

Construcción de una Herramienta para Visualización de Bases de Conocimiento GeneXus

Taller V - Marzo 2001

Integrantes:

Ana Rozza

Marcos Crispino

Tutores:

Enrique Almeida

Gustavo Crispino

**Facultad de Ingeniería
Universidad de la República**

1 Índice de Documentos

<u>INFORME FINAL</u>	<u>A</u>
<u>ANÁLISIS DE REQUERIMIENTOS</u>	<u>B</u>
<u>DISEÑO</u>	<u>C</u>
<u>INTERFACES ENTRE MÓDULOS</u>	<u>D</u>
<u>CÓMO IMPLEMENTAR UN ALGORITMO DE OPTIMIZACIÓN</u>	<u>E</u>
<u>MANUAL DE USUARIO</u>	<u>F</u>
<u>APÉNDICE A - REFERENCIAS</u>	<u>G</u>
<u>APÉNDICE B - CLASE OptimizadorPrueba</u>	<u>H</u>

Informe Final

Índice

<u>1</u>	<u>INTRODUCCIÓN</u>	5
<u>1.1</u>	<u>OBJETIVO</u>	5
<u>1.2</u>	<u>ALCANCE</u>	5
<u>1.3</u>	<u>DEFINICIONES Y ABREVIACIONES</u>	5
<u>1.3.1</u>	<u>DEFINICIONES</u>	5
<u>1.3.2</u>	<u>SIGLAS</u>	5
<u>1.4</u>	<u>ORGANIZACIÓN</u>	5
<u>2</u>	<u>DESCRIPCIÓN GENERAL</u>	7
<u>2.1</u>	<u>ORGANIZACIÓN DE LA DOCUMENTACIÓN</u>	7
<u>2.2</u>	<u>CÓMO SE DEBE LEER ESTE INFORME</u>	7
<u>3</u>	<u>PRESENTACIÓN DEL PROYECTO</u>	9
<u>3.1</u>	<u>INTRODUCCIÓN</u>	9
<u>3.2</u>	<u>PRESENTACIÓN DEL PROBLEMA</u>	9
<u>3.3</u>	<u>INTEGRANTES</u>	9
<u>3.4</u>	<u>OBJETIVO DEL PROYECTO</u>	10
<u>4</u>	<u>DESARROLLO DEL PROYECTO</u>	11
<u>4.1</u>	<u>ORGANIZACIÓN</u>	11
<u>4.2</u>	<u>ETAPAS</u>	11
<u>4.3</u>	<u>CRONOGRAMA</u>	12
<u>5</u>	<u>ETAPA DE INVESTIGACIÓN</u>	14
<u>5.1</u>	<u>OBJETIVOS</u>	14
<u>5.2</u>	<u>CRONOGRAMA</u>	14
<u>5.3</u>	<u>DESARROLLO</u>	14
<u>5.3.1</u>	<u>ESTUDIO DEL LENGUAJE DE PROGRAMACIÓN JAVA™</u>	14
<u>5.3.2</u>	<u>ESTUDIO DE GXPUBLIC™</u>	15
<u>5.3.3</u>	<u>ESTUDIO DE XML</u>	15
<u>5.3.4</u>	<u>ESTUDIO DE ALGORITMOS DE OPTIMIZACIÓN</u>	15
<u>5.4</u>	<u>RESULTADOS Y EVALUACIÓN</u>	17
<u>6</u>	<u>ETAPA DE DEFINICIÓN DE REQUERIMIENTOS</u>	18
<u>6.1</u>	<u>OBJETIVOS</u>	18
<u>6.2</u>	<u>CRONOGRAMA</u>	18
<u>6.3</u>	<u>DESARROLLO</u>	18
<u>6.4</u>	<u>RESULTADOS Y EVALUACIÓN</u>	18

<u>7</u>	<u>ETAPA DE IMPLEMENTACIÓN</u>	19
<u>7.1</u>	<u>OBJETIVOS</u>	19
<u>7.2</u>	<u>CRONOGRAMA</u>	19
<u>7.3</u>	<u>DESARROLLO</u>	20
<u>7.3.1</u>	<u>DISEÑO</u>	20
<u>7.3.2</u>	<u>DATOS BÁSICOS</u>	21
<u>7.3.3</u>	<u>INTERFAZ DE COMUNICACIÓN ENTRE MÓDULOS</u>	21
<u>7.3.4</u>	<u>LECTOR DE KBS</u>	21
<u>7.3.5</u>	<u>OPTIMIZADOR</u>	21
<u>7.3.6</u>	<u>VISUALIZADOR</u>	22
<u>7.3.7</u>	<u>INTEGRACIÓN DE MÓDULOS</u>	22
<u>7.4</u>	<u>RESULTADOS Y EVALUACIÓN</u>	22
<u>8</u>	<u>ETAPA DE DOCUMENTACIÓN</u>	23
<u>8.1</u>	<u>OBJETIVOS</u>	23
<u>8.2</u>	<u>CRONOGRAMA</u>	23
<u>8.3</u>	<u>DESARROLLO</u>	23
<u>8.4</u>	<u>RESULTADOS Y EVALUACIÓN</u>	23

2 Introducción

2.1 Objetivo

El objetivo de este documento es presentar el desarrollo y los resultados producidos en el proyecto de Taller 5 "Construcción de una herramienta para visualización de bases de conocimiento GeneXus", así como una breve descripción del producto obtenido. A su vez, este informe servirá como nexo entre los demás documentos producidos durante el proceso.

