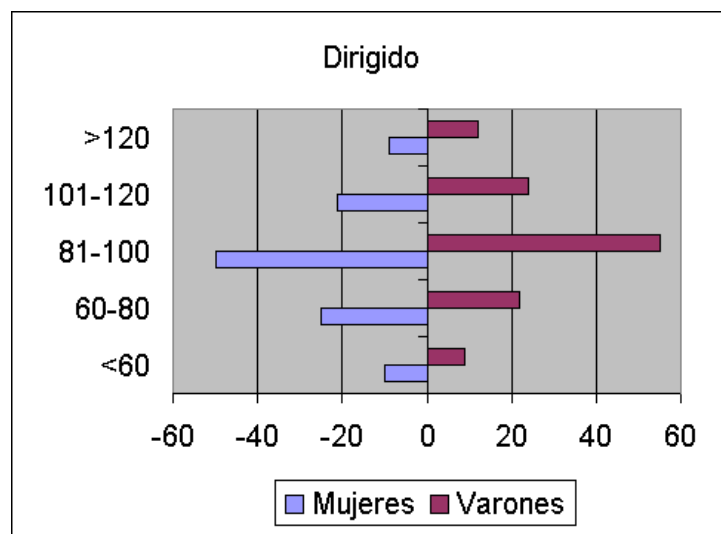
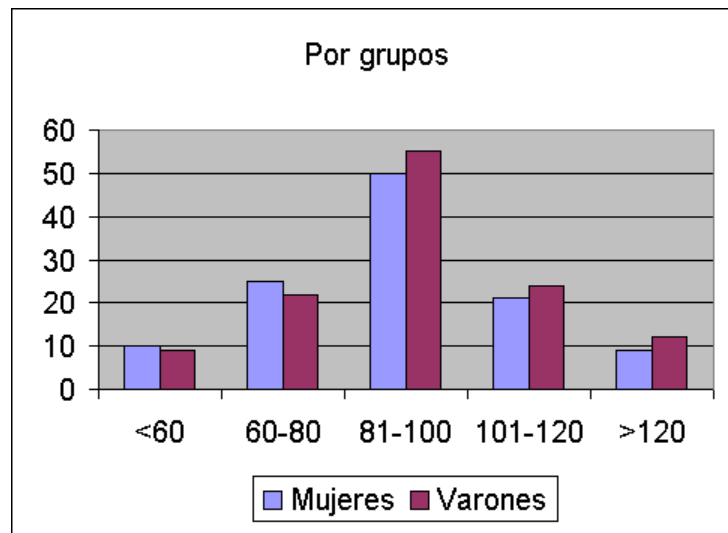


Figura 16. Grafo. Tomado de [35]

3.4.5.9 Histograma

En estadística, un histograma es una representación gráfica de una variable en forma de barras, donde la superficie de cada barra es proporcional a la frecuencia de los valores representados. En el eje vertical se representan las frecuencias, y en el eje horizontal los valores de las variables, normalmente señalando las marcas de clase, es decir, la mitad del intervalo en el que están agrupados los datos.



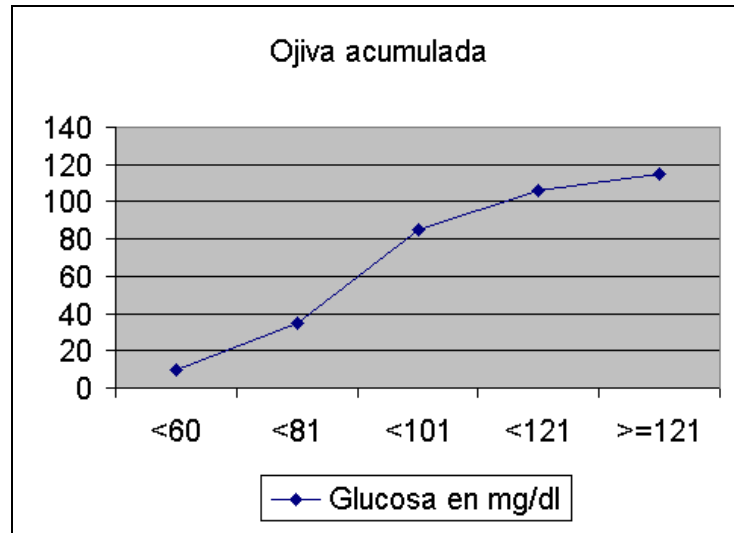


Figura 17. Histograma. Tomados de [36]

3.4.5.10 Dendrograma

Un dendrograma es un tipo de representación gráfica o diagrama de datos en forma de árbol (Dendro=árbol) que organiza los datos en subcategorías que se van dividiendo en otros hasta llegar al nivel de detalle deseado (asemejándose a las ramas de un árbol que se van dividiendo en otras sucesivamente). Este tipo de representación permite apreciar claramente las relaciones de agrupación entre los datos e incluso entre grupos de ellos aunque no las relaciones de similaridad o cercanía entre categorías. Observando las sucesivas subdivisiones podemos hacernos una idea sobre los criterios de agrupación de los mismos, la distancia entre los datos según las relaciones establecidas, etc. También podríamos referirnos al dendrograma como la ilustración de las agrupaciones derivadas de la aplicación de un algoritmo de clustering. [15]

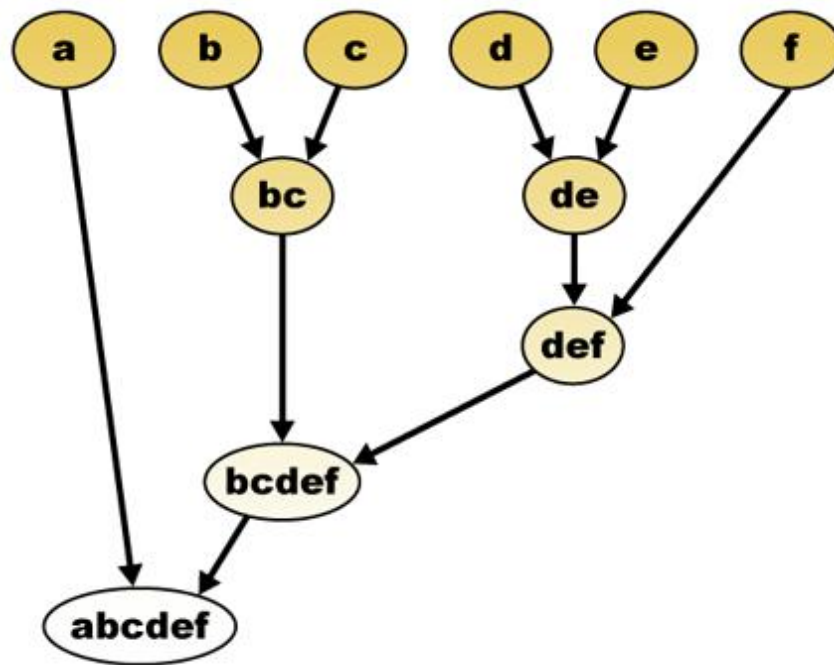


Figura 18. Dendrograma. Tomado de [15]

3.4.5.11 *Gráfico de Coordenadas paralelas*

Los gráficos de coordenadas paralelas permiten comparar en un solo plano datos con muchas dimensiones. Para cada variable (dimensión) se define un eje y se colocan en paralela, verticalmente, los ejes que corresponden a cada variable. Luego se coloca en cada eje el valor que corresponde a cada caso y se unen estos puntos mediante segmentos de recta. Obviamente no es posible representar una gran cantidad de casos, pero sí una gran cantidad de variables. Así, por ejemplo, si se quiere caracterizar diferentes tipos de automóviles, se podría indicar la cilindrada en un eje, el número de puertas en otro, el gasto de gasolina por km en un tercero, etc. Cada línea quebrada correspondería a un modelo de auto. (Este sistema fue inventado en 1985 por Alfred Inselberg). [16]

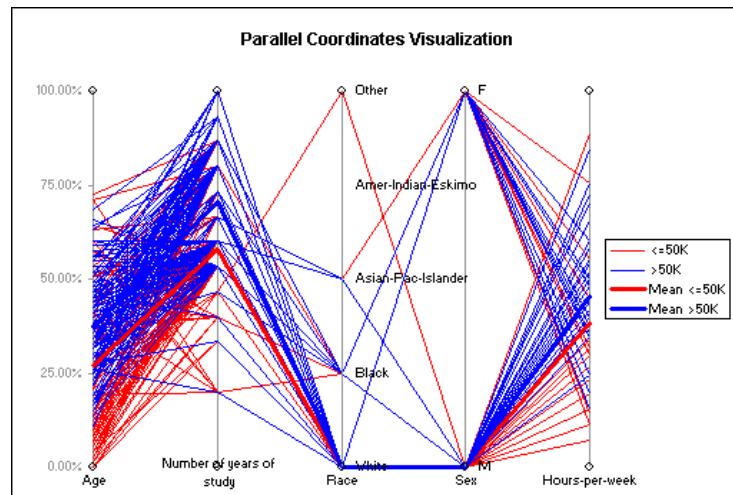


Figura 19. Gráfico de Coordenadas Paralelas. Tomado de [37]

3.4.5.12 Gráfico de Radar

Este gráfico coloca los datos de grupo, tales como países o clientes, en el perímetro del radar. El gráfico de radar entonces ubica valores numéricos, aumentando en valor, desde el centro del radar hasta el perímetro. De esta forma, el usuario puede determinar, con solo un vistazo, cómo los datos de grupo específicos están relacionados con todos los datos del grupo. [17]

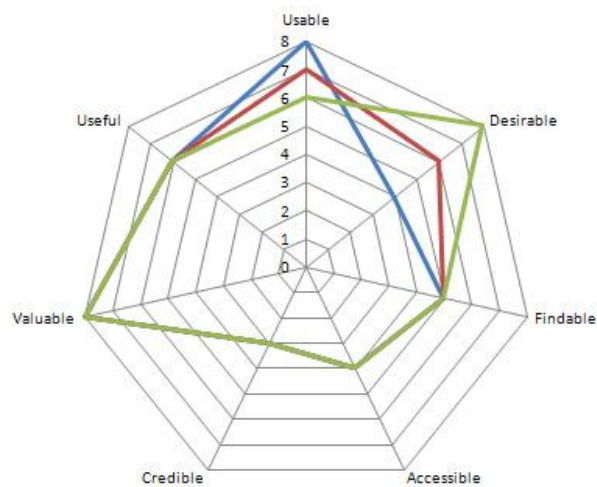


Figura 20. Gráfico de Radar. Tomado de [38]

3.4.5.13 Treemaps

Los Treemaps son un tipo de visualización ideada a principios de los 90 por Ben Shneiderman, que permite representar jerarquías de forma que se optimiza el llenado del espacio y, además, ver los atributos de la misma e identificar patrones anómalos o propiedades de la jerarquía usando colores.[18]

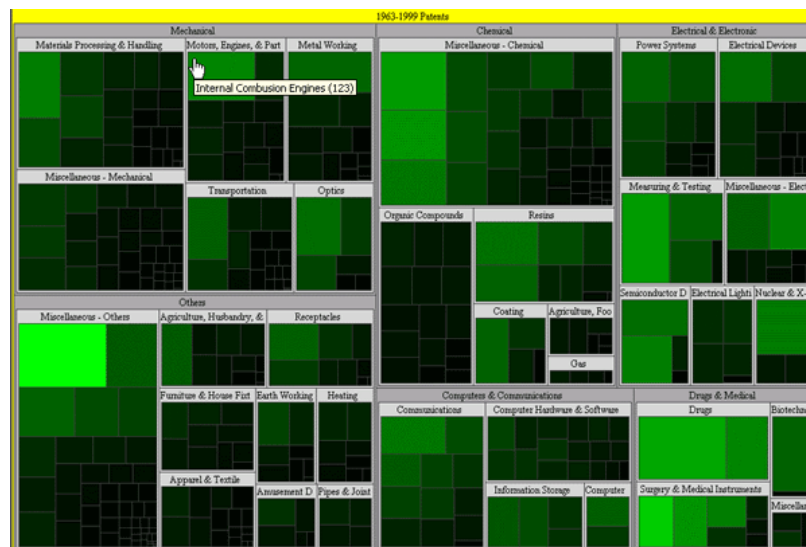
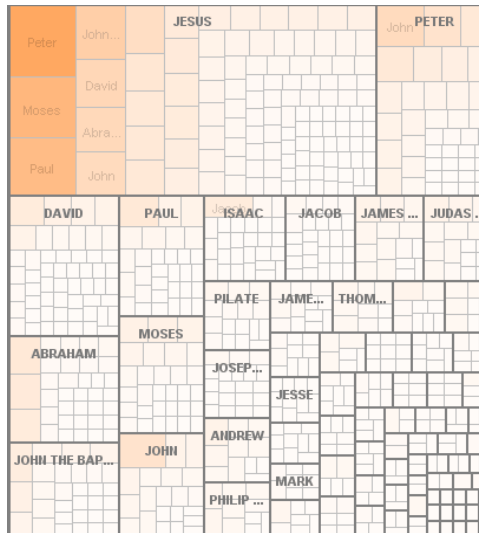


Figura 21. Treemap. Tomado de [39]

3.4.6 Refinar

El conocimiento de cómo están organizados los datos que se van a trabajar no es suficiente para determinar el método de visualización más adecuado.

Muchas de las practicas utilizadas en diseño gráfico proveen una guía que ayudan a encontrar y redactar las preguntas que se debe hacer para determinar con mayor precisión y claridad cuál es la mejor forma para representar las características más relevantes de los datos a trabajar.

Una de las mejores referencias existentes en el área del refinamiento visual entorno a representación de datos es la serie de libro de Tufte “*The Visual Display of Quantitative Information*” (1983), “*Envisioning Information*” (1990), “*Visual Explanations: Images and Quantities, Evidence and Narrative*” (1997).

3.4.6.1 Contraste y diferenciación

Existen varias técnicas que permiten alcanzar uno de los objetivos del diseño de visualización de información el cual es comparar características de diferentes elementos. A continuación se presentan algunas técnicas usadas para esta comparación.

- **Contraste:** El ojo del ser humano tiene la característica de buscar contraste en todos los espacios que visualiza, lo que convierte esta técnica en una herramienta muy útil para diferenciar elementos.
- **Jerarquía:** Directamente relacionado encontramos esta técnica, donde los elementos son mostrados en un orden definido que permite definir la importancia que tiene un elemento respecto a otro.

- Agrupación: Se trata de organizar elementos con características comunes o que tengan un elemento común a todos (gravitacionar en torno a un elemento).

3.4.6.2 Características visuales usadas en la diferenciación.

Las características de mayor acogida por los diseñadores para crear diferenciación y generar contraste en las representaciones gráficas son: el tamaño de los elementos, su color o colores característicos y la ubicación de los elementos en un espacio o plano.

- Peso y Tamaño: Es un elemento que debe ser utilizados con especial cuidado ya que puede ser un excelente resaltador de características, pero al ser usado en exceso puede distraer con facilidad al usuario.
- Color: El color es comúnmente usado para diferenciar categorías de datos, pero su uso puede presentar muchos problemas dado que:
 - El ojo del ser humano común solo distingue 5 colores separados, por lo que cuando se trate de representar muchas categorías con diferentes colores no se conseguirá este objetivo de forma adecuada. (Via Ware 2002).
 - Puede llegar a usarse de una forma inadecuada del color, por ejemplo al pintar dos categorías del mismo color donde sus elementos no tienen relación alguna.
 - En algunas ocasiones no hay medios físicos para reproducir los colores.
 - Algunas personas con problemas de visión o percepción distorsionada de los colores no permiten que esta técnica sea útil en algunos grupos determinados.

- Podemos cambiar color por patrones de diseño en blanco y negro, o iconos que nos permitan identificar de una manera más apropiada las diferencias entre uno y otro.
- Ubicación: La ubicación permite generar una jerarquía visual, el ordenamiento o agrupación de los elementos. Además, al realizar una ubicación adecuada de los elementos permite conducir o guiar la lectura del mismo, logrando incluso que el usuario lea el diagrama en su totalidad.

3.4.7 Interacción

Esta etapa hace referencia a como los datos se relacionan entre sí dentro del entorno gráfico que se diseño en la representación. Esta tipo de procesos son estudiados ampliamente por Visible Language Workshop (VLW).

En esta etapa también se tiene en cuenta como los usuarios se controlan y se relacionan con la representación de los datos. Este proceso es ampliamente estudiado por Human Computer Interact (HCI).

3.4.7.1 Interacción de datos

Existen algunas reglas o consejos que pueden hacer que se diseñe de una forma más adecuada el proceso de interacción de los elementos entre ellos tenemos reglas de distribución y ubicación basada en la física.

- Reglas básicas de distribución: Aquí se encuentran directrices relacionadas con la definición de los elementos dentro del espacio de la visualización. Se aconseja:

- Conectores gráficos que permitan la conexión entre elementos de forma clara y sencilla.
 - Los elementos deben distribuirse en áreas definidas.
 - Manejar espacios proporcionales entre ellos.
 - Hacer uso de escalas para mostrar los datos.
 - Sin espacio suficiente, mejor opte por listar los elementos
-
- Ubicación basada en la física: Existen reglas basadas en series de masas y elasticidad que permiten que los elementos interactúen de cierta forma. Se tiene las leyes de elasticidad de Hook quien define que un elemento se puede atraer (acercar) o desviar (alejarse) con base en un criterio definido.

3.4.7.2 Interacción de Usuario

Los llamados controles de vista y movimiento permiten desplazarse en un diagrama completo cuando este es muy grande para ser visualizado en su totalidad. Entre este tipo de controles los más conocidos son las barras de desplazamiento, además también entran dentro de esta categoría todos los controles que permitan resaltar elementos específicos.

Existen efectos, como escalado o zoom o ojo de pez, que permiten trabajar mejor con un diagrama ampliando áreas del diagrama definidas por el usuario, o enfocar espacios para un mejor análisis.

Entre las recomendaciones para la interacción entre el usuario y el sistema se aconseja que el usuario solo tenga acceso a las opciones que necesite o las más importantes o las que la mayoría de posibles usuarios necesiten. Es mejor que un usuario cuente con una visualización sencilla y que se pueda personalizar que una visualización compleja e inmanejable.