2.2 Alcance

En este documento se puede encontrar una descripción del proyecto, detallando las distintas etapas del mismo, así como justificaciones de las decisiones tomadas, ya sea de diseño o implementación, cuando ello corresponda.

2.3 Definiciones y Abreviaciones

2.3.1 Definiciones

2.3.1.1 Base de Conocimiento

Una base de conocimiento es un conjunto de información a partir de la cual GeneXus™ genera y mantiene la base de datos de la aplicación y todos los programas que acceden a ésta.

2.3.2 Siglas

- XML: eXtensible Markup Language
- JDBC: Java Database Connectivity
- ADO: ActiveX Data Objects
- KB: Knowledge Base (Base de Conocimiento)

2.4 Organización

Este informe se organiza en varios capítulos, cuyo contenido se presenta a continuación.

Este capítulo presenta una introducción al resto del documento, así como las definiciones de los términos que se utilizarán en el resto del informe.

El capítulo 2 (Descripción General) describe cómo se organiza el resto de la documentación que forma parte del proyecto, y que se presenta como documentación final del mismo.

El capítulo 3 (Presentación del Proyecto) da una breve introducción al proyecto, presentando el problema y los objetivos del mismo, así como una idea general del contexto en el cual se sitúa.

El capítulo 4 (Desarrollo del Proyecto) describe la organización del grupo de trabajo durante el proyecto, así como la metodología empleada y el cronograma planteado. Se comentan también en este capítulo las distintas etapas del proyecto y qué relación tuvo cada una de éstas con el cronograma planteado inicialmente.

El capítulo 5 (Etapa de Investigación) describe la etapa correspondiente a la investigación. Se detallan los objetivos de la misma y se comenta el cronograma para ésta. Además, se presenta el desarrollo de esta etapa, comentando los conocimientos obtenidos que sean importantes de destacar. Finalmente, se comentan los resultados y se da una breve evaluación de esta etapa.

El capítulo 6 (Etapa de Definición de Requerimientos) presenta el desarrollo de la fase de especificación de requerimientos, detallando los objetivos de la etapa, así como el cronograma para la misma.

El capítulo 7 (Etapa de Implementación) describe la implementación en un lenguaje de programación de la herramienta cuyas características se encuentran especificados en los documentos producidos anteriormente. De esta etapa se comenta el cumplimiento del cronograma, así como el desarrollo de la misma y los problemas encontrados. Finalmente se describirá brevemente el producto obtenido.

El capítulo 8 (Etapa de Documentación) presenta el desarrollo de la fase de documentación, y al igual que en los capítulos anteriores se presenta el cronograma planteado y el realizado, así como los documentos producidos en esta etapa.

3 Descripción General

En este capítulo se describe el resto de la documentación del proyecto, así como la relación que la misma tiene con este informe.

3.1 Organización de la documentación

Se presenta en esta sección una breve descripción de cada uno de los documentos que se incluyen como parte de la documentación final del proyecto. Los mismos se introducirán en el orden en que se redactaron a excepción del Informe Final que actúa como nexo entre los demás documentos presentados.

El primer documento que se presenta es el "Informe Final" del proyecto (este informe). El objetivo de este documento es presentar el proyecto y los resultados producidos en el mismo.

A continuación, se presenta el documento "Análisis de Requerimientos" [Crispino-Rozza-1], en el cual se especifican todas las características que deberá tener el producto final. Este documento es la base para la construcción del producto, ya que todo el resto del proceso y la documentación se basan en él.

Posteriormente se incluye el documento de "Diseño" [Crispino-Rozza-2] del producto, en el cual se detallan los distintos módulos que componen el sistema, así como la estructura básica de cada uno de éstos. Este documento tiene como principal objetivo el servir de base a la implementación.

El siguiente documento es "Interfaces Entre Módulos" [Crispino-Rozza-3] en el cual se explica de forma clara y no ambigua la forma que se utilizará para realizar la comunicación entre los módulos que componen el producto final.

Se incluye también el documento "Cómo Implementar un Algoritmo de Optimización" [Crispino-Rozza-5], que explica la forma de implementar nuevos algoritmos de optimización a partir de la estructura ya existente sin necesidad de modificar el código del resto de la aplicación.

Finalmente, se incluye el "Manual de Usuario" [Crispino-Rozza-4], en el cual se explica el funcionamiento del sistema y la forma de interactuar con él.

El apéndice A contiene las referencias bibliográficas que se utilizan en cada uno de los documentos anteriormente mencionados.

3.2 Cómo se debe leer este informe

El presente informe no pretende entrar en detalle en los distintos temas que trata. Los detalles se encuentran en el resto de los documentos, descritos en la sección anterior. Si se desea obtener información detallada en alguno de los tópicos aquí tratados, se deberá recurrir a los documentos anexos.

Se da a continuación una breve guía de los documentos que es recomendable leer en forma conjunta a los distintos capítulos.