4 MANEJO DE TRAZABILIDAD EN LAS HERRAMIENTAS ACTUALES

4.1 ENTERPRISE ARCHITECT

Enterprise Architect es una herramienta para la ingeniería de software asistida por computador (CASE por sus sigla en Inglés) para el diseño y construcción de sistemas software, modelado del proceso negocios y, en general, para cualquier modelado que se desee realizar.

Enterprise Architect está basada en la especificación UML 2.1. UML define un lenguaje visual que permite modelar un sistema o dominio en particular sin importar si existe o es tan solo una idea.

Enterprise Architect es una herramienta que cubre todos los aspectos del ciclo de desarrollo de sistemas, permitiendo manejar la trazabilidad del proyecto de forma completa, desde la fase inicial del diseño y a través del desarrollo y mantenimiento. También provee soporte para pruebas y control de cambios.

4.1.1 Trazabilidad en Enterprise Architect

Enterprise Architect permite realizar el seguimiento de trazabilidad por medio de la relación *Realización* y la relación *Trace* las cuales permiten evidenciar la evolución que un artefacto UML tiene durante todo el proceso de desarrollo de un proyecto Software.

Durante el desarrollo de un proyecto UML haciendo uso de la herramienta, se puede crear las relaciones de trazabilidad de dos formas diferentes: la primera creando un nuevo artefacto UML relacionándolos directamente con la relación

seleccionada, en este caso la relación *Realización* (Figura 22). La segunda creando la relación entre dos elementos previamente creados usando la relación que se desee, en este caso la relación *Realización* y la relación *Trace*. En este último caso existen dos formas de manera gráfica (Figura 23) y por medio de la matriz de trazabilidad (Figura 24).

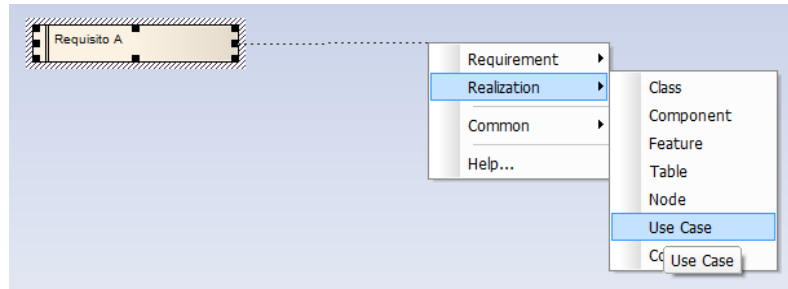


Figura 22. Creación de un artefacto UML nuevo a partir de una relación Realización

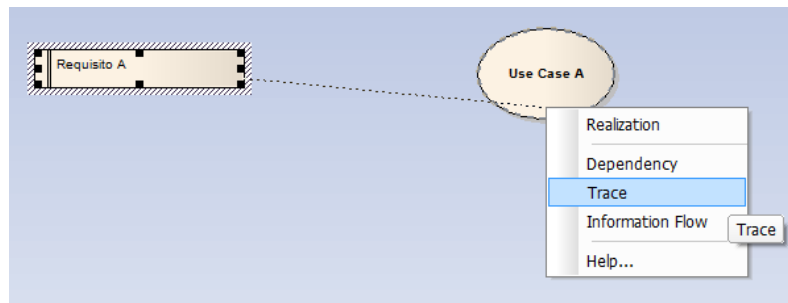


Figura 23. Creación de una relación Trace a partir de dos artefactos UML previamente creados

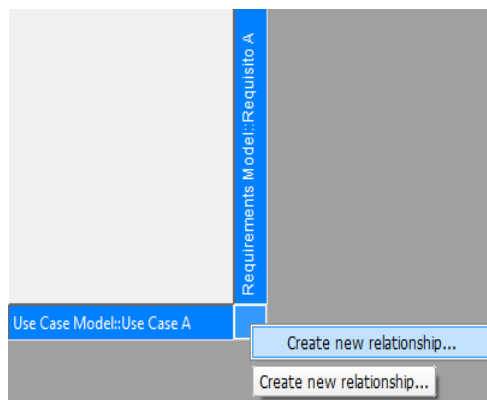


Figura 24. Creación de una relación Realización haciendo uso de la matriz de trazabilidad

También, por medio de la matriz de trazabilidad, se puede hacer la revisión de cómo los elementos UML utilizados en un proyecto determinado se encuentran relacionados entre sí, tanto a nivel vertical (Figura 25) como horizontal (Figura 26).

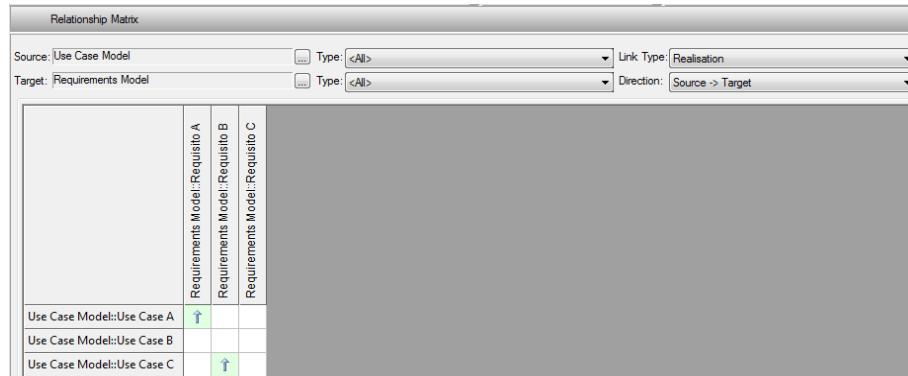


Figura 25. Enterprise Architect - Matriz de Trazabilidad con relaciones horizontales

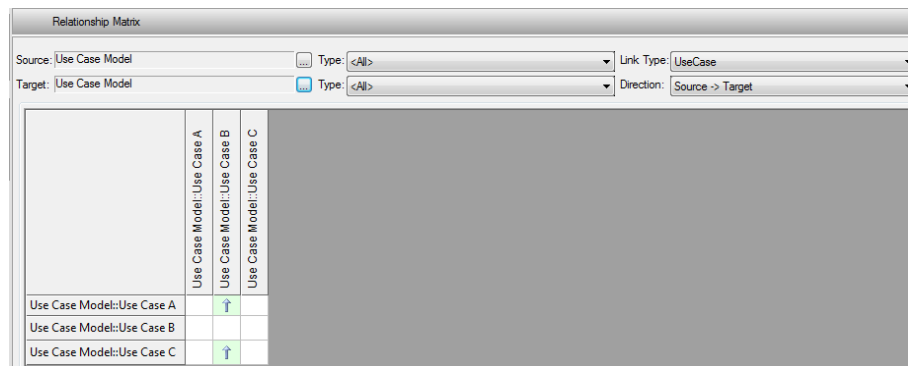


Figura 26. Enterprise Architect – Matriz de Trazabilidad con relaciones verticales

Enterprise Architect no ofrece un diagrama específico para el manejo de trazabilidad, lo que deja la matriz de trazabilidad como única herramienta para la gestión de trazabilidad, cosa que no permite realizar un seguimiento de la evolución de los artefactos de manera que se puede realizar un análisis del impacto del cambio adecuado dentro de un proyecto que utilice esta herramienta CASE.

4.2 OBJECTEERING UML

Objecteering UML Modeler es la principal herramienta del grupo Objecteering. Esta herramienta provee una cobertura total de los artefactos UML para el modelado de sistemas. Usando Objecteering UML Modeler, se puede:

- Crear y editar modelos en el explorador de modelos, además de los diagramas UML
- General toda la documentación del proyecto que se desee
- Generar todo el código para la aplicación en diferentes lenguajes de programación (Java, C++, C#, ...)
- Verificar la calidad y precisión del modelo por medio de Objecteering usando el verificador de consistencia y métricas. [24]

Algunas de las características más importantes existentes en esta herramienta son:

- Integración con UML2, BPMN, y Enterprise Architect
- Completo manejo de MDD (model driven development), desde los objetivos de análisis hasta la codificación y pruebas.
- Transición transparente entre fases, incluyendo manejo de Trazabilidad entre necesidades, requisitos, modelado y código.
- Modelo de verificación de consistencia dinámico y personalizado.
- Integración con la tecnología MDA, permitiendo mejorar las características de transformación de modelos y generación de código.
- Un repositorio único y compartido para todos los modelos durante las diferentes fases de desarrollo.
- Documentación automática de productos y publicación en documentos Web de los mismos.

- Integración y sincronización con Eclipse IDE.
- Herramientas de apoyo para el trabajo en equipo, manejo de modelos actuales, fusión de modelos e integración con el gestor de versiones.
- Lenguaje, metodología y plataforma independiente. (Linux, Windows)

4.2.1 Trazabilidad en Objecteering UML

Esta herramienta tiene una única forma de manejar la trazabilidad, e incluso las relaciones horizontales del modelo, y se trata de *Fast Link Creator* un editor gráfico que permite crear y visualizar las relaciones que se desee, entre ellas la relación de *Trazabilidad*. Por medio de *Auxiliry View* se elimina las relaciones que se hayan creado en *Fast Link Creator*.

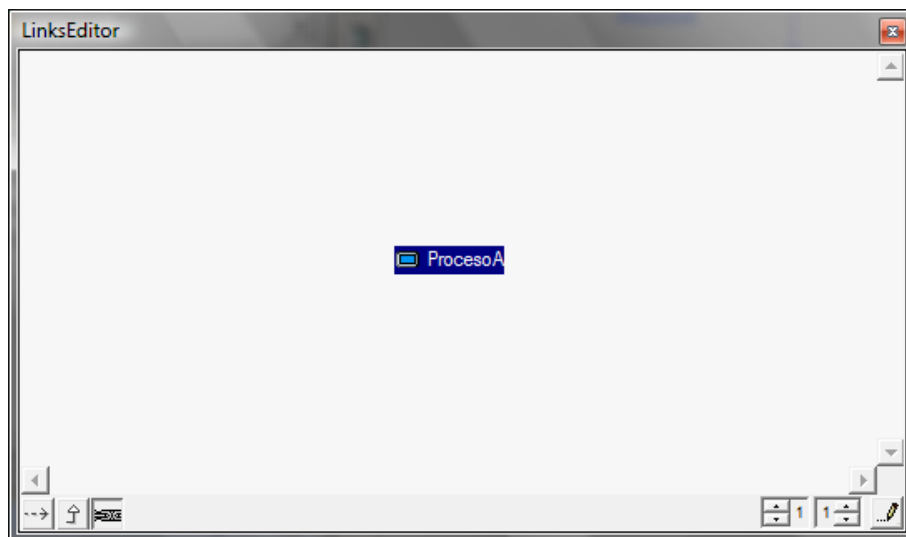


Figura 27. Objecteering Modeler – Fast Link Creator

Si se desea crear una relación de trazabilidad entre dos elementos debe hacerse lo siguiente:

1. Seleccione el primer elemento de la relación, esto hará que este elemento aparezca en el Fast Link Creator (Figura 27).
2. Seleccione el tipo adecuado de vista de relación. En el caso que se está estudiando se selecciona la vista de relación de trazabilidad (Figura 28).
3. Dando doble click sobre el editor habilite el modo edición. Se reconoce el estado porque el área de trabajo de esta ventana cambia su fondo liso por uno punteado (Figura 29).
4. A continuación busque el elemento que desee relacionar en el UML Explorer y arrástrelo hasta el Fast Link Creator creando así la relación de trazabilidad (Figura 30).



Figura 28. Objecteering Modeler – Relaciones de Fast Link Creator, la ultima Trazabilidad

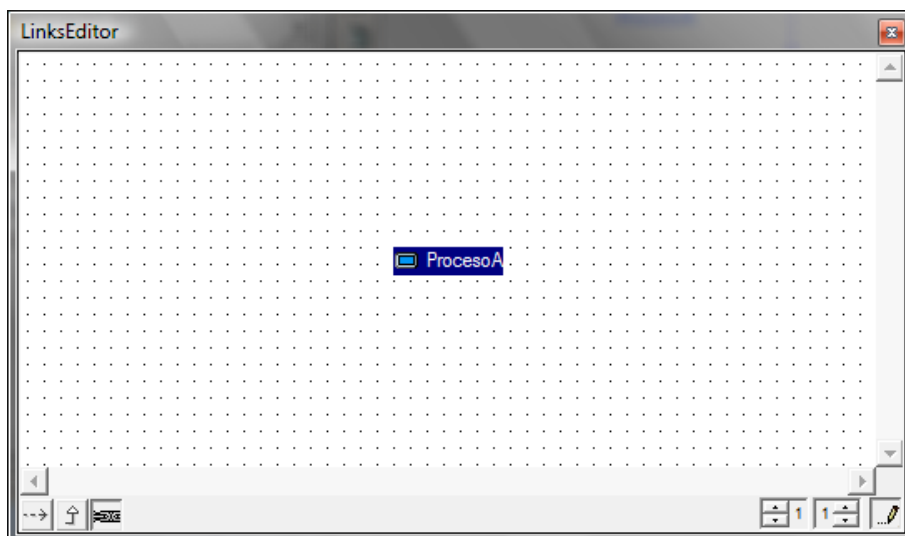


Figura 29. Objecteering Modeler – Fast Link Creator en estado de Edición

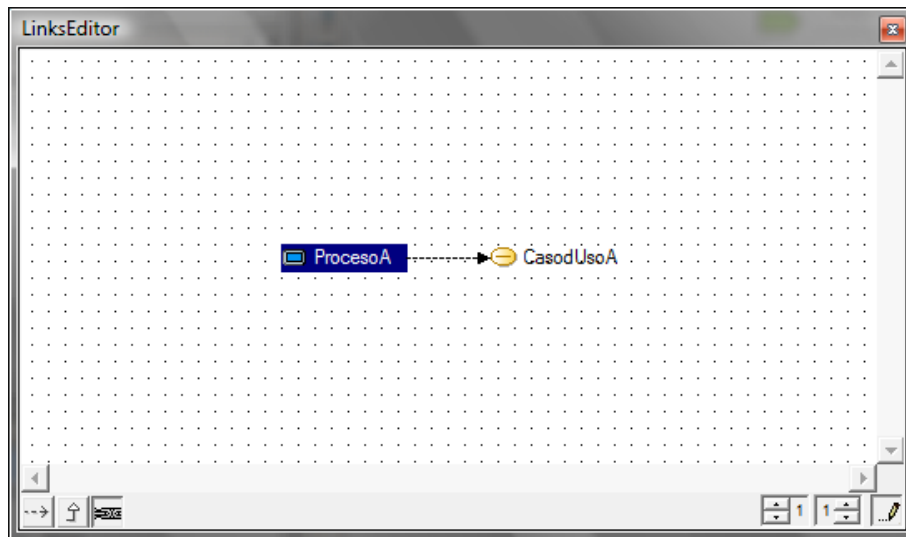


Figura 30. Objecteering Modeler - Fast Link Creator mostrando una relación de Trazabilidad

Podemos ver esta relación de trazabilidad también en la ventana AuxiliaryView, en la pestaña AuxiliaryProperties (Figura 31). La sintaxis que se nos presenta es, primero la relación (muestra el icono de relación de trazabilidad) a continuación el elemento B (el cual fue relacionado) y finalmente el elemento A (elemento inicial).

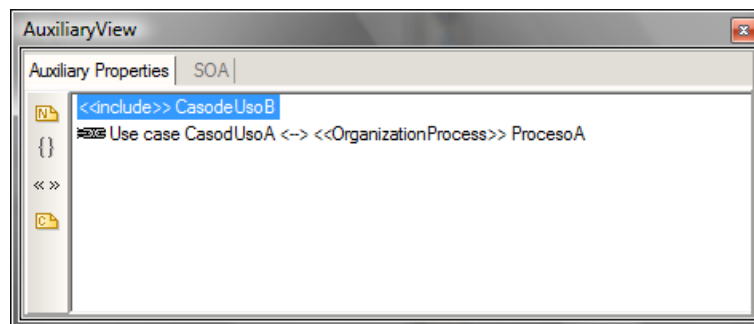


Figura 31. Objecteering Modeler – Auxiliary View

En esta última vista es donde se puede eliminar la relación creada. Tan solo se selecciona la relación que se desee borrar, se da click derecho y allí la opción eliminar (Figura 32) o seleccionar la relación a borrar y presionar la tecla suprimir.

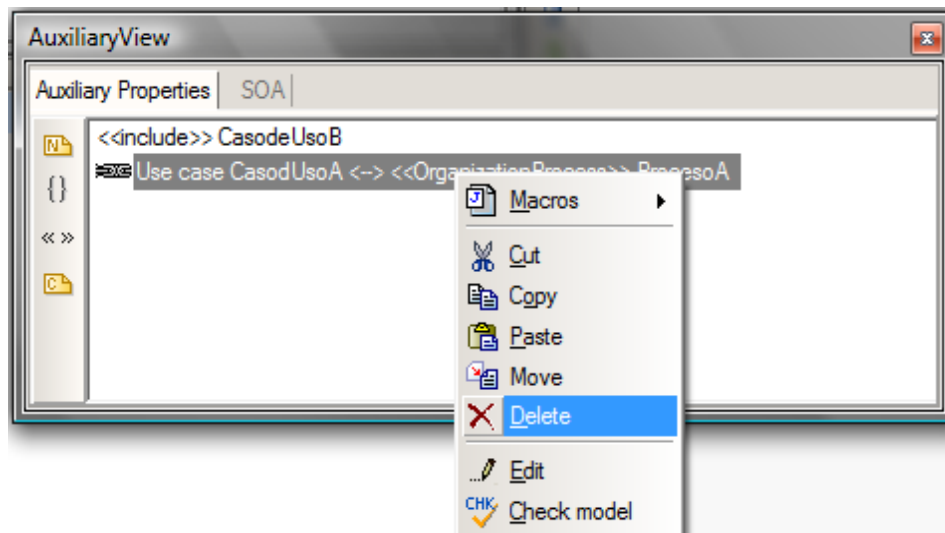


Figura 32. Objecteering Modeler – Auxiliary View Borrado de relación.