En el capítulo 5, se recomienda leer el documento de "Manual de Usuario" [Crispino-Rozza-4], ya que en este documento se da una descripción más completa de los algoritmos de optimización implementados.

En el capítulo 6, se recomienda leer el documento de "Análisis de Requerimientos" [Crispino-Rozza-1], debido a que este documento fue el resultado de la etapa comentada en este capítulo.

En el capítulo 7 se recomiendan leer los documentos "Análisis de Requerimientos" [Crispino-Rozza-1], "Diseño" [Crispino-Rozza-2] y "Manual de Usuario" [Crispino-Rozza-4].

4 Presentación del Proyecto

Por intermedio de este capítulo, se introduce al proyecto, de forma de adquirir una idea general del mismo, así como de los objetivos que se pretenden alcanzar con el producto resultante.

4.1 Introducción

El presente proyecto se enmarca dentro de la asignatura Taller V de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la Universidad de la República. En particular, dicho proyecto corresponde al taller del año 2000.

Se trata de un proyecto que no está vinculado ni forma parte de otros, si bien su planteo inicial consistía en desarrollar una herramienta capaz de interactuar con la herramienta de desarrollo GeneXus™ de la empresa ARTech™. El desarrollo del mismo condujo a la construcción de una herramienta más general y totalmente independiente, que es capaz de interactuar con GeneXus™, pero que puede ser utilizada con otros fines.

4.2 Presentación del Problema

GeneXus™ brinda una forma de mostrar las tablas de un modelo de una base de conocimiento con sus relaciones, pero el diagrama que construye ubica las tablas alfabéticamente de forma de llenar una página a lo ancho y todas las que sean necesarias a lo alto. Este mecanismo es útil cuando se dispone de modelos chicos, de no más de veinte tablas.

La idea del proyecto surge cuando se intenta trabajar con modelos mucho más grandes, del orden de 500 tablas. Es evidente que en casos como éstos es cuando más se necesita contar con un diagrama de tablas claro, ya que la dificultad del mismo hace imposible que una persona conozca todas las tablas del modelo y cómo éstas se relacionan. También es evidente que el diagrama construido por GeneXus™ no es de utilidad en esta situación, porque la cantidad de líneas en el diagrama hace que sea imposible visualizar de forma clara las relaciones entre las tablas.

El objetivo de este proyecto es, entonces, construir una herramienta que facilite la visualización de un diagrama de tablas de un tamaño considerable, de una forma que resulte sencillo de usar con GeneXus™ y de tal forma que el resultado pueda ser obtenido en un tiempo razonable. Para esto, es necesario investigar la existencia de algoritmos capaces de resolver el problema y en el caso de no encontrar ningún algoritmo enteramente satisfactorio, desarrollar uno o adaptar el que se considere más adecuado.

El desarrollo del proyecto llevó a implementar una herramienta capaz de obtener la representación de un grafo, siempre que éste se encuentre en el formato especificado, no necesariamente representando un diagrama de tablas.

4.3 Integrantes

El grupo encargado de llevar a cabo el proyecto está formado por dos personas: Ana Rozza y Marcos Crispino. Además de los mencionados integrantes, Enrique Almeida y Gustavo Crispino se encargaron de la tutoría por parte de Concepto SRL y la Facultad de Ingeniería respectivamente.

4.4 Objetivo del Proyecto

El objetivo del proyecto es, como se mencionó anteriormente, la construcción de una herramienta que permita dibujar un diagrama de tablas de una base de conocimiento GeneXus™ de forma que el mismo sea fácil de entender por parte del usuario.

El producto debe contar también con una interfaz gráfica que resulte amigable para el usuario, y debe ser capaz de obtener el diagrama de bases de conocimiento relativamente grandes (del orden de las 500 tablas) en un tiempo razonable (del orden de minutos). No es posible dar una estimación más precisa, ya que el tiempo que demora la optimización de un grafo es altamente dependiente de la estructura del grafo, y de los parámetros que se elijan (se recomienda leer el "Manual de Usuario" [Crispino-Rozza-4] para obtener una explicación de los algoritmos y sus parámetros). En las pruebas realizadas, se ha encontrado por ejemplo que para algunas configuraciones de los parámetros, no se encuentra una representación planar para un grafo que sí lo es, aunque variando los parámetros se encuentra una solución planar rápidamente. Una de las pruebas realizadas fue para un grafo planar de 21 nodos en un Pentium Celeron de 466MHz con 32Mb de memoria RAM, la cual, con los parámetros por defecto, demoró 6 segundos en encontrar una representación planar. Con la misma configuración, un grafo de 65 nodos no planar, demoró 80 segundos, mientras que para este último grafo, con otros parámetros se obtuvo una solución comparable a la anterior en 10 segundos.

Se intenta que el diseño de la herramienta sea suficientemente general como para poder utilizar grafos que representen otras realidades que puedan resultar de interés. Por ejemplo, podría implementarse un algoritmo de optimización para resolver el problema del viajante, o podrían utilizarse los algoritmos ya implementados para visualizar el flujo de control de un programa. La utilización de grafos está tan extendida en el área de la computación, que las aplicaciones en las cuales se podrían necesitar una representación legible son muchas, y por lo tanto, implementar la herramienta de la forma más general posible puede resultar útil para otros casos.