La herramienta permite de forma manual la creación de un grafo de trazabilidad lo que hace que esta información sea mucho más significativa para la construcción de un análisis de impacto del cambio. También se debe tener en cuenta que la generación manual del gráfico hace que esta tarea sea mucho mas tediosa e incluso puede llegar a ser imprecisa.

4.3 BORLAND CALIBERRM

Borland® CaliberRM™ es una herramienta para la gestión de requisitos de software la cual facilita la colaboración, el impacto de análisis y la comunicación, permitiendo a los equipos de desarrollo entregar una versión clave del proyecto con gran precisión y predictibilidad. CaliberRM además ayuda a organizaciones pequeñas, grandes y distribuidas a asegurar que las aplicaciones que construyen cumplen con todos las necesidades de los usuarios por medio de comunicación total entre analistas, desarrolladores, testers y cualquier persona que tenga relación directa con el proyecto, a través de todo el ciclo de vida. [23]

Algunas de las características más importantes existentes en esta herramienta son:

- Repositorio centralizado: Borland CaliberRM provee un repositorio central donde se almacenan todos los requisitos del proyecto.
- Adaptabilidad: Borland CaliberRM se adapta a la medida de los procesos, dando velocidad y agilidad en el proceso de gestión de requisitos.
- Trazabilidad de requisitos de principio a fin: La arquitectura abierta de Borland CaliberRM permite conectar los requisitos de software hacia una gran variedad de artefactos a través de todo el ciclo de vida del proyecto.
- Análisis de impacto de principio a fin: Múltiples métodos para la visualización de Trazabilidad ayudan a los usuarios a entender inmediatamente el alcance del análisis, necesario para medir el impacto de un cambio específico.
- Integración: con otros productos de Borland permitiendo relacionar los requisitos propuestos a través de las diferentes herramientas utilizadas durante el ciclo de vida.

4.3.1 Trazabilidad en Borland CaliberRM

CaliberRM realiza la gestión de requisitos asociados a uno o varios proyectos, esto permite que CaliberRM permita crear relaciones de trazabilidad entre cualquier requisito en un mismo proyecto u otros gestionados por la herramienta de la herramienta y además cuenta como característica la integración con otras herramientas, herramientas con las cuales puede también realizar trazabilidad desde los requisitos hasta sus artefactos propios.

En el ejemplo presentado para mostrar como la herramienta gestiona trazabilidad se hizo la integración con *Borland Together*, herramienta para el modelado de proyectos software.

Después de que han sido creados los requisitos, se puede crear las relaciones de traza entre los diferentes requisitos de un mismo proyecto o de otro que se encuentre activo dentro de CaliberRM. Para esto se debe seleccionar el requisito a trazar del árbol de requisitos ubicado en la parte izquierda de la herramienta y allí seleccionar la pestaña *traceability* que es el lugar donde se puede crear, modificar y eliminar las relaciones de trazabilidad (Figura 34).

Aquí se observa las relaciones existentes que llegan hacia el requisito y que salen del requisito que se está observando.

Para crear una relación de traza entre dos requisitos se debe dar click sobre el botón *Modify*, allí aparecerá una ventana para la selección del requisito con el cual se desea establecer la relación de trazabilidad (Figura 35).

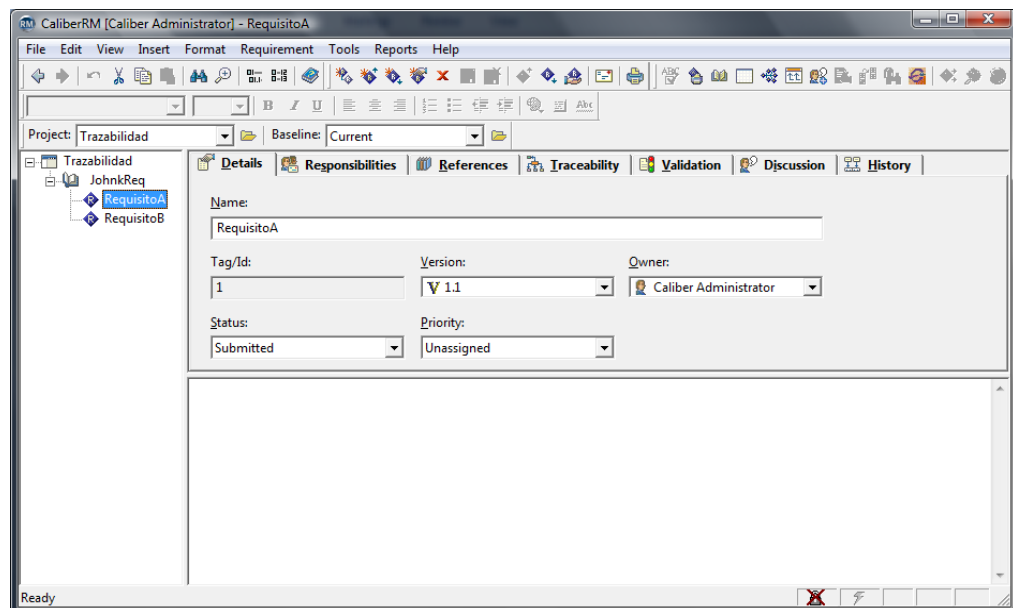


Figura 33. CaliberRM – Pantalla de edición de requisitos

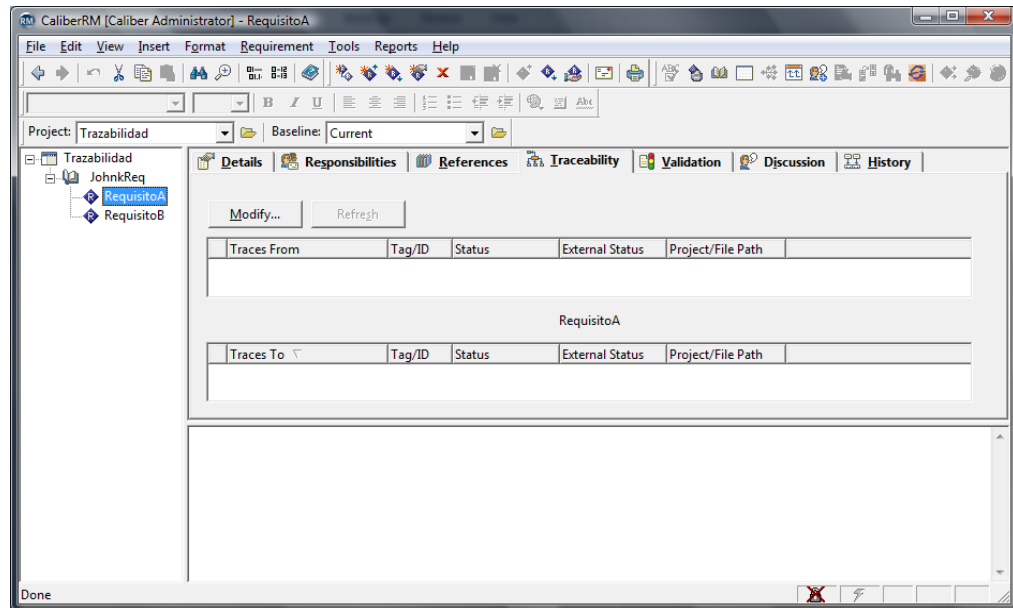


Figura 34. CaliberRM – Pantalla de edición de trazabilidad de requisitos

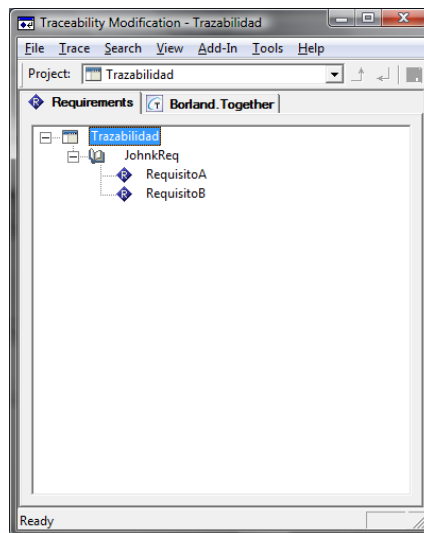


Figura 35. CaliberRM – Selección de requisito con el cual se desea realizar una traza

Al seleccionar el requisito, en la parte superior derecha se activan los botones que establecen la relación, el primero que traza desde el requisito (Figura 36) y el segundo que traza hacia el requisito (Figura 37). Se presiona el botón adecuado

según la dirección que se le desee dar a la relación de trazabilidad para posteriormente presionar el botón de guardado, ubicado junto al botón *Trace to*, y comentar el cambio realizado (Figura 38). Finalmente en la pantalla inicial se puede observar el registro del cambio realizado (Figura 39).

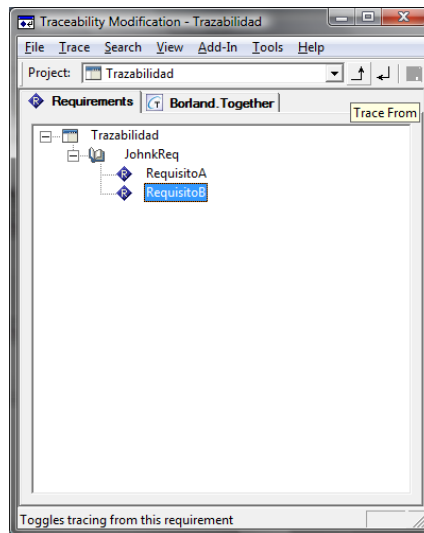


Figura 36. CaliberRM – Botón que traza desde el requisito

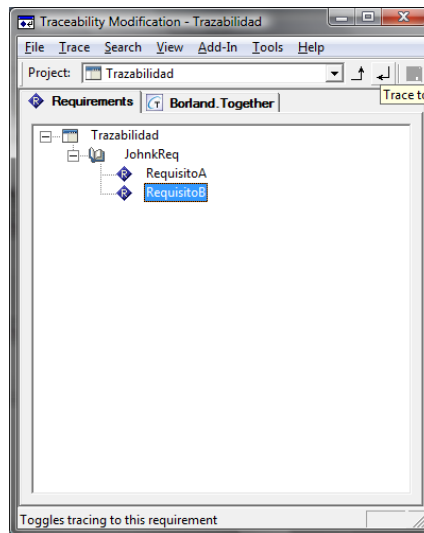


Figura 37. CaliberRM – Botón que traza hacia el requisito

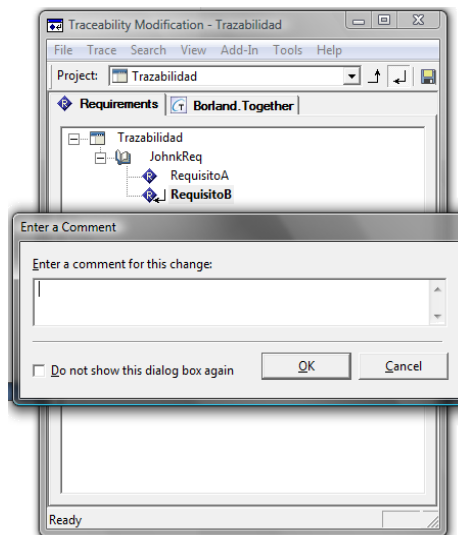


Figura 38. CaliberRM – Comentar el cambio realizado

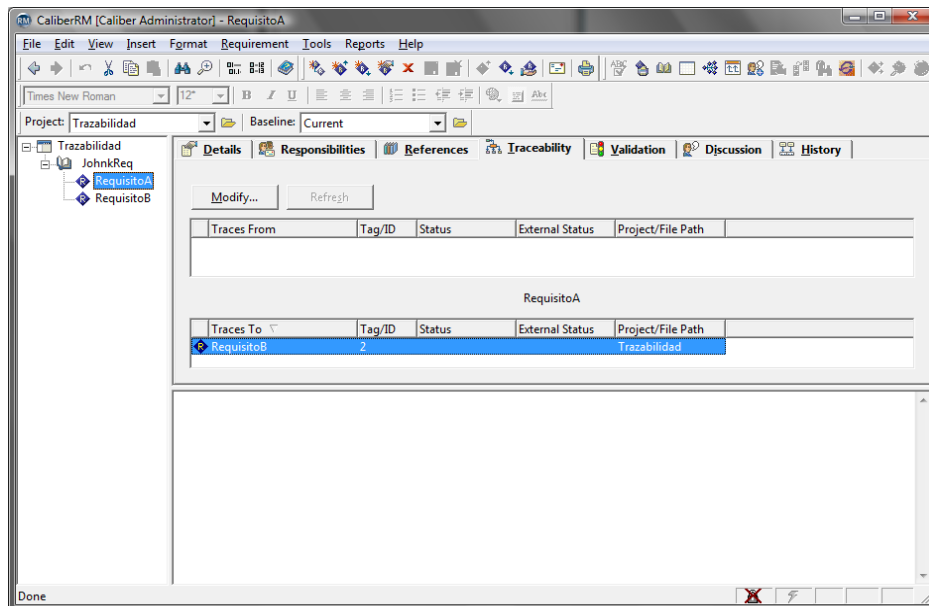


Figura 39. CaliberRM – Pantalla de edición de trazabilidad de requisitos listando relaciones de traza

Ahora bien si se desea crear las relaciones de trazabilidad entre los requisitos y otros elementos de modelado UML se debe instalar el cliente CaliberRM en el entorno de modelado Borland Together, por medio de esta utilidad se puede crear

las relaciones de trazabilidad hacia el proyecto de requisitos en Borland CaliberRM.

Cuando se desee crear una relación de trazabilidad se debe dar click derecho el artefacto UML del modelo y seleccionar la opción Requirements y allí Manage Traces (Figura 40).

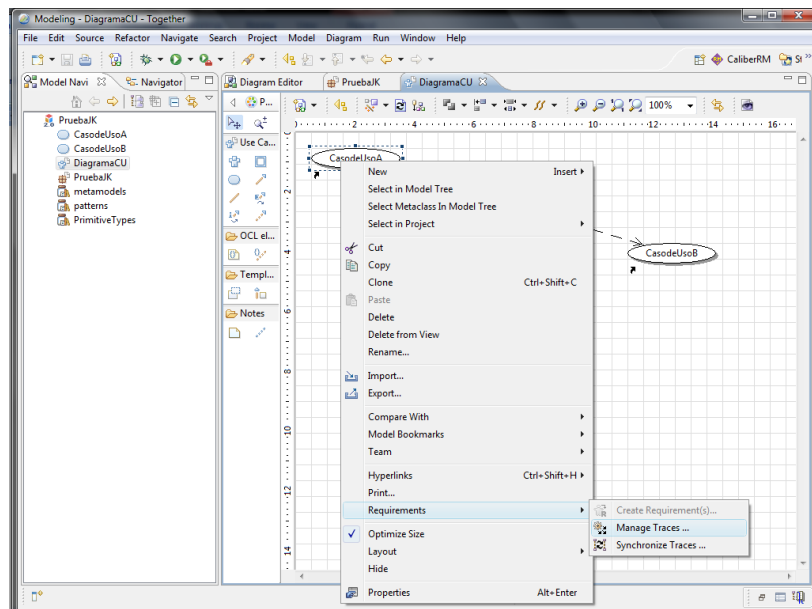


Figura 40. Together – Menú de despliegue de caso de uso, seleccionando Manage Traces

Al seleccionar esta opción, aparece la ventana donde se crean las relaciones de traza entre el elemento seleccionado y los requisitos en CaliberRM. Aquí se selecciona el proyecto donde se encuentren los requisitos, se despliega el árbol y allí aparecerán los diferentes requisitos creados, se selecciona el requisito con el cual se desee crear la relación y se presiona el botón Add con lo que se crea la relación. Finalmente se debe presionar el botón ok cuando se haya terminado de seleccionar los requisitos relacionados (Figura 41).

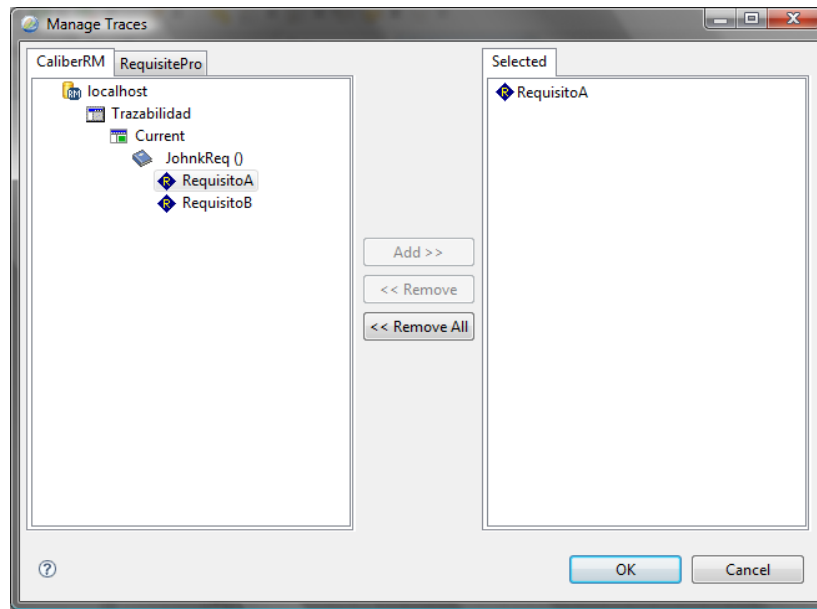


Figura 41. Together – Selección de requisitos para crear relación de traza con un proyecto CaliberRM

Para poder visualizar las relaciones de trazabilidad que existen en los diferentes proyecto, Borland CaliberRM ofrece una matriz de trazabilidad donde haciendo cruces entre elemento representa las relaciones existentes y un grafo de trazabilidad el cual muestra como están relacionados los elemento de manera gráfica.

Para usar la matriz de trazabilidad se debe ir al menú y allí seleccionar Traceability Matrix (Figura 42). Posterior a esto se muestra la ventana de filtro de parámetros de la matriz, aquí se debe seleccionar que elementos aparecerán en las filas y cuales en las columnas (Figura 43). Debe notarse que existe una pestaña que permite que los elementos seleccionados sean parte de los modelos construidos con Borland Together. Finalmente se presiona el botón OK y aparece la matriz de trazabilidad con los elementos escogidos (Figura 44).

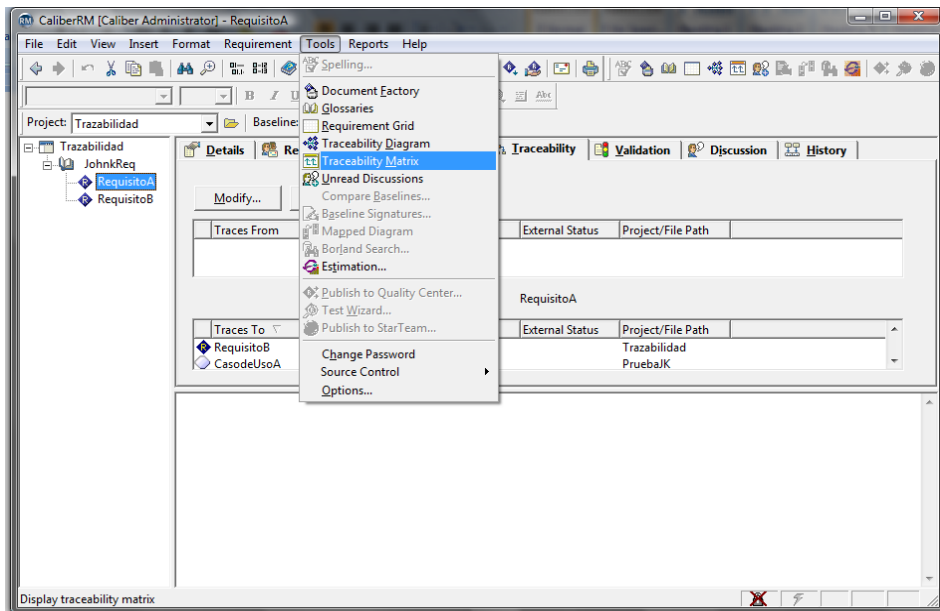


Figura 42. CaliberRM – Menú Traceability Matrix

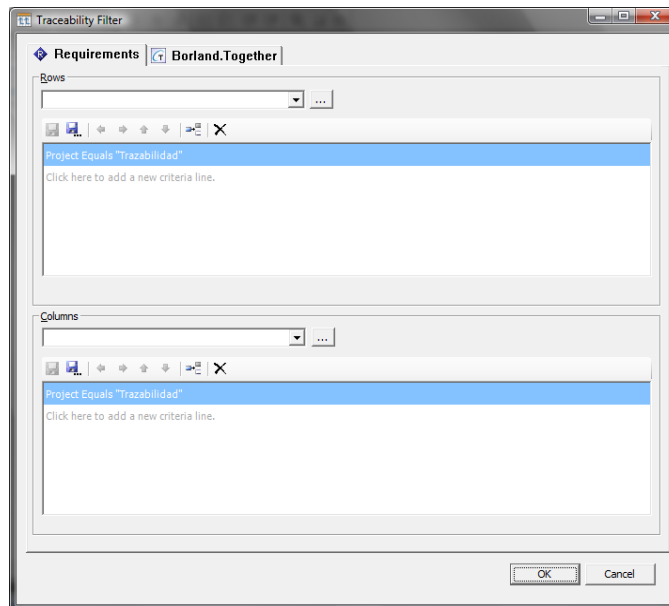


Figura 43. CaliberRM – Ventana de filtro de parámetros de la matriz de trazabilidad

Para hacer uso del grafo de trazabilidad, se debe seleccionar, en el árbol del proyecto, el requisito que se desea observar paso seguido ir al menú Tools y allí

seleccionar Traceability Diagram (Figura 45). Con lo que automáticamente se construirá el grafo de trazabilidad partiendo del elemento seleccionado (Figura 46).

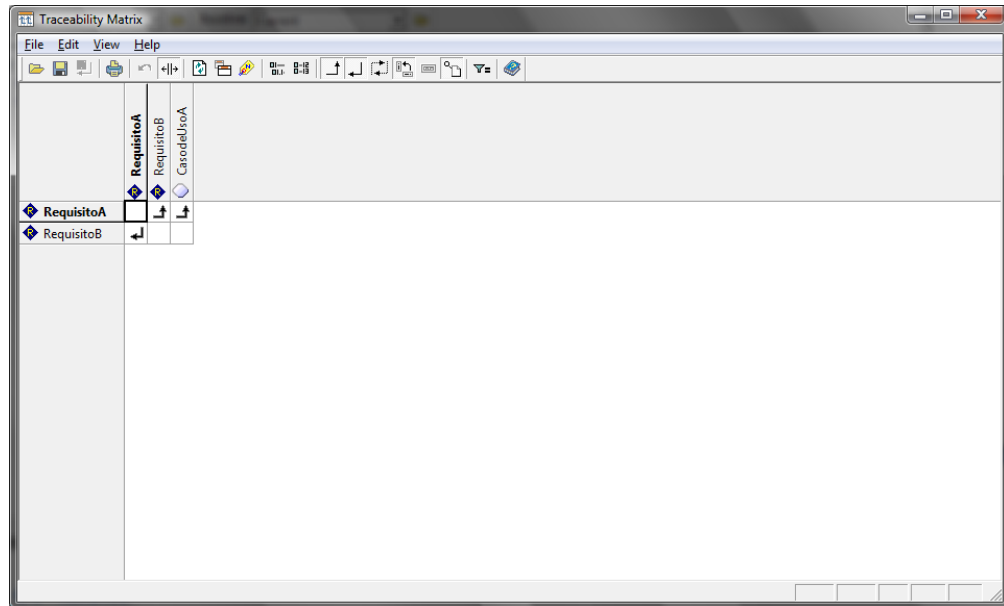


Figura 44. CaliberRM – Matriz de trazabilidad

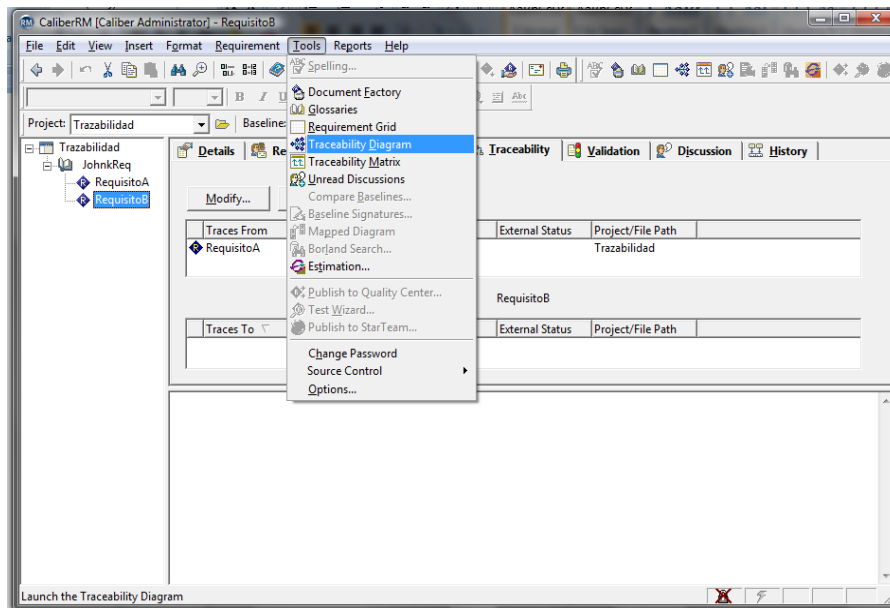


Figura 45. CaliberRM – Menú Traceability Diagram

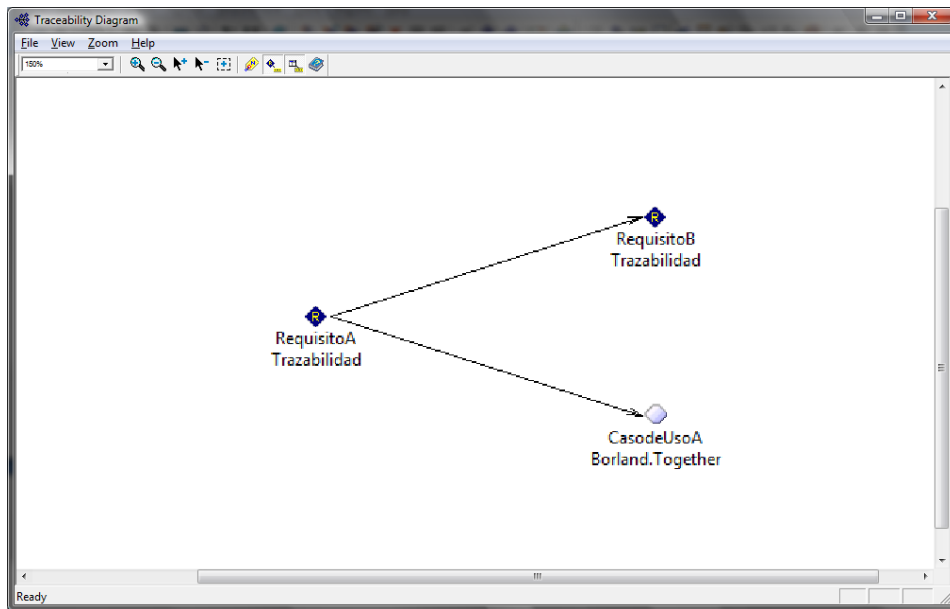


Figura 46. CaliberRM – Grafo de trazabilidad

Esta herramienta es muy completa a nivel de requisitos, la falla que encuentro es la necesidad de ser complementada con otros aplicativos y que existe la exigencia de integrarlos por medio de paquetes para que funcione de forma avanzada dentro de un proyecto de modelado.

4.4 AR2CA

AR2CA (Arquitectura de Refinamiento y Reutilización de Componentes de Análisis) es una herramienta CASE que tiene como principal objetivo poner al alcance de los desarrolladores de software una herramienta que apoye el desarrollo de aplicaciones bajo el enfoque Orientado a Objetos, utilizando la notación UML. [22]

AR2CA ha sido desarrollada por el Grupo de Ingeniería de Software de la Universidad EAFIT y ha recibido apoyo de dos proyectos cofinanciados por Colciencias. El primer proyecto titulado “Extensiones en herramientas CASE con

énfasis en formalismos y reutilización” extendió la funcionalidad de la herramienta para manejo multiusuario y edición de expresiones OCL. El segundo titulado “Marco Metodológico para el Desarrollo Orientado a Aspectos – MMEDUSA”, del cual hace parte este trabajo de grado, pretende dar a la herramienta la capacidad de apoyar el diseño de aplicaciones orientadas por aspectos, a través del modelado de concerns y sus relaciones de trazabilidad.

La herramienta está construida en Java (versión 1.5), como una aplicación multiusuario que trabaja como cliente desconectado (Java Swing) y que permite realizar conciliación de modelos de manera asincrónica soportándose en los servicios provistos por un servidor de CVS (CVSNT 2.0 51d). Maneja información para administración de usuarios en una base de datos (MySQL). La herramienta integró la especificación UML 2.0 desarrollada por el grupo Eclipse1 y la adoptó para que funcione como un módulo articulado a la capa de presentación y control propia de AR2CA. [23]

4.4.1 Trazabilidad en AR2CA

En la herramienta CASE AR2CA actualmente no existe un módulo que permita la gestión de trazabilidad de un proyecto software, pero si existe un control de cómo un artefacto UML evoluciona a través de todo el proceso de vida de un proyecto software. Esto se hace principalmente por medio de diálogos que permiten generar nuevos diagramas a partir de los diagramas o elementos existentes, utilizando reglas de transformación definidas en XML. Esta transformación semiautomática genera una traza de la operación que se realizó y de esta manera se conservan las relaciones evolutivas que se aplicaron.

El objetivo de este trabajo de grado es la construcción de un módulo que permita a AR2CA tener las herramientas que le permitan gestionar la trazabilidad de sus modelos tanto a nivel conceptual como a nivel gráfico.

4.5 RESULTADOS DEL ANÁLISIS

El principal resultado obtenido de la revisión de las diferentes herramientas CASE fue que no existe un desarrollo lo suficientemente completo para cubrir todas las necesidades que un análisis de impacto requiere para tener éxito. Y cuando se encontró una herramienta que cuenta con herramientas adecuadas no está diseñada para soportar todo el proceso de desarrollo de software.

A pesar de que se han hecho esfuerzos por mejorar la interacción entre los analistas, la evolución de los modelos y la trazabilidad aún existe un gran trabajo por hacer, esto puede explicarse debido a que el interés en esta rama de la calidad del software es reciente y no se ha sido explorado en su totalidad.

También es importante tener en cuenta el costo de licenciamiento de las herramientas, ya que a pesar de ser reconocidas en el mercado y contar con un valor relativamente económico no tiene aplicativos que soporten el área en estudio; y la herramienta que cuenta con los módulos necesarios para cubrir trazabilidad y posibilitar un análisis de impacto de mayor efectividad tiene un costo mucho mayor que el de las otras herramientas.

5 ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL MÓDULO DENTRO DE AR2CA

5.1 ANÁLISIS DE LA FUNCIONALIDAD

En primera instancia para el desarrollo de este proyecto se hizo una recolección de requisitos con el fin de identificar las necesidades relacionadas con el manejo de trazabilidad. Los requisitos que se definieron son:

Nombre	Descripción
Req1	Generar una representación gráfica de las relaciones de trazabilidad
Req2	Definición de una iconografía clara
Req3	Establecer el tipo de requisitos
Req4	Estereotipar las relaciones
Req5	Definir relaciones validas y restringir las no validas
Req6	Caracterizar relaciones
Req7	Permitir el manejo de vistas
Req8	Generar una representación matricial
Req9	Aplicar zoom a las vistas
Req10	Generar de manera interactiva el grafo de trazabilidad

Tabla 1. Requisitos recolectados para la gestión de trazabilidad en AR2CA

Con base en las anteriores características que se plantearon para el desarrollo se definió crear dos formas para la gestión de la trazabilidad dentro de la herramienta CASE AR2CA, la construcción de un módulo de trazabilidad que genere una matriz de trazabilidad y la generación de un grafo de trazabilidad que permita navegar a través de las relaciones establecidas.

La matriz de trazabilidad agrupara los requisitos: Req5, Req6, Req7, Req8 dejando para el grafo de trazabilidad los requisitos: Req1, Req2, Req3, Req4, Req9 y Req10.

Teniendo en cuentas esto requisitos se hizo un análisis del módulo de trazabilidad que se desea construir, identificando la funcionalidad por medio de dos diagramas de casos de uso. Los casos de uso definidos en el primer diagrama, casos de uso de la matriz de trazabilidad, son los siguientes (Figura 47):

1. Generar Matriz de Trazabilidad: La principal funcionalidad del sistema es la construcción automática de una matriz de trazabilidad que se basa en una serie de parámetros, lo cuales definen los elementos y las relaciones que se desea ver en la matriz. Los parámetros puntualmente son: los tipos de elementos para el eje X, los tipos de elementos para el eje Y, y las relaciones entre los elementos encontrados de cada tipo para cada eje.
2. Gestionar Vistas de la Matriz: Las diferentes combinaciones de elementos y relaciones son llamadas Vistas, y en muchos casos se desea guardar estas Vistas con el fin de utilizarlas en otro momento. Por lo que el objetivo de este caso de uso es realizar el almacenamiento, ejecución y eliminación de las vistas generadas en la matriz de trazabilidad
3. Configurar Matriz de Trazabilidad: Tiene como objetivo definir los parámetros necesarios que definen como debe ser creada la matriz de trazabilidad. Cuando se desea construir la matriz de trazabilidad se debe seleccionar los elementos y las relaciones que se desea ver en la matriz, esta selección se hace de todos los diferentes tipos de elementos y relaciones que AR2CA soporta.
4. Gestionar Relaciones entre elementos: El objetivo de este caso de uso es permitir la gestión de las relaciones entre los elementos UML que sean mostrados en la matriz generada. Aquí se busca especificar cómo se leen

los datos de los elementos y las relaciones existentes tanto en un proyecto como en la herramienta CASE AR2CA del una de las principales estructuras de la herramienta, el catalogo.

Además de mostrar los casos de uso, este diagrama permite observar que el módulo se relaciona directamente con el catalogo de AR2CA para sus tareas principales.

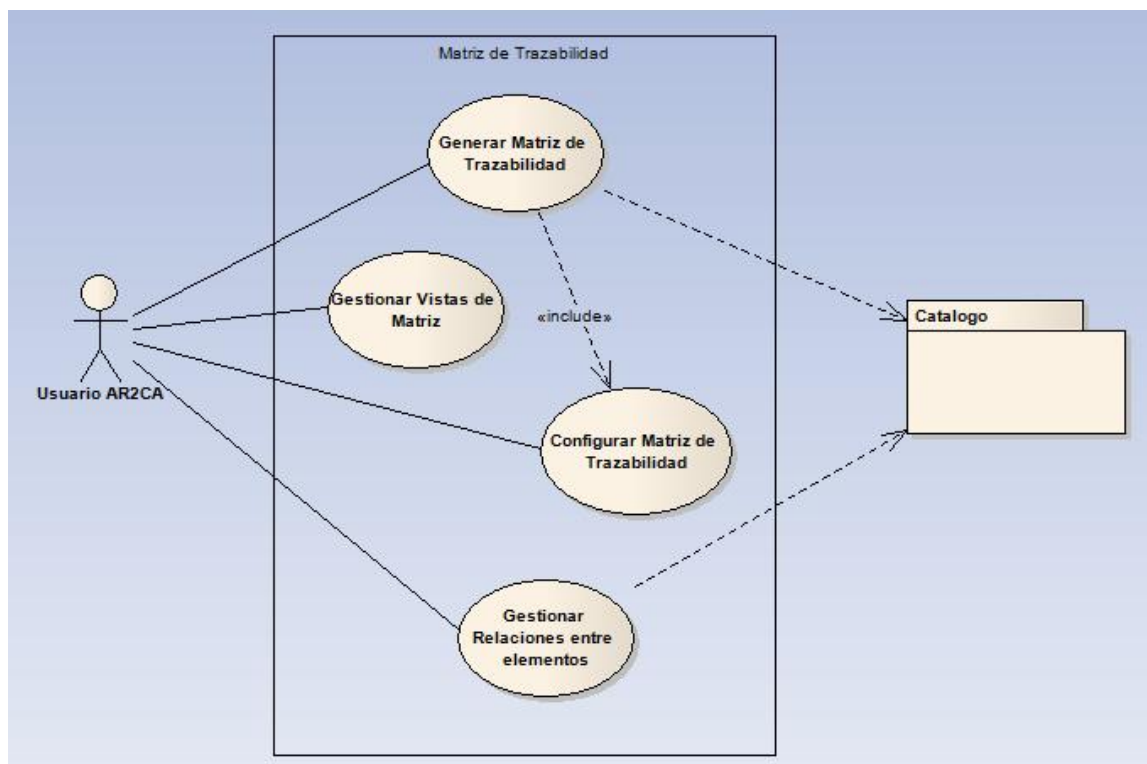


Figura 47. Casos de Uso relacionados con la Matriz de Trazabilidad

El segundo diagrama (Figura 48) muestra los casos de uso relacionados con el grafo de trazabilidad, estos son los siguientes:

1. Generar Grafo de Trazabilidad: El objetivo principal esta condensado en este caso de uso, construir de forma automática un grafo de trazabilidad

que muestre de forma optima las relaciones tanto horizontales como verticales existentes entre los elementos de un proyecto de modelado.