5 Desarrollo del Proyecto

En este capítulo se describe la organización del grupo de trabajo durante el proyecto, así como la metodología de trabajo y el cronograma planteado. Se comentan también las distintas etapas del proyecto y qué relación tuvo cada una de estas con el cronograma planteado inicialmente.

5.1 Organización

Dentro del grupo no hubo una separación fija de roles, sino que todas las tareas fueron desarrolladas por ambos integrantes. En general la forma de trabajo fue durante todo el proyecto, primero pensar la solución al problema en grupo, luego realizar la tarea en forma individual, para finalmente revisar el resultado nuevamente en grupo. Algunas de las tareas realizadas durante el proyecto fueron llevadas a cabo en su totalidad en forma conjunta por parte de los dos integrantes del grupo, como por ejemplo la especificación de requerimientos o el diseño de la arquitectura del sistema.

Con respecto a la forma de interacción con los tutores, se prefirió establecer un mecanismo de comunicación mediante al uso de correo electrónico cada vez que surgía una duda o un problema, sobre la opción de establecer un cronograma de reuniones fijas. Esto se debe al carácter del proyecto, a la alta disponibilidad de los tutores durante todo el proyecto, y a que no se consideró necesario realizar un control demasiado estricto en este sentido.

5.2 Etapas

El desarrollo del proyecto puede separarse en cuatro etapas, las cuales se presentan a continuación y se detallan en los siguientes capítulos. Puesto que estas etapas se superponen en el tiempo, se presentarán según el orden en que comenzaron.

La primera etapa fue la etapa de investigación. Dentro de esta etapa, pueden reconocerse dos tareas: estudio de las herramientas a utilizar y estudio de los algoritmos que se utilizarían para obtener la representación gráfica del diagrama.

Dentro de la segunda etapa, se encuentra el planteo formal del problema, mediante la especificación de requerimientos del producto.

La tercera etapa fue la etapa de desarrollo, la cual implicó el diseño y la implementación de la herramienta en un lenguaje de programación, así como el testeo de la misma.

Finalmente, la última etapa consiste en la redacción del informe final, la redacción del manual de usuario de la aplicación y la integración de todos los documentos obtenidos durante el proceso.

5.3 Cronograma

A continuación se muestra el cronograma del proyecto, detallando cada una de las tareas realizadas. Este es el cronograma original del proyecto y fue confeccionado al comienzo del mismo.

Etapas	Tarea	Inicio	Fin	Duración
I	Estudio de Algoritmos (1ª parte)	15/03/00	30/04/00	6 semanas
	Estudio de Algoritmos (2ª parte)	01/08/00	30/09/00	4 semanas
	Estudio de Herramientas (Java)	01/04/00	14/05/00	6 semanas
	Estudio de Herramientas (GXPublic)	01/07/00	14/07/00	4 semanas
II	Análisis de Requerimientos	01/04/00	30/04/00	4 semanas
III	Diseño	01/05/00	31/05/00	4 semanas
	Implementación de Tipos de Datos Generales	15/05/00	31/05/00	2 semanas
	Definición de Interfases entre Subsistemas	15/05/00	31/05/00	2 semanas
	Implementación Módulo Visualizador	01/06/00	14/07/00	6 semanas
	Implementación Módulo Optimizador	15/08/00	14/10/00	8 semanas
	Implementación Módulo Lector de KBs	15/07/00	14/08/00	4 semanas
	Integración de Módulos	01/10/00	31/10/00	4 semanas
	Testeo	01/10/00	31/10/00	4 semanas
Corrección de Errores y Mejoras	01/11/00	30/11/00	4 semanas	
IV	Documentación Final	01/11/00	30/11/00	4 semanas

Tabla 1: Cronograma Original del Proyecto

La carga horaria semanal por integrante del equipo de trabajo se estimaba en unas 15 horas semanales. No se realizó ningún control formal sobre la cantidad de horas trabajadas en el proyecto, pero las estimaciones realizadas indican que en promedio, dicha carga horaria fue respetada, teniendo momentos de más trabajo y momentos en los cuales no se llegó a completar las 15 horas.

Durante el transcurso del proyecto se trató de seguir el cronograma, objetivo que fue cumplido de manera satisfactoria, aunque fue necesario realizar algunas modificaciones.

La primera etapa se llevó a cabo durante todo el proceso, poniendo especial énfasis en las fechas indicadas. Sin embargo, fue necesario realizar estudios, ya sea de algoritmos o del lenguaje de programación, durante todo el proceso, cada vez que surgía una dificultad o una idea nueva.

La etapa de análisis de requerimientos se cumplió según el cronograma, aunque fue necesario realizar pequeñas modificaciones de carácter aclaratorio en etapas posteriores, ninguna de las cuales afectó el contenido global del documento.