2. Controlar Interacción Grafo-Usuario: Este caso de uso agrupa todas las tareas necesarias para que el grafo se reconstruya a petición del usuario, se mueva en el espacio de trabajo y tenga una dinámica gráfica acorde con las necesidades de la herramienta y de los posibles análisis posteriores.
3. Construir Estructura de Datos: El objetivo aquí es organizar los datos de tal forma que sea mucho más sencillo de trabajarlos que en su estructura original. La estructura original es el catalogo de AR2CA, esta estructura es compleja y muchos de los datos ofrecidos no son relevantes por lo que es necesario crear una estructura mucho más sencilla que permita que el rendimiento del grafo sea de alto nivel.

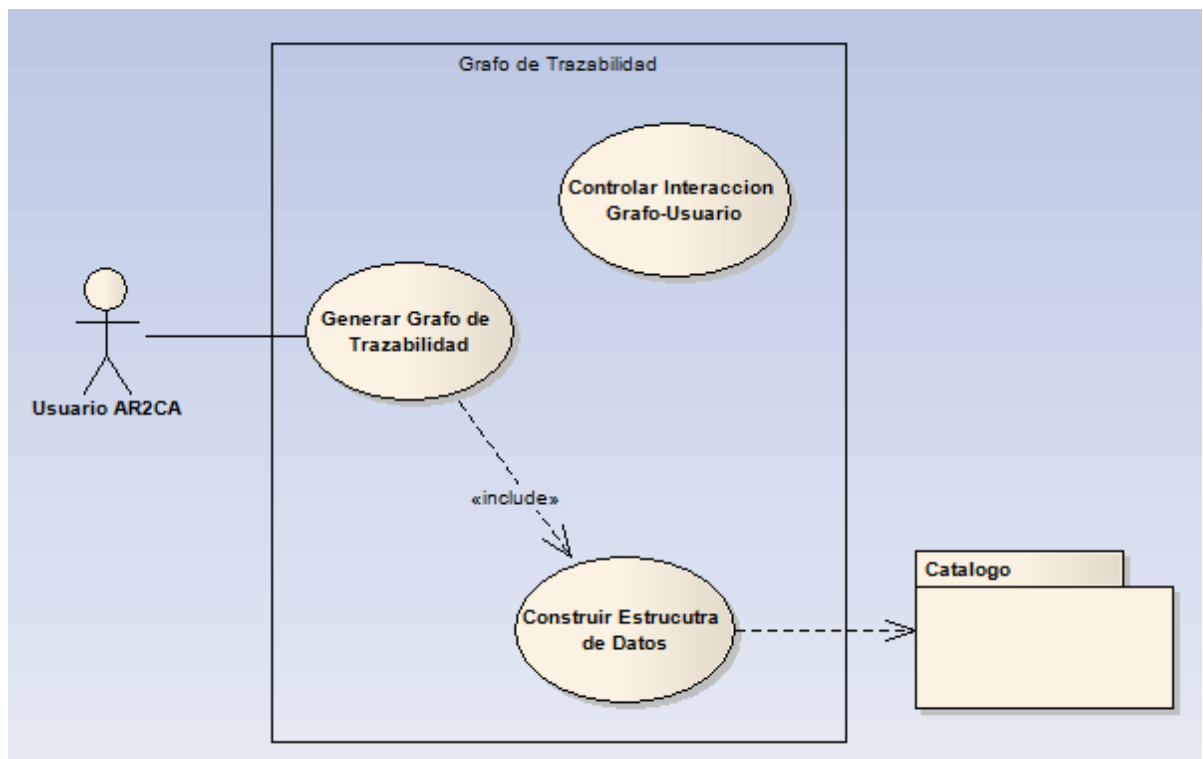


Figura 48. Casos de uso relacionados con el Grafo de Trazabilidad

5.2 DISEÑO DE LA INTERFACE GRÁFICA

Para el diseño de la matriz se tomo como base los ejemplos vistos en las diferentes herramientas comerciales existentes actualmente. Con base en estas herramientas se definió que la matriz debe permitir seleccionar los elementos a relacionar tanto para el eje X como para el Eje Y, y las relaciones existentes en los modelos, también se estableció que las relaciones se visualizaran en forma icónica en los puntos de corte de la matriz. También se tuvo en consideración crear una diferenciación entre las relaciones de horizontales y las relaciones verticales, para ello se definieron colores de fondo diferentes a cada una de estas relaciones, azul para las relaciones verticales y rojo a las relaciones horizontales.

Para el diseño del grafo de trazabilidad se toma como referencia el modelo propuesto por Benjamín Fry para el desarrollo de herramientas para la visualización de la información. Este modelo define varias etapas donde cada una desarrolla un aspecto importante dentro del desarrollo de este tipo de herramientas, a continuación se describe cual fue el trabajo realizado en cada etapa.

Adquisición y análisis de datos

AR2CA durante el proceso de creación de los modelos, que posteriormente serán el insumo para la creación tanto de la matriz como del grafo de trazabilidad, instancia la información de cada uno en dos partes. La primer estructura que AR2CA maneja para el almacenamiento de la información de los modelos es el Catalogo, el cual es una clase Treemap (Clase propia del lenguaje JAVA) la cual guarda toda la información de los elementos usados en los diferentes modelos en un nivel meta o, dicho de otra forma, de una manera abstracta. La segunda estructura que AR2CA usa para almacenar la información de los modelos es la clase *ControlDiagramas*, la cual pertenece al paquete

arca.controladorVista.controlvista, esta guarda en una estructura Treemap la información gráfica de los modelos hechos en el desarrollo de un proyecto. Cada modelo guarda la información de los elementos UML usados en su desarrollo.

En la Tabla 3 se explican los servicios utilizados para el correcto funcionamiento de los módulos de trazabilidad. Estas dos estructuras sirven como insumo de datos tanto para la matriz como para el grafo de trazabilidad.

Filtrado

En la etapa de filtrado la tarea del aplicativo es seleccionar el elemento origen, a partir de este elemento se hace el despliegue de los artefactos que son generados por el, o de los que son necesarios para la generación de éste.

Dentro del aplicativo el filtrado será hecho por el usuario de la herramienta. El usuario selecciona el elemento a ser base del análisis de los diferentes modelos que trabaja o en el browser de AR2CA, posterior a esto, al dar click derecho, se selecciona la opción “Grafo de Trazabilidad”.

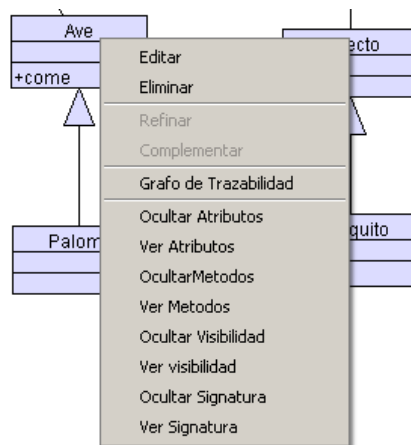


Figura 49. AR2CA - Imagen del Menú Contextual de Trazabilidad

Finalmente se mostrara el Browser de Trazabilidad donde el usuario podrá ver y crear la visualización de trazabilidad por medio de la interacción. (Figura 75)

Minar

La siguiente actividad propuesta es minar, esta tarea aplica cuando se maneja grandes cantidades de datos y procesos matemáticos y estadísticos. Los datos que generan los modelos UML dentro de AR2CA no tienen el tamaño necesario para aplicar un proceso matemático – estadístico de este tipo.

Representar

En la actividad de representación se define la forma o el tipo de visualización que se desea usar para mostrar los datos.

Se debe hacer especial énfasis en la siguiente pregunta para encontrar la representación que más se adecue con nuestro objetivo: ¿Cuál es la mejor forma para representar los datos que fueron entregados? Para responder esta pregunta es recomendable tener en cuenta tanto la calidad, la cantidad y el tipo de datos que se desea representar como una de las principales premisas de la visualización: usar la forma más simple de expresar las características más importantes de todo el conjunto de datos.

Para el módulo de trazabilidad se definió que la mejor forma de representar las estructuras almacenadas en AR2CA es un grafo.

Se selecciono el grafo como la mejor representación porque se quiere mostrar las relaciones existentes entre los elementos en forma de red. Mostrando las conexiones existentes entre un elemento y otro como flechas estereotipadas.

Inicialmente se definió tres modelos: grafo por niveles completo, grafo de trazabilidad completo y grafo de trazabilidad interactivo (Figuras 50, 51, 52).

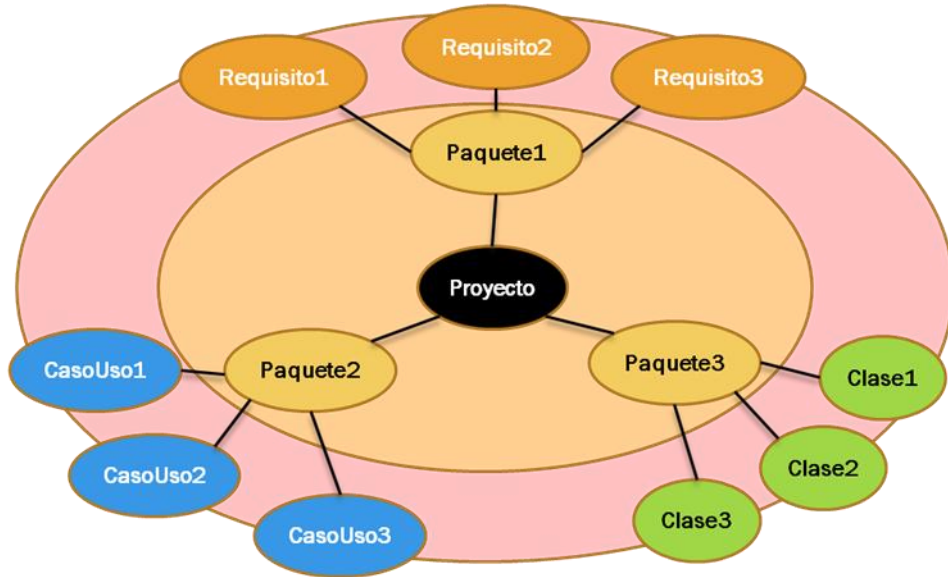


Figura 50. Grafo por niveles completo

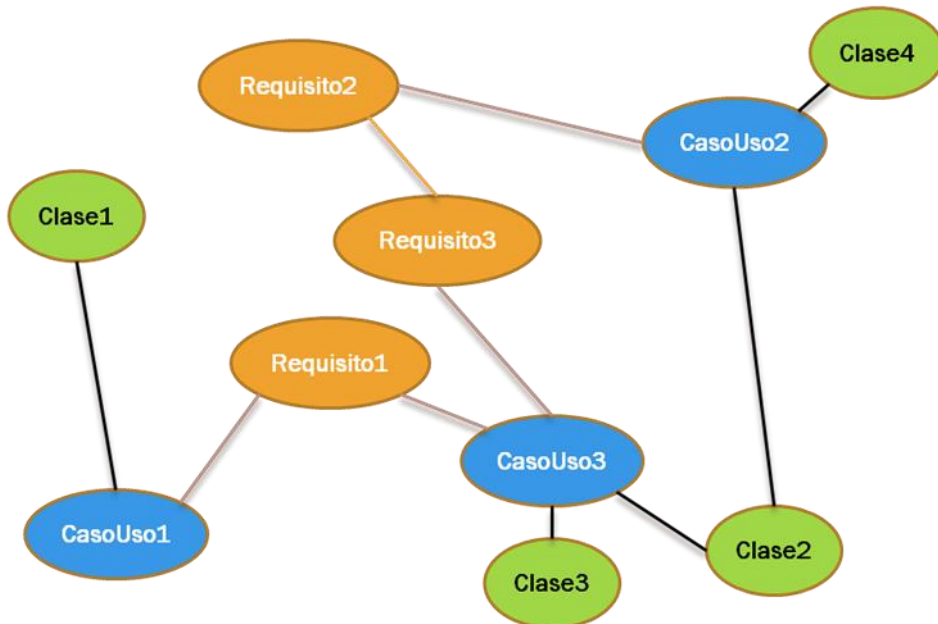


Figura 51. Grafo de Trazabilidad Completo

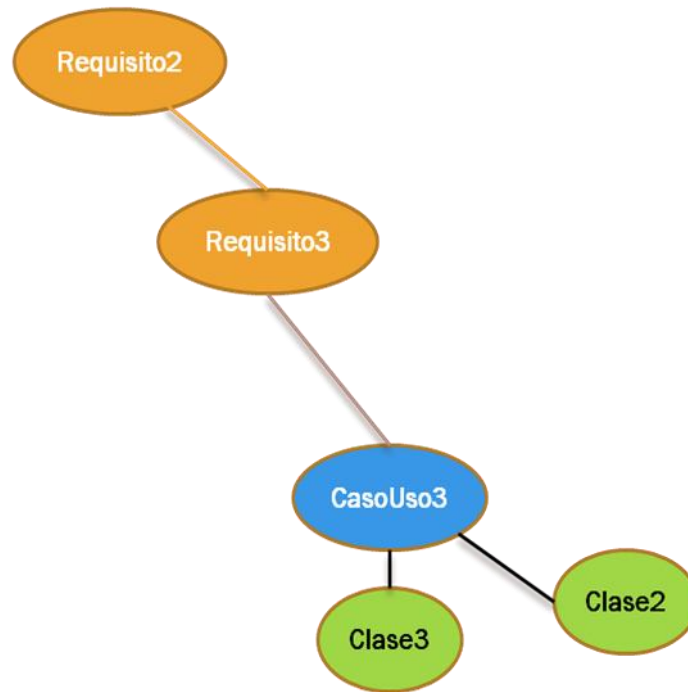


Figura 52. Grafo de Trazabilidad Interactivo

Dada la interacción que se quiere dar al aplicativo, la última alternativa fue seleccionada. La idea presentada hace referencia a la posibilidad de seleccionar que se desea ver desde un elemento inicial y con diferentes acciones mostrara los elementos hacia adelante, hacia atrás o las relaciones estructurales del mismo.

También, en el diseño de la representación se definió que los elementos serán trabajados como cuadros de texto divididos en dos partes. La primer parte contiene el tipo de elemento de modelado que se está viendo, y la segunda parte mostrará el nombre y datos relacionados con el elemento

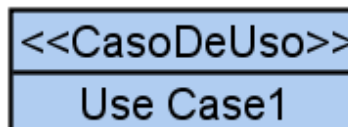


Figura 53. AR2CA – Elemento de modelado mostrado en la Visualización de Trazabilidad

Por su parte las relaciones se definieron como conectores dirigidos desde el elemento origen hacia el elemento destino con una etiqueta que muestra, a manera de estereotipo, cada tipo de relación existente dentro de la herramienta CASE AR2CA.

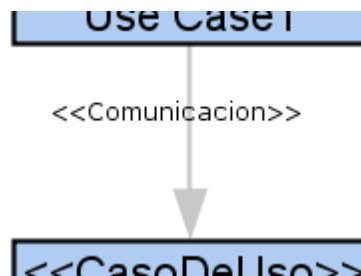


Figura 54. AR2CA – Relaciones de modelado mostrado en la Visualización de Trazabilidad

Refinación

En la etapa de refinación se estructura la forma adecuada para que los datos importantes o de mayor relevancia sean mostrados como tal, que sea resaltado por medio de características gráficas como contraste, color o forma.

Interacción

Finalmente, la interacción está compuesta por todas las acciones que permiten generar el grafo de trazabilidad y mejorar la visualización de los diferentes elementos involucrados

La interacción será por pasos, la idea es después de realizar el proceso de refinamiento, en el Browser de trazabilidad se mostrara el elemento base y los elementos que se hayan solicitado (antecedentes, sucesores, hermanos, o todos). El usuario podrá realizar de nuevo el proceso de refinamiento seleccionando en el browser un nuevo elemento sin perder la ruta de trazabilidad ya recorrida.