En la etapa de desarrollo se produjeron algunas desviaciones. El módulo Lector de KBs se desarrolló en menos tiempo que el esperado, pero el módulo Optimizador llevó unas dos semanas más de lo planificado. La integración entre los módulos se realizó en tan sólo una semana, debido a que durante el diseño se planificó el desarrollo de tal forma que la integración fuera lo más sencilla posible. El testeo y la corrección de errores del

producto no se llevaron a cabo en las fechas especificadas, sino que se realizaron durante el desarrollo de cada uno de los módulos. El resto de las tareas se realizaron de acuerdo al cronograma.

La etapa de documentación final fue postergada debido a la huelga de estudiantes y docentes en la Universidad de la República, por lo que no tiene demasiado sentido utilizar los datos reales de la duración de la etapa. Sin embargo, dados los resultados obtenidos a la mitad de esta etapa, podemos afirmar que la misma hubiera sido terminada según el cronograma de no haber sido por la circunstancia mencionada.

6 Etapa de Investigación

A continuación se describe la primera etapa, que es la etapa correspondiente a la investigación. Se detallan los objetivos de la misma y se comenta el cumplimiento del cronograma. Además, se presenta el desarrollo de esta etapa, comentando las conclusiones más relevantes obtenidas. Finalmente, se comentan los resultados y se da una breve evaluación de la etapa.

6.1 Objetivos

El objetivo de esta etapa es familiarizar a los integrantes del proyecto con los conocimientos que serán necesarios más adelante en el desarrollo de la herramienta.

Dentro de los conocimientos necesarios, se encuentran el estudio del lenguaje de programación Java™, el estudio de la herramienta GXPublic™ necesaria para obtener la información almacenada en las bases de conocimiento y la investigación de algoritmos existentes para la optimización de la representación de un grafo.

Durante el desarrollo del proyecto, surgieron otras áreas en las cuales fue necesario investigar. Las mismas son el metalenguaje de marcado XML y la existencia de alguna herramienta que permitiera conectar un driver JDBC (Java™) con un proveedor ADO (GXPublic™).

6.2 Cronograma

Inicialmente, la etapa de investigación aparecía dividida en dos partes: estudio de algoritmos de optimización y estudios de herramientas. A su vez, la etapa de estudio de algoritmos estaba dividida en dos; una al comienzo del proyecto y otra, la de profundización de los conocimientos, antes de comenzar la implementación de los algoritmos. Esta separación no existió en la práctica, ya que esta tarea se realizó durante todo el proyecto, poniendo especial énfasis en las semanas anteriores al comienzo de la implementación de los algoritmos.

Por su parte, el estudio de herramientas aparecía también dividido en dos partes, las cuales fueron respetadas según el cronograma inicial. La primera parte consistía en el estudio del lenguaje de programación Java™ y la segunda en el estudio de GXPublic™. A estas dos tareas, fue necesario sumarle el estudio del lenguaje de marcado XML, el cual se realizó paralelamente a la etapa de diseño.

6.3 Desarrollo

Esta etapa fue dividida en cuatro partes, cada una de las cuales se comenta a continuación. Para cada una de estas partes, se detallará el material utilizado, así como las conclusiones obtenidas.

6.3.1 Estudio del lenguaje de programación Java™

Esta tarea se había planificado para las primeras semanas del proyecto, en las cuales se realizó la mayor parte del aprendizaje. Sin embargo, fue necesario volver a la bibliografía constantemente durante la etapa de desarrollo para aclarar dudas y adquirir nuevos conocimientos.

Para el estudio inicial del lenguaje de programación se utilizó el tutorial [JavaTutorial99]. Posteriormente, durante la etapa de desarrollo se utilizó extensamente las API de Java [JavaAPI99] y los libros [Horstmann99] y [Horstmann00] principalmente cuando surgía alguna duda de implementación.

6.3.2 Estudio de GXPublic

Para el estudio de la herramienta GXPublic se utilizó el manual de la misma. Esta herramienta cuenta con funcionalidades que van más allá de lo necesario en este proyecto, por lo que la investigación se redujo sólo a aquellas partes que fueran relevantes, a saber, la estructura de las tablas que utiliza y la forma de conectarse para obtener la información.

La forma que utiliza GXPublic para brindar información, es mediante la implementación de un proveedor ADO, al cual puede accederse como si fuera una base de datos relacional. Sin embargo, los servicios que ofrece son limitados, lo cual llevó a tener que modificar la implementación inicial del módulo que accede a la base de conocimiento, ya que estas limitaciones no se habían tomado en cuenta. Las limitaciones en cuanto a las consultas que pueden realizarse sobre GXPublic son:

- sólo se permite realizar consultas SELECT sobre toda la tabla y no sobre atributos específicos
- no se pueden utilizar consultas que realicen un JOIN sobre varias tablas.

6.3.3 Estudio de XML

Al igual que con GXPublic™, el lenguaje extensible de marcado XML es mucho más extenso y potente que lo que se necesitaba conocer para el proyecto. Es por esto que el estudio de XML se restringió únicamente a los conocimientos estrictamente necesarios para la implementación de la herramienta.

El documento con el que se inició el estudio de XML fue [XMLTutorial99], el cual brindó la mayor parte de los conocimientos necesarios para la utilización de este lenguaje. Sin embargo, fue también necesario estudiar en parte el tutorial [JAXPTutorial00] para comprender la forma de integrar este lenguaje con el lenguaje de programación Java™.