5.3 DISEÑO DETALLADO

5.3.1 Arquitectura de AR2CA

La arquitectura de AR2CA está fundamentada en tres patrones de arquitectura: el patrón layer, el patrón MVC y el patrón DAO. El patrón layer permite estructurar las clases en paquetes con un rol claro dependiendo de la capa en la cual está ubicado; el patrón MVC permite definir el esquema de notificación de cambios en el elemento de UML que se esté trabajando a la capa de presentación; el patrón DAO permite desacoplar la funcionalidad de los objetos del modelo de los servicios de persistencia. La figura 55 muestra una vista general de paquetes de la herramienta. En el área central se tiene la funcionalidad básica de manejos de modelos en sus capas de presentación, control, modelo y acceso a datos. En la franja de la izquierda se tienen los paquetes que contienen la funcionalidad para configurar la herramienta a través de factorías y las clases que representan la información meta de los elementos y los servicios relacionados con la administración de usuarios y dominios y la clase encargada de manejar los datos de la sesión de usuario. En la franja de la derecha se tienen los paquetes que agrupan servicios de utilidad específica tales como las utilidades de dibujo, el catálogo de elementos activos, los servicios de CVS, el editor de documentos XML, el editor de expresiones OCL y los servicios de transformación y generación de modelos. [22]

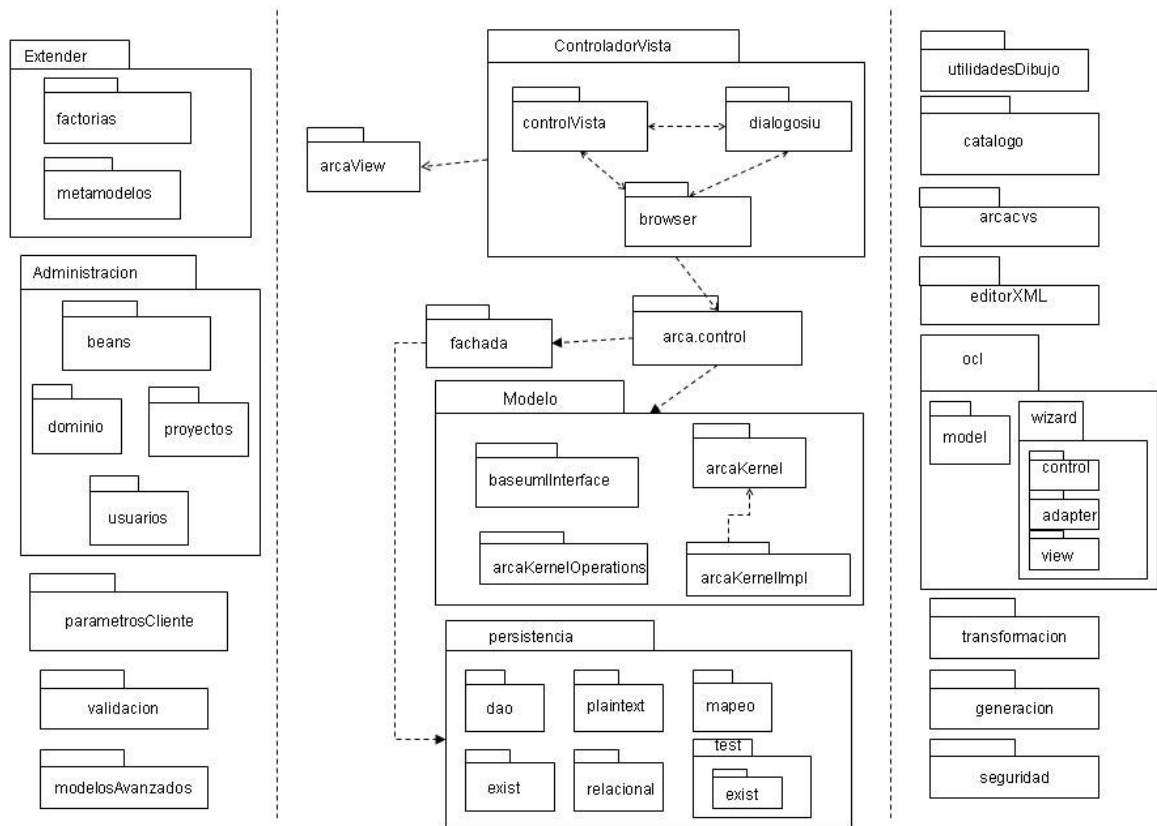


Figura 55. Diagrama de Paquetes de AR2CA [22]

A continuación se presenta una breve descripción de cada uno de los paquetes.

Paquete	Descripción
arcaView	Contiene las clases que construyen los elementos gráficos que pueden ser usados por los modelos UML disponibles en ARCA además de las clases que construyen las barras de herramientas para el manejo de estos elementos al interior de la aplicación. Rol: Vista
arcaKernel	Contiene la declaración de interfaces del modelo UML2.0. Rol: Modelo
arcaKernelImp	Contiene las implementaciones del modelo UML2.0.

	Rol: Modelo
arcaKernelOperation	<p>Contiene las clases que ofrecen servicios generales al interior de los modelos UML2.0.</p> <p>Rol: Modelo (servicios generales)</p>
baseUMLInterfaces	<p>Este paquete contiene declaración de las interfaces de los elementos del modelo que son conocidas por los objetos de vista y control. Sirven como interfaces adaptadoras para ocultar a los niveles superiores de la aplicación la complejidad de servicios de las implementaciones particulares de UML.</p> <p>Rol: Adaptadores entre el modelo y las capas de presentación y control</p>
Administración	Módulo que contiene los servicios de control de operaciones sobre los modelos y el manejo de proyectos y dominios
Browser	<p>Contiene las clases que contribuyen a la creación del Browser de ARCA desde donde se pueden visualizar los nodos que hacen alusión a un elemento determinado creado dentro de un modelo de ARCA así como su nivel de abstracción.</p> <p>Rol: Control de Vista.</p>
Catalogo	<p>Contiene las clases que almacenan en memoria los elementos que se encuentran activos en una sesión de usuario. Los elementos que suben a memoria son almacenados en una estructura indexada (java.util.TreeMap) que facilita la ubicación de los elementos por medio de su oid2.</p> <p>Rol: Servicios</p>
Control	Contiene las clases que concentran todos los servicios de

	la interfaz hacia el modelo. Rol: Control
controlVista	Contiene los elementos gráficos por medio de los cuales el usuario interactúa para ejecutar acciones sobre los modelos. Contiene la clase que representa el marco principal de la aplicación (ArcaFrame) y los elementos de dibujo que representan un modelo desde el punto de vista gráfico (DibujoCanvas, ConexionDibujo, ElementoDibujo). Rol: Control Vista
Arcacvs	Paquete que contiene los servicios subir (Check-in) y bajar (check-out) de elementos desde el servidor CVS
Persistencia	Contiene los servicios de persistencia de los diferentes elementos. Tiene dos paquetes anidados: plainText y relacional para la persistencia tanto en documentos XML para los modelos como en la base de datos para la información administrativa. Rol: Servicios de infraestructura (persistencia)
dialogosIU	Contiene los diálogos por medio de los cuales el usuario puede ingresar o editar elementos del modelo. Rol: Control de Vista
Dominio	Contiene la clase que describe el dominio que se encuentra activo. Rol: Modelo
editorXML	Clases que configuran un editor desde el cual puede generarse documentación para los modelos creados en ARCA a partir del lenguaje XML.
Extender	Contiene las clases que representan la información meta de los elementos para permitir su parametrización externa.
Fachada	Contiene los servicios que se realizan contra el

	almacenamiento persistente. Sirve para separar la lógica de acceso a datos del resto de clases del aplicativo.
manejoModelos	Contiene las clases que son responsables de controlar los modelos que se encuentran activos en la vista.
Modelo	Módulo que agrupa la estructura de modelos propia de UML. Está conformado por el paquete que describe las interfaces definidas en la especificación de UML; las implementaciones de dichas interfaces (arcaKernellImpl) y las interfaces que sirven como estructura adaptadora hacia las capas de presentación y control de la herramienta (baseumlInterfaces)
Ocl	Paquete que contiene los servicios de construcción del restricciones OCL (precondiciones, poscondiciones, invariantes y derivaciones) asociadas a los modelos del usuario
parametrosCliente	Contiene las clases encargadas de leer y actualizar las variables de ambiente en el entorno de un cliente.
Proyecto	Contiene la clase que hace un manejo de los proyectos definidos dentro de un dominio.
transformación	Paquete que contiene las clase para transformación a XML, tanto importar como exportar (GeneradorXML), la generación a plantillas de documentación en HTML (GeneradorDocumentacionHTML) y la clase que maneja las reglas de transformación de los elementos del modelo (UMLTransformer)
Usuario	Paquete creado con la finalidad de manejar la información asociada a los usuarios de la herramienta con sus respectivos privilegios y roles
utilidadesDibujo	Contiene las clases que ofrecen servicios generales de

	dibujo.
Seguridad	Provee servicios de seguridad tales como encriptación de cadenas de caracteres.

Tabla 2. Descripción de los paquetes que componen la arquitectura de AR2CA

5.3.2 Uso de servicios existentes

Después de haber revisado la arquitectura de la herramienta CASE AR2CA, y de realizar una breve descripción de cada uno de los paquetes se debe definir cuáles son los servicios existentes que serán de utilidad durante el desarrollo tanto de la matriz como del grafo de trazabilidad.

A continuación se presenta una descripción de las clases y los servicios que serán de utilidad durante el desarrollo de los módulos de trazabilidad.

Los servicios utilizados en la matriz de trazabilidad y en el grafo de trazabilidad se describen en la siguiente tabla:

CLASE	SERVICIO	FUNCIONALIDAD
Catalogo	darElemento (String)	Enviando como parámetro el OID de un elemento, se obtiene un objeto con su información
	darUnDibujo (String)	Enviando como parámetro el OID de un diagrama, se obtiene un objeto con su información
	existeRelacion (ElementoModelado,	Este servicio retorna un valor booleano definiendo si los

	ElementoMedelado, int)	elementos enviados como parámetros tiene o no una relación.
	darRelacionesEstructura (ElementoModelado, ElementoMedelado, int)	Este método retorna un vector con las relaciones de estructura existente entre los elementos enviados como parámetro.
ArcaControlSingleton	ingresarNuevaRelacion (DibujoCanvas, ElementoModelado, ElementoModelado, int)	Este método crea una nueva relación entre dos elementos en un diagrama determinado. Toda la información necesaria es enviada como parámetros.
	eliminarElemento (ElementoModelado)	Este método elimina un elemento del proyecto que se esté trabajando a nivel conceptual
DibujoCanvas	darElementosNodales()	Este servicio retorna un vector con todos los elementos de tipo nodal existentes en el proyecto activo.
	eliminarElemento (ElementoModelado)	Este servicio elimina un elemento del proyecto que se esté trabajando a nivel gráfico
ElementoModelado	getNombre()	Este servicio retorna una cadena de caracteres con el nombre que identifica al objeto instanciado
	getTipoElemento()	Este servicio retorna valor

		entero el cual identifica el tipo de elemento al cual pertenece un objeto instanciado
	getNombreOID()	Este servicio retorna una cadena de caracteres con el OID que identifica al objeto instanciado
ArcaModelosTabla	darNombreMeta (int)	Retorna una cadena de caracteres con el nombre de un determinado tipo de elemento identificado por su valor entero
	getRelacionesValidas()	Este método permite verificar que relaciones son validas para un determinado elemento
MetaElemento	getNombre()	Este servicio retorna una cadena de caracteres con el nombre que identifica al objeto instanciado
	tieneTrazabilidad (ElementoModelado)	Este método permite determinar si un elemento soporta trazabilidad

Tabla 3. Servicios utilizados en la matriz de trazabilidad existentes en AR2CA

En la siguiente parte se detallan todas las clases nuevas y los cambios realizados a las existentes para completar el correcto funcionamiento de los módulos de matriz de trazabilidad y grafo de trazabilidad.

5.3.3 Creación de Servicios

Después de realizar el diseño de las herramientas para la gestión de la trazabilidad dentro de AR2CA se pasa a su implementación como módulo dentro de esta. A continuación se describen los cambios realizados en la herramienta CASE para el desarrollo del módulo de trazabilidad.

Para que AR2CA tenga las herramientas necesarias para el soporte de trazabilidad tanto a nivel gráfico como conceptual, ha sido modificada su arquitectura adicionando un paquete de utilidad específica (Figura 56) ubicado justo con los otros paquetes de este tipo en la parte derecha del gráfico.

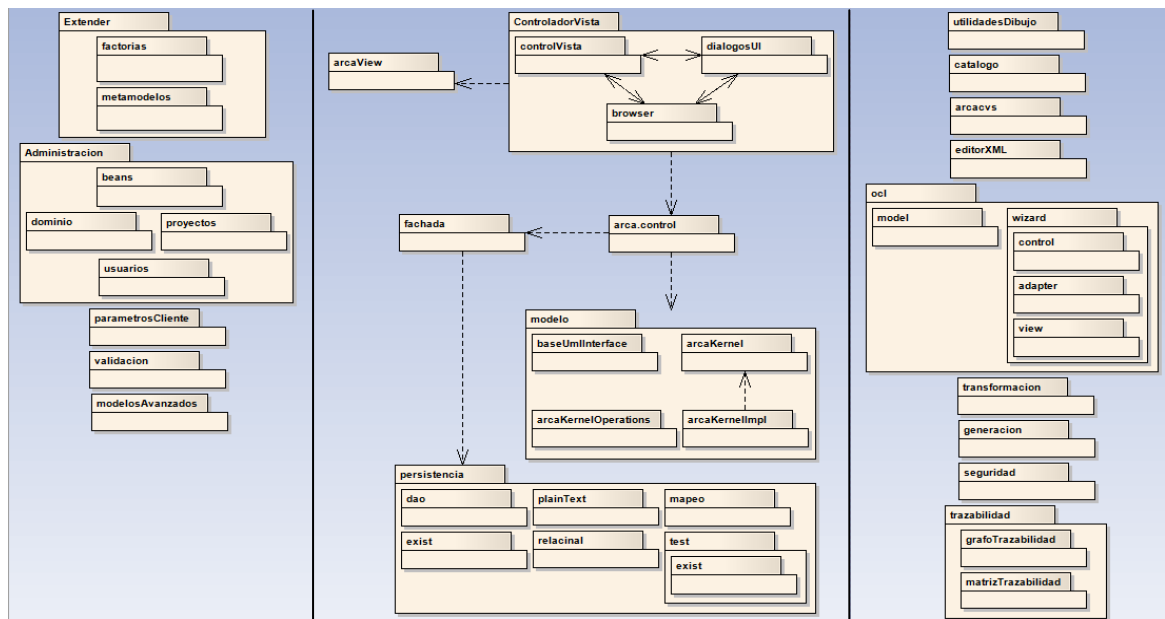


Figura 56. Arquitectura de AR2CA modificada con el módulo de trazabilidad

Algunas de las clases existentes fueron complementadas con la adición de servicios que permiten que tanto la matriz como el grafo de trazabilidad funcionen correctamente. En la siguiente tabla se listan las clases y los servicios creados con una pequeña descripción de su funcionalidad.

CLASE	SERVICIO	FUNCIONALIDAD
ArcaModelosTabla	obtenerTiposElementos()	Retornar un vector con los diferentes tipos de elementos soportados por AR2CA
	obtenerTiposRelaciones (boolean)	Retornar un vector con los diferentes tipos de relaciones soportadas por AR2CA
CatalogoArca	darOidDiagrama (ElementoModelado, ElementoModelado)	Retornar el cadena Identificadora de un diagrama que contenga los dos objetos de tipo ElementoModelado
	darDiagramas()	Retornar un vector donde se encuentran todos los diagramas existentes en un proyecto
	darRelacion (ElementoModelado, ElementoModelado)	Retornar una clase ElementoModelado que contenga la relación existente entre los dos objetos de tipo ElementoModelado
AreaDibujo	darDibujos()	Retornar un vector donde se encuentran todos los diagramas de forma gráfica existentes en un proyecto

Tabla 4. Servicios creados en las clases existentes en AR2CA

Adicionalmente para la representación gráfica de la trazabilidad por medio de un grafo se hace uso del toolkit de visualización *Prefuse*. *Prefuse* es un framework de gran amplitud que ayuda a los desarrolladores de software a crear aplicaciones

interactivas que permitan una visualización gráfica de la información haciendo uso del lenguaje de programación *JAVA*. *Prefuse* puede ser usado para construir aplicaciones standalone, componentes visuales embebidos en aplicación de gran escala, y en applets para Web. *Prefuse* intenta simplificar en gran medida el proceso de representación y ejecutar de manera eficiente el manejo de datos, mapeando los datos a una representación gráfica (a través de posición espacial, tamaño, forma, color, etc.), y la interacción con los datos. Algunas características de *Prefuse* son:

- Las estructuras de datos Tablas, Grafos y Arboles soportan atributos arbitrarios, indexación, y consultas, todo con un uso eficiente de memoria.
- Cuenta con componentes para el diseño, color, tamaño, forma, codificaciones, además de técnicas de distorsión y de animación.
- Una biblioteca con controles para la interacción y manipulación directa de los elementos gráficos.
- Animación soportada por medio de mecanismos de temporización de eventos.
- Consultas dinámicas sobre los datos filtrados.
- Un motor para la simulación de fuerzas físicas para diseño dinámico y animación.
- API simple y amigable para la creación, el proceso, la interacción, el desarrollo y el dibujado de los componentes gráficos.

Internamente el paquete trazabilidad tiene dos paquetes matrizTrazabilidad y grafoTrazabilidad. matrizTrazabilidad es el paquete donde se agrupan todas las clases creadas para la gestión de la matriz de trazabilidad. grafoTrazabilidad es el paquete donde se agrupan todas las clases creadas para la gestión del grafo de trazabilidad.

El paquete matrizTrazabilidad está compuesto por las clases DialogoSeleccionarElementosMatriz.java, DialogoSeleccionarRelacionesMatriz.java y MatrizTrazabilidad.java. DialogoSeleccionarElementosMatriz.java es una clase que hereda de javax.swing.JDialog, esta clase permite seleccionar los artefactos UML que se desea utilizar en la construcción de la matriz de trazabilidad tanto para el eje X como para el eje Y (Figura 57). DialogoSeleccionarRelacionesMatriz.java es una clase que hereda de javax.swing.JDialog, esta clase permite seleccionar las relaciones UML que se desea utilizar en la construcción de la matriz de trazabilidad (Figura 58). MatrizTrazabilidad.java es la clase principal del paquete ya que esta implementa la matriz de trazabilidad como tal, esta clase crea la estructura interna que permite luego construir y visualizar la matriz de trazabilidad del proyecto que se esté trabajando. La Figura 59 muestra la matriz de trazabilidad implementada en AR2CA.