El metalenguaje XML permite definir estructuras jerárquicas de forma muy sencilla, y con un formato que resulta fácil de leer y de procesar por parte de un parser, ya que la estructura jerárquica en que se almacenan los datos en el archivo XML está definida en el mismo archivo, lo cual hace que sea más manejable y adaptable que por ejemplo un archivo de texto plano.

6.3.4 Estudio de Algoritmos de Optimización

Para los algoritmos de optimización se buscó inicialmente la existencia de un algoritmo en Internet que sirviera para cualquier grafo y que fuera eficiente. Esta búsqueda se realizó sin lograr resultados totalmente satisfactorios, ya que el objetivo inicial de la búsqueda era obtener un algoritmo que encontrara la representación óptima para un grafo.

La búsqueda en Internet se llevó a cabo en buscadores de propósito general, como ser AltaVista, Yahoo y MetaCrawler, encontrando algunos sitios con contenido de interés, de los cuales se destacan los de mayor utilidad:

- Graph Drawing
(<http://davis.wpi.edu/~matt/courses/graphs/>)
contiene información sobre los distintos tipos de diagramas que pueden utilizarse, así como los criterios más comunes de legibilidad de grafos.
- Interactive Graph Drawing
(<http://www.cs.rpi.edu/projects/pb/graphdraw/>)
contiene una aplicación en la cual se dibujan los nodos y aristas de un grafo predefinido. Se utilizó como ejemplo para la implementación de la interfaz gráfica.
- The Hitch-Hiker's Guide To Evolutionary Computation
(<http://www.cs.bham.ac.uk/Mirrors/ftp.de.uu.net/EC/clife/www/>)
Frequently Asked Questions sobre algoritmos evolutivos.
- Artificial Life and Genetic Algorithms
(<http://http1.brunel.ac.uk:8080/depts/AI/alife/home.htm>)
contiene información sobre algoritmos genéticos, explicando los componentes del mismo.

Por lo tanto, dado que no se encontró ningún algoritmo que permitiera encontrar la representación óptima para cualquier grafo, se comenzó a estudiar la posibilidad de implementar otros tipos de algoritmos. Los tipos de algoritmos que fueron considerados son los algoritmos evolutivos y las heurísticas. Las ideas para la implementación de estos algoritmos surgieron de la búsqueda antes mencionada, que si bien no logró encontrar un algoritmo óptimo, fue suficientemente buena como para obtener las ideas necesarias para implementar los algoritmos. Algunos de los libros utilizados fueron adquiridos por la empresa Concepto S.R.L. a partir de recomendaciones encontradas en Internet.

6.3.4.1 Algoritmos Heurísticos

Para las heurísticas, se utilizó como referencia [DiBattista99], realizando modificaciones a los algoritmos cuando se consideró necesario.

De esta clase de algoritmos finalmente se implementó uno que elegimos llamar "Algoritmo de las Fuerzas". Este algoritmo considera a cada nodo como una carga eléctrica positiva, y a cada arista como un resorte, con una elongación natural y una constante de estiramiento que son comunes a todas. El planteamiento de este modelo tiene como consecuencia que sobre cada nodo se aplica una fuerza y el algoritmo mueve de a un nodo a la vez de forma tal de minimizar la fuerza que se aplica sobre el mismo. Este modelo permite que el grafo se aproxime a un punto de equilibrio, en el cual la fuerza que se ejerce en cada nodo es mínima.

Se decidió implementar también una variante de este algoritmo, en el cual el modelo planteado es el mismo, pero al grafo original se le quitan los nodos de grado uno para realizar la optimización, y luego éstos se agregan en el momento en que se construye la representación. Este algoritmo es más rápido que el anterior solo en aquellos grafos que contienen un número grande de nodos de grado uno.

Finalmente, como algoritmo de inicialización, esto es el algoritmo que construye la primera representación a partir del grafo, se implementó un algoritmo que llamamos "Algoritmo del Baricentro". Este algoritmo, ubica inicialmente los nodos de mayor grado en los vértices de un rectángulo y luego agrega los demás en el baricentro formado por todos los adyacentes que ya se encuentran en la representación que se está construyendo. En la bibliografía consultada este algoritmo se aplica a grafos conexos, por lo cual fue necesario adaptarlo para poder utilizarlo con grafos que tienen más de una componente conexas.

El libro utilizado como referencia contenía otros algoritmos, pero éstos eran únicamente para casos particulares, como por ejemplo árboles o diagramas de circuitos eléctricos, por lo cual no eran de utilidad en nuestro caso.

6.3.4.2 Algoritmos Evolutivos

Para los algoritmos evolutivos, se utilizaron [Fogel00], [Michalewicz99], [Haupt98] y varios sitios en Internet con información sobre el tema.

De este tipo de algoritmos solamente se implementó uno, ya que el rendimiento y la velocidad de convergencia del mismo fue inferior al algoritmo de las Fuerzas y sus variantes. El motivo por el cual este algoritmo no es eficiente, es por la enorme cantidad de operaciones que son necesarias realizar para evaluar la función de adecuación de cada representación gráfica.

Cabe destacar que en la bibliografía consultada, no se mencionaba el uso de estos algoritmos para optimizar representaciones de grafos, pero sí para problemas que usan grafos como por ejemplo en problema del viajante.

6.4 Resultados y Evaluación

Los estudios del lenguaje de programación Java™ y la herramienta GXPublic™, permitieron desarrollar la aplicación, de tal forma de cumplir con los requerimientos especificados.

El lenguaje XML fue utilizado con éxito para realizar la comunicación entre los módulos que componen la herramienta. Este lenguaje permite representar de manera jerárquica y muy simple estructuras anidadas, como por ejemplo que un grafo tiene un conjunto de nodos y un conjunto de aristas, que un nodo tiene un nombre e información asociada, y que una arista debe tener un origen y un destino. El estudio de la relación entre este lenguaje de marcado y el lenguaje de programación llevó a que la implementación de la comunicación entre los módulos fuera sencilla de implementar y que no presentara mayores dificultades.

Por último, el estudio de los distintos algoritmos, llevó a implementar algoritmos capaces de obtener buenas representaciones para grafos, que era el objetivo más importante del proyecto. El algoritmo de las Fuerzas se implementó con pequeñas modificaciones con respecto al que se encuentra en [DiBattista99]. El algoritmo Compactar fue una variante del mismo que fue diseñada por los integrantes del grupo a partir del algoritmo de las Fuerzas. El algoritmo Genético fue una implementación de los algoritmos evolutivos a un problema en el cual no se encontró que se hubiera utilizado con anterioridad, si bien el esqueleto del algoritmo ha sido utilizado con muy buenos resultados en otras áreas.

7 Etapa de Definición de Requerimientos

En este capítulo se describe la etapa de especificación de requerimientos. Se detallan los objetivos de esta etapa y se comenta el cronograma de la misma. Además, se presenta el desarrollo de la etapa, comentando el proceso de especificación de requerimientos y el documento generado. Finalmente se comentan los resultados y se presenta una breve evaluación de la etapa.

7.1 Objetivos

El objetivo de esta etapa es especificar de forma clara y no ambigua los requerimientos de la aplicación a desarrollar. Es importante que el documento generado en esta etapa sea completo, es decir que no falten requerimientos, y que además los requerimientos queden bien claros, ya que será la base de las etapas de implementación y documentación.

7.2 Cronograma

Esta etapa cuenta con una única actividad que es el análisis de requerimientos, tarea que se llevó a cabo al comienzo del proyecto junto con el estudio de herramientas.

La etapa de análisis fue terminada según el cronograma planteado. Se puso especial énfasis en la culminación en fecha de esta etapa, ya que es necesario contar con la especificación de requerimientos para poder realizar el resto de las tareas involucradas en el proyecto.

7.3 Desarrollo

Durante la duración de esta etapa, los dos integrantes del grupo estuvieron enfocados principalmente en la elaboración del documento de especificación de requerimientos. Para esto fueron necesarias varias reuniones con el tutor Enrique Almeida, así como numerosas revisiones del documento por parte del tutor Gustavo Crispino.

El formato de documento utilizado en la especificación de requerimientos se basó en el estándar [IEEE-830].

7.4 Resultados y Evaluación

El resultado de esta etapa fue un documento conteniendo los requerimientos de la herramienta a desarrollar, especificados de manera formal. El mismo se presenta junto con este informe y puede encontrarse en [Crispino-Rozza-1].

El documento obtenido fue utilizado posteriormente para el diseño de la arquitectura del sistema y para la etapa de implementación, cumpliendo el objetivo del mismo. Cabe destacar que al comienzo de la etapa de implementación fue necesario realizar pequeñas modificaciones a este documento, aunque estas fueron mayormente de carácter aclaratorio y no fueron modificaciones que afectaran el contenido del mismo.

8 Etapa de Implementación

En este capítulo se describe la etapa de implementación de la herramienta en un lenguaje de programación, la cual fue la más costosa en horas de trabajo y en duración. Se detallan los objetivos de la etapa, así como el cronograma planteado y el realizado. Además, se presenta el desarrollo de la etapa, para finalmente concluir con un resumen de los productos obtenidos y una evaluación de la misma.

8.1 Objetivos

El objetivo de esta etapa es el de implementar en un lenguaje de programación de alto nivel una herramienta que cumpla con los requisitos planteados en el documento de especificación de requerimientos [Crispino-Rozza-1]. Esta herramienta debe ser suficientemente buena, para que la misma pueda ser realmente utilizada por aquellas personas que la necesiten.

8.2 Cronograma

La etapa de implementación aparece dividida en nueve tareas claramente diferenciables, algunas de ellas netamente de implementación y otras dedicadas a tareas auxiliares. Estas nueve tareas están repartidas durante todo lo largo del proyecto, haciendo que esta etapa sea claramente la que más trabajo llevó.

El diseño de la arquitectura del sistema se llevó a cabo inmediatamente después de la etapa de especificación de requerimientos y fue la primer tarea de esta etapa. Podríamos decir que el diseño es la piedra fundamental de la construcción de la herramienta, ya que es en el documento generado por esta tarea que se basa la implementación de los componentes. Esta etapa tuvo una duración de aproximadamente un mes y en ella se produjo el documento de "Diseño" [Crispino-Rozza-2].