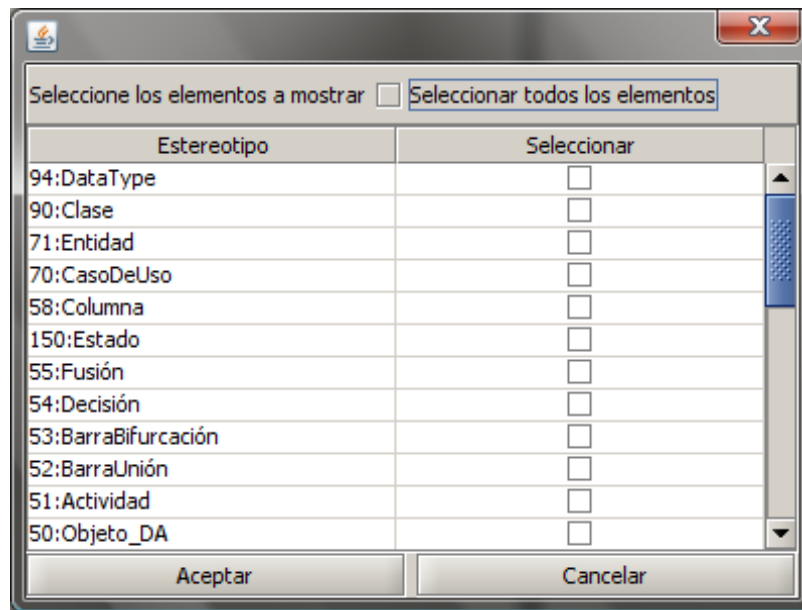


Figura 57. AR2CA – Dialogo de selección de Artefactos para la matriz de trazabilidad

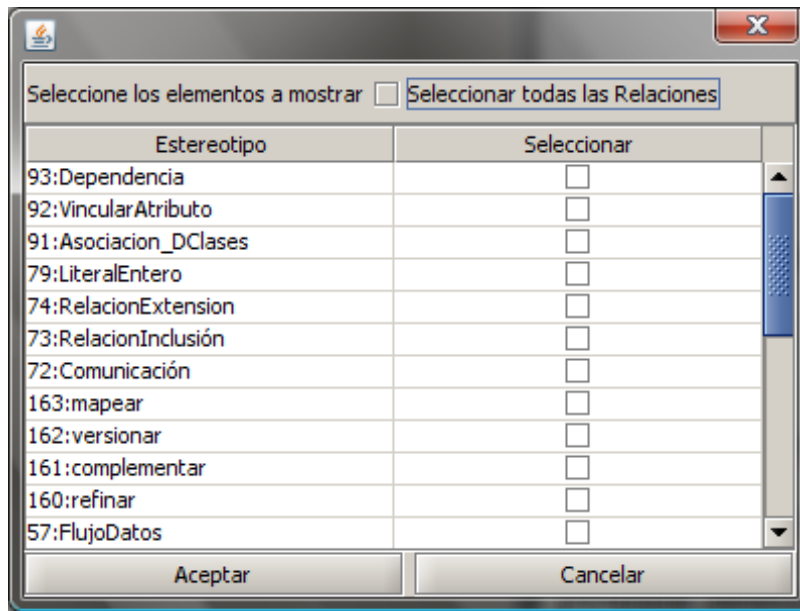


Figura 58. AR2CA – Dialogo de selección de Relaciones para la matriz de trazabilidad

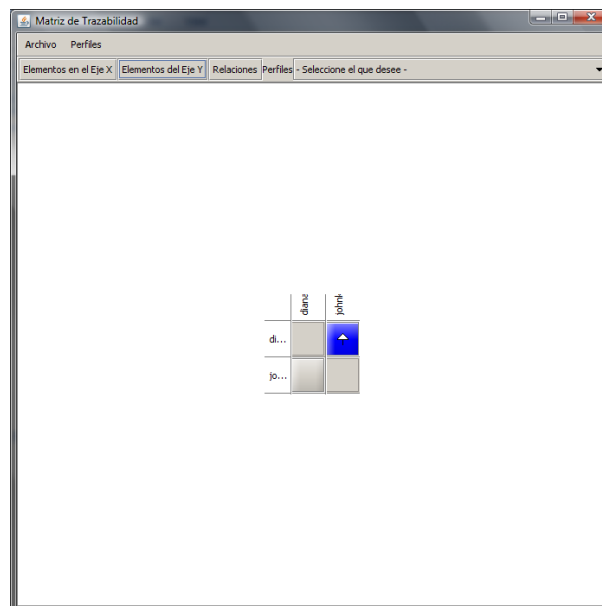


Figura 59. AR2CA – Matriz de Trazabilidad

El paquete grafoTrazabilidad está compuesto por la clase *GrafoTrazabilidad.java* la cual carga el toolkit de visualización *Prefuse* el cual fue modificado para que realice la carga de los datos necesarios para la construcción, a partir de los datos

existente en el catalogo donde AR2CA gestiona todos los elementos de los modelos UML del proyecto activo.

Puntualmente dentro *Prefuse* se modificaron las clases *GrafLib.java* y *DataView.java*. Originalmente *GrafLib.java* se encargaba de cargar los datos para la creación de un grafo o de un árbol a partir de un archivo XML donde se definían nodos, relaciones y la dirección de estas. Con la modificación realizada se consigue crear el método *cargarElementosCatalogo()* el cual lee y estructura los datos relacionados con los modelos existentes en AR2CA para su posterior uso por *DataView.java* y las clases relacionadas con esta. *DataView.java* se encarga de configurar la forma en que se dibujan tanto los nodos como los enlaces dentro de un árbol o un grafo, representación que nos concierne. En esta clase se definió la forma en que el grafo es visto, la forma en que los elementos de grafo interactúan entre sí, con el entorno y con el usuario, además de establecer la forma en que este es animado.

La Figura 60 Muestra el grafo de trazabilidad implementado en AR2CA.

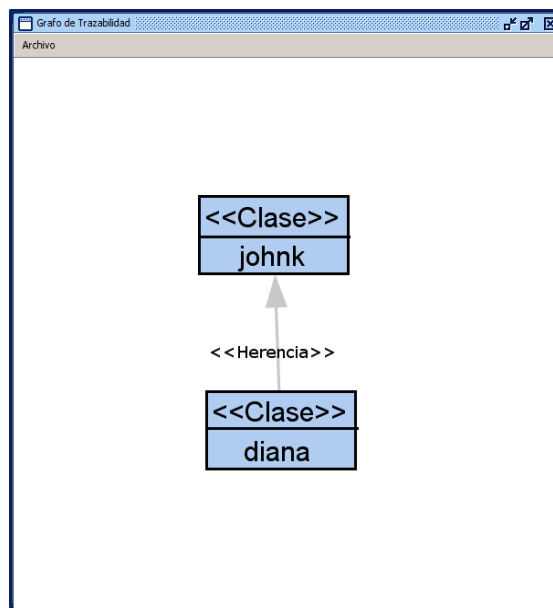


Figura 60. AR2CA – Grafo de Trazabilidad

5.4 TRAZABILIDAD EN AR2CA POSTERIOR A LA IMPLEMENTACIÓN

Dentro de la herramienta CASE AR2CA existen dos formas de gestionar las relaciones de trazabilidad y de estructura, matriz de trazabilidad y grafo de trazabilidad. La primera permite analizar las relaciones existentes entre diferentes elementos seleccionados por medio de una matriz donde se muestra iconográficamente la relación existente y por medio de colores el tipo al que esta pertenece. El grafo de trazabilidad muestra una gráfica representando los elementos y las relaciones entre ellos por medio de estereotipos, la construcción de este grafo se hace de forma interactiva con el usuario.

Dentro de las características de AR2CA encontramos la de generar relaciones de traza automáticamente, estas relaciones se crean por medio de un Wizard de mapeo. Para ejecutarlo se debe dar click derecho sobre el diagrama que se desee mapear, allí seleccionar el menú Mapeo y luego el diagrama al cual se desee mapear. (Figura 61). Al dar click sobre el diagrama automáticamente se ejecuta el Wizard (Figura 62) que permite mapear los elementos del diagrama hacia elementos nuevos o existentes en un diagrama destino.

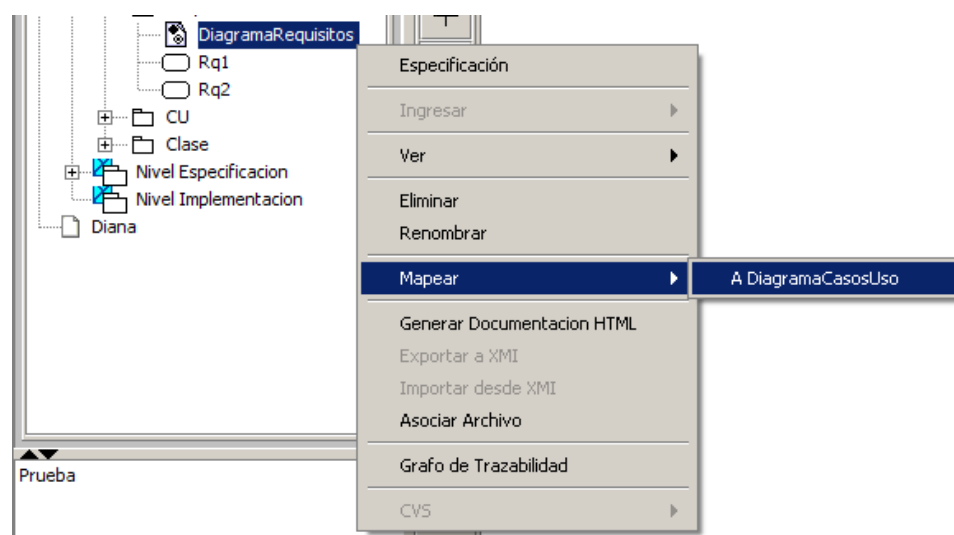


Figura 61. AR2CA – Menú Mapeo para un diagrama de requisitos

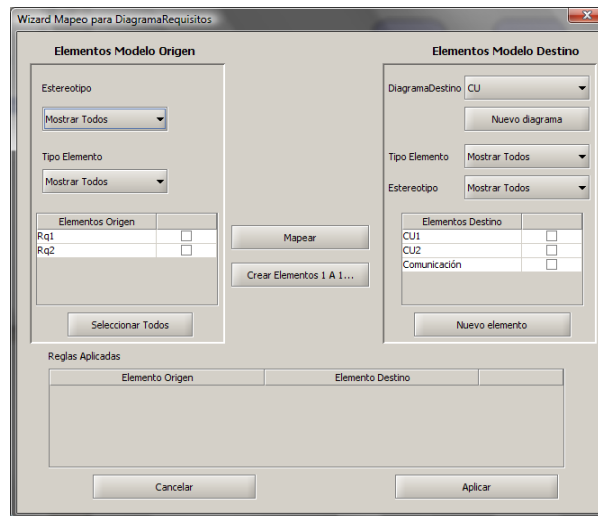


Figura 62. AR2CA – Wizard Mapeo para un diagrama de requisitos hacia un diagrama de Casos de uso

Aquí sencillamente seleccionamos el elemento origen (parte izquierda) que deseamos mapear hacia el elemento destino (parte derecha) que corresponda. Posterior a la selección se debe presionar el botón Mapear con lo que queda establecida la relación de mapeo. Todas las relaciones creadas en este Wizard se muestran en la tabla Reglas Aplicadas. (Figura 63). Finalmente se presiona el botón aplicar para que estas relaciones queden establecidas, el sistema muestra un mensaje de confirmación diciendo el numero de reglas aplicadas (Figura 64).

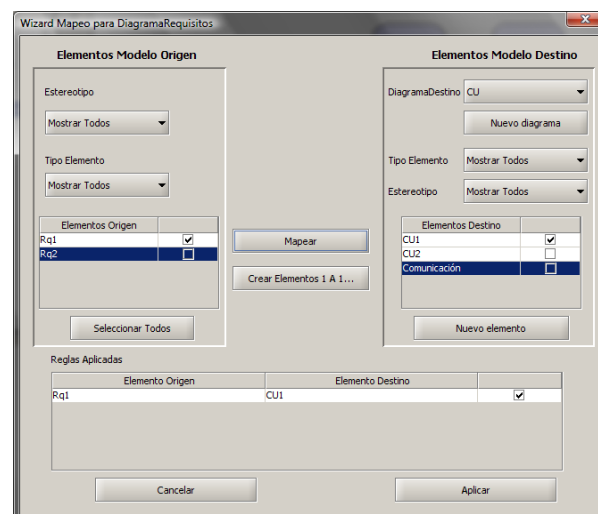


Figura 63. AR2CA – Wizard Mapeo para un diagrama de requisitos hacia un diagrama de Casos de uso configurado

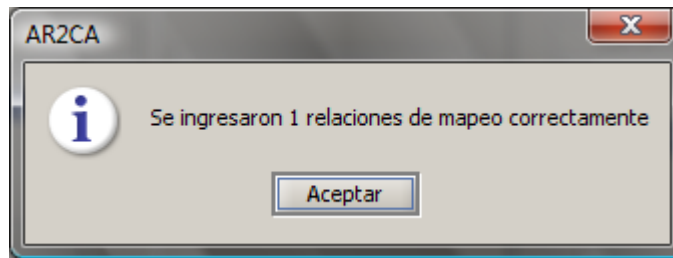


Figura 64. AR2CA – Wizard Mapeo – Mensaje de confirmación de aplicación de reglas

AR2CA posee una matriz de trazabilidad, herramienta que permite visualizar las relaciones existentes en un proyecto activo en forma de tabla. Para hacer uso de esta herramienta se debe ir al menú Herramientas y allí seleccionar la opción Trazabilidad (Figura 65). Al dar click aparece la matriz de trazabilidad, la cual debe ser configurada antes de mostrar la tabla con los elementos. (Figura 66).

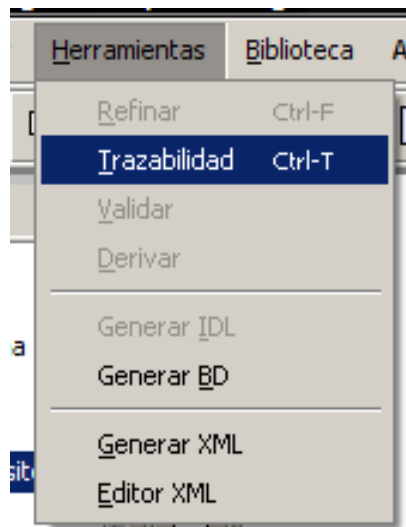


Figura 65. AR2CA – Menú Matriz de Trazabilidad

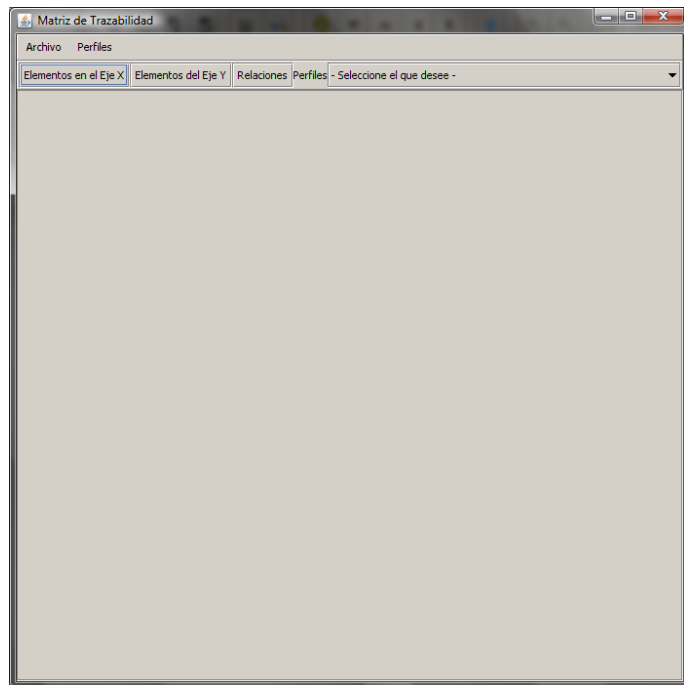


Figura 66. AR2CA – Matriz de Trazabilidad

Para configurar la matriz de trazabilidad se tiene la posibilidad de usar perfiles preestablecidos los cuales configuran automáticamente la herramienta. Estos se seleccionan en el menú desplegable Perfiles con solo hacer click en el que se desee (Figura 67). Con esto la matriz inmediatamente se construye y muestra los elementos seleccionados y las relaciones entre estos (Figura 68).

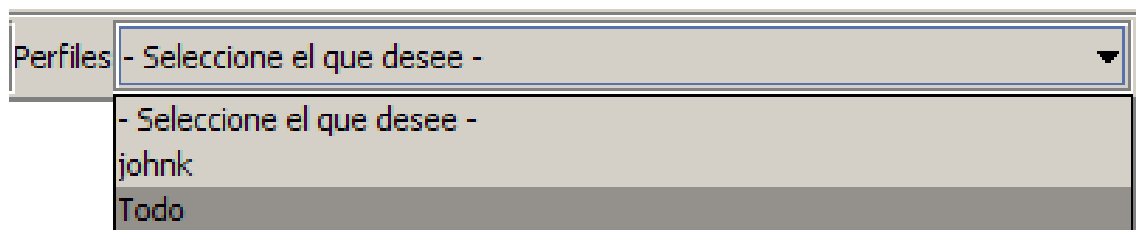


Figura 67. AR2CA – Menú Desplegable Perfiles

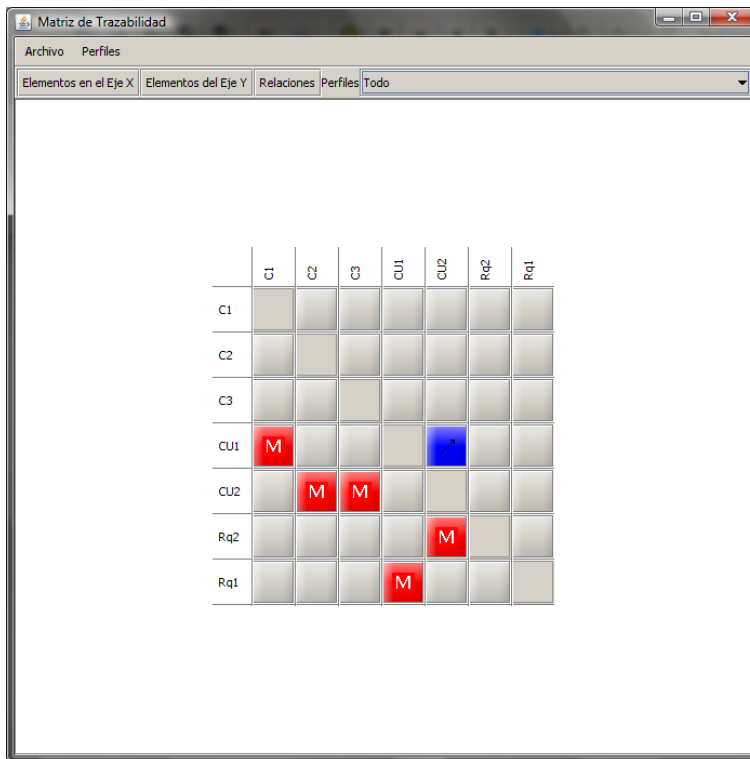


Figura 68. AR2CA – Matriz de Trazabilidad Llena

También se puede configurar un perfil de forma manual, esto con los botones ubicados en la parte superior izquierda (Figura 69) los cuales controlan los tipos de elementos que se ubicaran en el eje X, los tipos de elementos que se ubicaran en el eje Y, y los tipos de relaciones que se mostraran en la matriz. Cada uno despliega una ventana donde se selecciona elementos (Figura 70) o relaciones (Figura 71) según el caso.

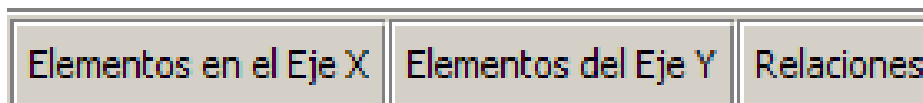


Figura 69. AR2CA – Botones de configuración de la matriz

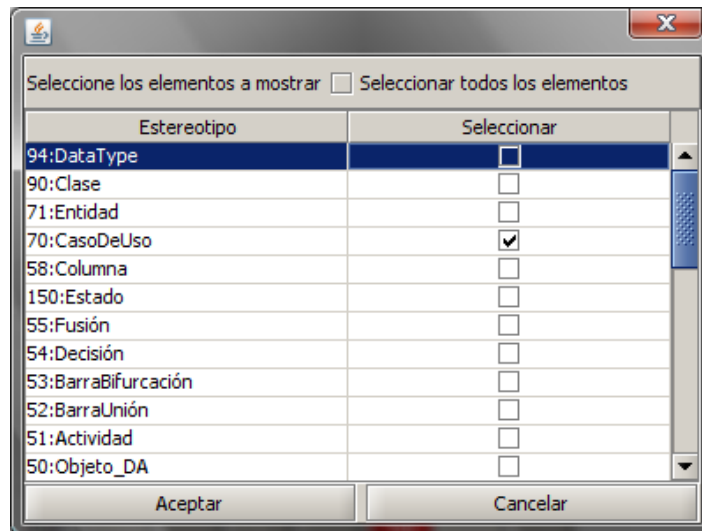


Figura 70. AR2CA – Pantalla de selección de Elementos

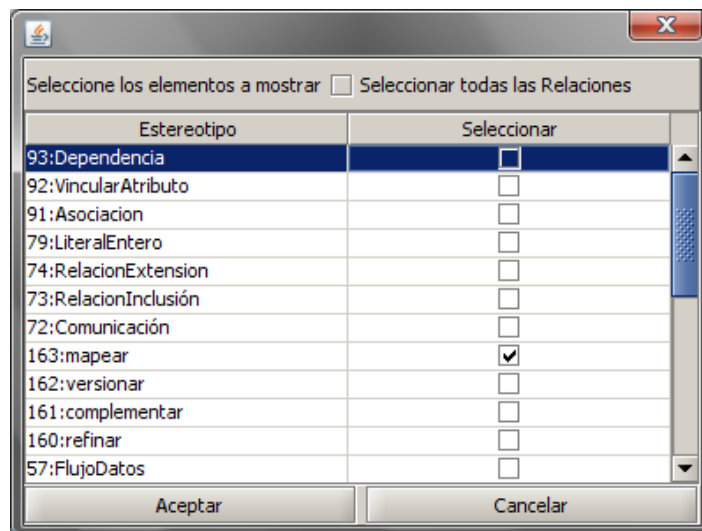


Figura 71. AR2CA – Pantalla de selección de Relaciones

Las ventanas de selección tienen la misma estructura. En la parte central una tabla con todos los tipos de elementos, en esta tabla se puede seleccionar uno a uno los tipos de elementos que se deseen usar. Los botones Aceptar el cual se presiona cuando se termine la selección o el botón Cancelar en caso de querer cerrar la ventana sin generar cambios. Finalmente en la parte superior una caja de chequeo que permite seleccionar todos los elementos de la tabla. (Figuras No. 70

y 71). En el momento de aceptar los cambios en la configuración la matriz se actualiza automáticamente presentando los nuevos resultados. (Figura 68).

En caso de querer convertir una configuración en perfil, sencillamente se debe ir al menú Perfiles y allí seleccionar la opción Adicionar Perfil – Selección Actual y este se guardara para posterior uso.

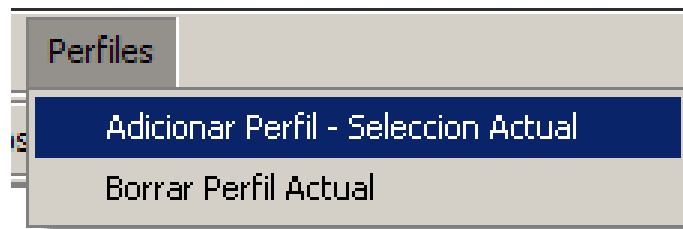


Figura 72. AR2CA – Menú Adicionar Perfil

Además de la matriz de trazabilidad, AR2CA tiene otra herramienta que es el grafo de trazabilidad. Para hacer uso del grafo se debe dar click derecho sobre el icono del elemento en el Browser (Figura 73) o sobre la figura en los diagramas de modelado (Figura 74) y dar click sobre la opción *Grafo de Trazabilidad*, al hacer esto se ejecuta la ventana del grafo de trazabilidad donde se muestra el elemento seleccionado como principal, los elementos con los cuales se relaciona directamente y las relaciones estereotipadas que tiene con estos elementos (Figura 75).

Para hacer que el grafo crezca se debe retroalimentar el grafo dando click en el elemento que se desea ampliar, con esto el grafo carga los datos relacionados con el nuevo elemento y adiciona estos a los existentes y crea el nuevo gráfico (Figura 76).

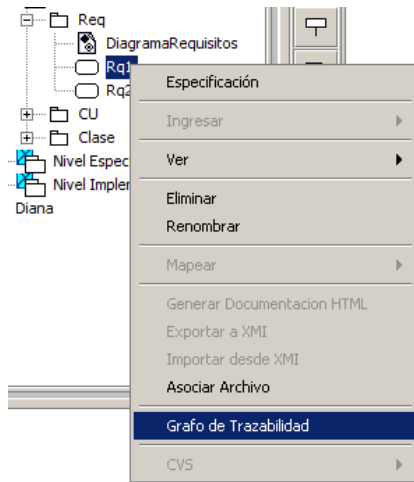


Figura 73. AR2CA – Menú Grafo de Trazabilidad Browser

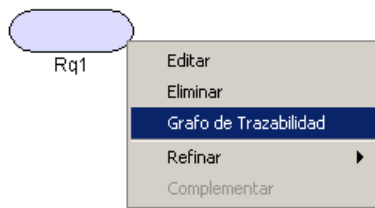


Figura 74. AR2CA – Menú Grafo de Trazabilidad Área de trabajo

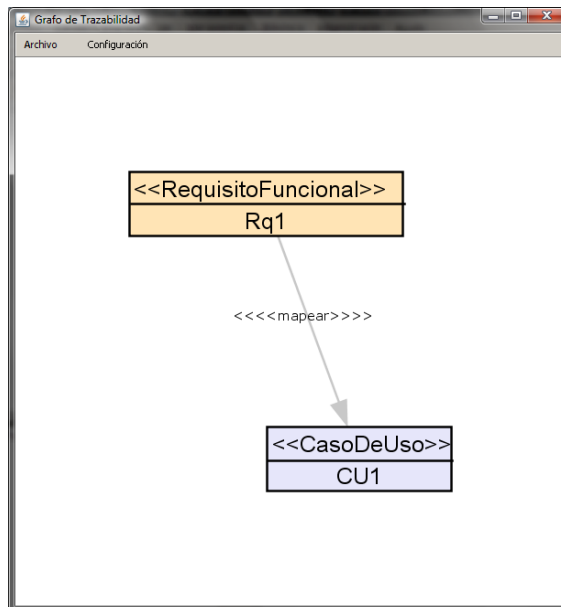


Figura 75. AR2CA – Menú Grafo de Trazabilidad Área de trabajo

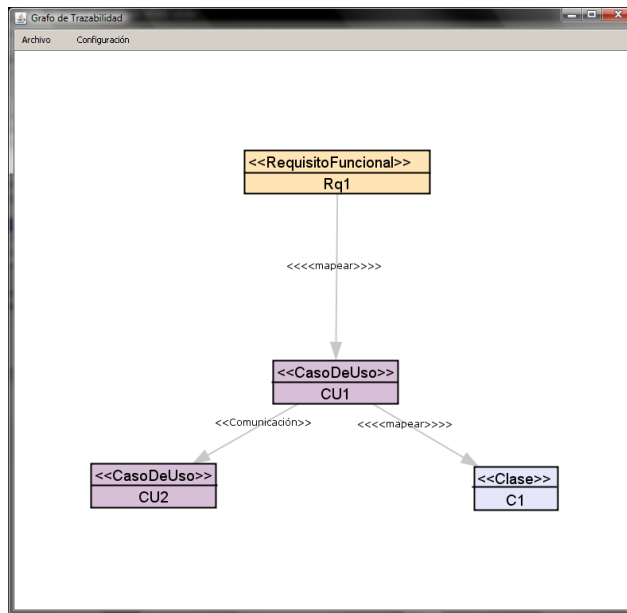


Figura 76. AR2CA – Menú Grafo de Trazabilidad Área de trabajo

6 CONCLUSIONES

Para llevar a cabo esta investigación se comienza con un estudio de los conceptos relacionados con trazabilidad y análisis de impacto y posterior a esto se estudia como las aplicaciones comerciales actuales gestionan la trazabilidad en sus proyectos de modelado. De aquí se capturan algunas características importantes que pueden aportar al desarrollo del módulo propuesto y se ven fallas que tiene algunas herramientas que de igual forma fortalecen este proyecto.

Dado que el desarrollo tiene un fuerte componente gráfico, se hace una investigación relacionada con visualización de la información, esto con el fin de definir la mejor representación gráfica y las herramientas que permitan que el desarrollo de este componente sea mucho más ágil y su uso más eficiente. Aquí se encuentran dos elementos muy fuertes dentro del desarrollo de esta investigación: El primero el modelo para el desarrollo de herramientas de visualización propuesto por Benjamín Fry en su tesis doctoral que se adapta a las necesidades y se aplica en el desarrollo del grafo de trazabilidad. El segundo es el Framework *Prefuse* con el cual se desarrolla el grafo de trazabilidad de forma mucho más rápida y que permite realizar una gestión de este mucho mejor.

Finalmente, se estudia la herramienta CASE AR2CA con el fin de definir el cómo se puede hacer la implementación tanto de la matriz de trazabilidad como del grafo de trazabilidad. Para terminar el trabajo con su implementación y prueba.

Los principales aportes del desarrollo de esta investigación fueron los siguientes:

- El uso de entornos gráficos para la representación de datos permite capturar información de una forma mucho más fácil, permitiendo generar conclusiones o toma de decisiones de la una forma más acertada.

- El uso de frameworks gráficos para el desarrollo de aplicaciones que así lo requieran, facilita el trabajo y hacen que se optimice el tiempo empleado para la finalización del proyecto.
- El análisis de datos no necesariamente se debe hacer sobre una gran cantidad de datos, si se consiguen datos completos y de alta calidad
- La estructura de datos que maneja AR2CA para el almacenamiento de los datos de los modelos desarrollados es idónea para generar modelos de trazabilidad que permitan mejorar el desarrollo de proyectos software utilizando la herramienta CASE en cuestión.
- La definición de una arquitectura de software clara y organizada permite extender cualquier herramienta de forma fácil y rápida permitiendo que esta evolucione con las necesidades de su sector de aplicación.

Para terminar puedo decir que la experiencia adquirida durante el desarrollo de esta investigación fue muy enriquecedora, gratificante y sobretodo significativa, porque abrió mi mente a nuevas formas de pensamiento, y me hizo reconocer que es necesario tener una visión de mundo que permita afrontar los diferentes problemas que se presentan durante el desarrollo laboran en las diferentes áreas de la ingeniería de sistemas

Por otro lado el pertenecer a un grupo tan emprendedor y exigente, como el grupo de investigación de ingeniería de software de EAFIT, me ayudó a crecer a nivel personal y profesional abriéndome oportunidades dentro de la industria del software, generando nuevas perspectivas, y cambiando mi manera de percibir las cosas involucrando la posibilidad de probar nuevas maneras de aprender, compartir y actuar.

7 RECOMENDACIONES

Como posibles recomendaciones o trabajos futuros se puede tener en cuenta

- Generar una especificación gráfica orientada a la descripción de la evolución de los elementos de modelado.
- Implementar otras herramientas de visualización que permitan mejorar la interacción entre el usuario y la herramienta, como por ejemplo ojo de pescado o zoom.
- Adaptar o implementar de forma completa de la trazabilidad en aspectos dentro de la herramienta case AR2CA.
- Generar una herramienta de trazabilidad independiente a la herramienta CASE, donde se apliquen los conceptos de visualización de la información a cualquier proyecto de modelado.

8 REFERENCIAS BIBLIOGRÁFICAS

[1] Tabares M. Análisis de la trazabilidad desde la perspectiva de la orientación a aspectos, Revista EIA, ISSN 1794-1237 Número 4 p. 95-102. Noviembre 2005, Escuela de Ingeniería de Antioquia, Medellín (Colombia)

[2]www.unisys.com/eprise/main/admin/corporate/doc/clarity/Services/Traceability_and_Impact_Analysis_WP.pdf

[3]www.wagse.informatik.uni-kl.de/teaching/re/ss2006/downloads/traceability.pdf

[4] Icontec. Sistema de gestión de la calidad: Fundamentos y vocabulario. Norma Técnica Colombiana NTC-ISO 9000. 2000-12-15.

[5] http://www.jiludwig.com/Requirements_Management_Tools.html

[6] <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos>

[7] <http://www.glosariográfico.com>

[8]<http://www.programaempresa.com/empresa/empresa.nsf/paginas/776B9BE6B7A57A48C125702900423FB6?OpenDocument>

[9] <http://msdn.microsoft.com/es-es/library/ms159640.aspx>

[10] <http://www.infovis.net/printMag.php?num=157&lang=1>

[11]

<http://www.infovis.net/printFicha.php?rec=revista&num=51&lang=1&palabra=arbol>

[12] http://es.wikipedia.org/wiki/Diagrama_de_caja

[13] http://en.wikipedia.org/wiki/Heat_map

[14] [http://es.wikipedia.org/wiki/%C3%81rbol_\(estructura_de_datos\)](http://es.wikipedia.org/wiki/%C3%81rbol_(estructura_de_datos))

[15] <http://es.wikipedia.org/wiki/Dendrograma>

[16] <http://escrituravisual.blogspot.com/2008/05/7-estadigrafa.html>

[17] <http://www.willydev.net/crystaldesde0/html/crconcharttypes.htm>

[18]

<http://www.infovis.net/printFicha.php?rec=revista&num=51&lang=1&palabra=arbol>

[19] <http://jheer.org/vizster/>

[20] <http://www-2.dc.uba.ar/congresos/cacic2002/Documents/PgmVisInfo.pdf>

[21] Lengler, R. Eppler, M. Towards a Periodic Table of Visualization Methods for Management, Institute of Corporate Communication, University of Lugano, Switzerland

[22] Blanco, G. Vásquez, J. Anaya, R. Manual de Usuario de AR2CA 3.0, Manual de Referencia, Version 0.1. Universidad EAFIT. Grupo de investigación de Ingeniería de Software. 2007

[23] <http://www.borland.com/us/products/caliber/rm.html>

[24] http://support.objecteering.com/objecteering6.1/help/us/objecteering_uml_modeler/intro/presentation.htm

[25] Pfleeger, S. Bohner, S. "A Framework for Software Maintenance Metrics," IEEE Transactions on Software Engineering, May 1990.

[26] Turver, R. Munro M. "An Early Impact Analysis Technique for Software Maintenance," Journal of Software Maintenance: Research and Practice, Volume 6, No. 1, January-February 1994.

[27] Arnold, R. S. Bohner, S. "Impact Analysis - Towards A Framework for Comparison," Proceedings of the Conference on Software Maintenance, Los Alamitos, CA, September 1993.

[28] Lee, M. "Change Impact Analysis of Object-Oriented Software". Doctoral Thesis. George Mason University. Fairfax, Virginia, 1998.

[29] Fry, B. "Computational Information Design". Doctoral Thesis. Massachusetts Institute of Technology. Cambridge, Massachusetts. 2004.

[30] Tukey, J. "Exploratory Data Analysis". Addison-Wesley. 1977.

[31] <http://mobbingopinion.bpweb.net/artman/uploads/tabla-1.jpg>

[32] http://www.ub.es/aplica_infor/spss/imagenes/bidi26.gif

[33] <http://www.fisterra.com/mbe/investiga/gráficos/images/Image125.gif>

[34] <http://www.nosolousabilidad.com/articulos/img/eye04.jpg>

[35] http://es.wikipedia.org/wiki/Diagrama_sagital

[36] http://www.hrc.es/bioest/Ejemplos_histo.html

[37] <http://www.kovcomp.co.uk/support/XL-Tut/demo-pcor.html>

[38] <http://www.findability.org/images/uxradar.jpg>

[39] <http://www.reportportal.com/help/images/TreeMap/TreeMap.gif>