La siguiente tarea, que se llevó a cabo durante las últimas dos semanas del diseño de la herramienta, fue la implementación de lo que denominamos "tipos de datos generales". Por tipos de datos generales se entienden aquellos que son compartidos por todos los componentes del sistema, como ser grafo, nodo y arista.

Durante este mismo período se realizó la definición de la forma en que se comunican los distintos subsistemas que componen la herramienta. No es fácil determinar el tiempo dedicado a esta tarea, ya que la mayor parte del trabajo consiste en idear la forma en que los módulos se comunican. La redacción del documento de especificación de estas interfaces se realizó en poco tiempo, cumpliendo con el plazo establecido y dando como resultado el documento "Interfaces entre Módulos" [Crispino-Rozza-3].

Luego de estas tres tareas que se realizaron de forma simultánea, se comenzó con la implementación del módulo Visualizador, la cual tuvo una duración aproximada de seis semanas. Cabe destacar que al cabo de las seis semanas, aún quedaba una lista de puntos pendientes que estaban por fuera de la especificación de requerimientos, por ser requerimientos secundarios o ideas que surgieron durante el desarrollo del módulo y que fueron realizados luego de terminar la implementación de los dos módulos restantes.

La quinta tarea de la etapa fue la implementación del módulo Lector de KBs, que es el encargado de comunicarse con GXPublic para obtener la información relevante de una base de conocimiento. El cronograma inicial planteaba el desarrollo de este módulo en cuatro semanas, sin embargo el desarrollo se llevó a cabo en dos semanas. Esta discrepancia se debe a que el uso de GXPublic resultó ser más sencillo que lo esperado, lo cual no se sabía en el momento en que se diseñó el cronograma.

Luego de esto, se implementó el módulo Optimizador que es el último de los tres módulos que componen el sistema. A diferencia del Lector de KBs, la implementación de este módulo fue más difícil que lo pensado inicialmente, por lo que esta etapa fue más larga que lo esperado, lo cual compensó las dos semanas ganadas durante la implementación del módulo Lector de KBs.

Inicialmente se destinaron cuatro semanas a la integración entre los módulos, las cuales no fueron necesarias ya que el diseño de la herramienta permitió desarrollar cada módulo pensando en la integración del mismo con los demás. Esto llevó a que la fase de integración no se realizara en las cuatro semanas especificadas, sino que la misma se realizó junto con el desarrollo de cada módulo, haciendo que cada uno cumpliera con el formato especificado en el documento de interfaz [Crispino-Rozza-3].

La etapa destinada al testeo de la aplicación debía durar cuatro semanas y llevarse a cabo junto con la integración entre los módulos. Sin embargo, dada la independencia entre los módulos, se prefirió testearlos individualmente al final de cada tarea de implementación. El testeo global fue realizado por el tutor Enrique Almeida, utilizando el producto como "beta tester", ya que comenzó a utilizar la herramienta aún antes de que ésta estuviera terminada. A medida que surgían errores, los mismos se corregían de acuerdo a su prioridad.

Finalmente, existió una etapa de corrección de errores que se realizó entre la finalización del módulo Visualizador y el final del proyecto y no en las cuatro semanas que tenía destinada. Esta modificación al cronograma se produjo porque cada vez que se detectaba un error, era necesario arreglarlo para poder continuar utilizando el módulo afectado.

8.3 Desarrollo

En esta sección se comenta el desarrollo de cada una de las tareas que componen la etapa de implementación.

8.3.1 Diseño

Durante esta etapa se realizó el documento de "Diseño" [Crispino-Rozza-2], en el cual se detalla la arquitectura del sistema.

Con la idea de lograr la mayor independencia posible entre la herramienta desarrollada y GeneXus™, se decidió implementar tres módulos, cada uno con una tarea específica.

Estos módulos son:

- i) el Lector de KBs, que se encarga de obtener la información de una base de conocimiento GeneXus™
- ii) el Optimizador, que se encarga de construir la representación gráfica del diagrama

- iii) el Visualizador, que se utiliza para mostrar el diagrama resultante junto con la información que tenga asociada.

El documento especifica los componentes que deberá tener cada uno de los módulos, así como la relación y la separación entre ellos.

Para el desarrollo de la herramienta, se decidió utilizar el lenguaje de programación Java™, debido a las facilidades que ofrece su orientación a objetos y la cantidad de componentes estándar, como primitivas para el dibujo en dos dimensiones o componentes de interfaz gráfica, que facilitan el desarrollo de nuevos componentes. Otro de los motivos por el cual se eligió este lenguaje es el conocimiento previo del mismo (parcial) que poseían los integrantes del grupo.

8.3.2 Datos Básicos

Los datos básicos del sistema son aquellos que se utilizan en los tres módulos, como ser los objetos *Grafo*, *Nodo*, *Arista*, *GrafoRep* (representación gráfica del grafo), etc.

8.3.3 Interfaz de Comunicación entre Módulos

Para definir la interfaz de comunicación que se utilizaría entre los módulos del sistema, se optó por el lenguaje de marcado XML, debido a la facilidad para definir estructuras jerárquicas, la fácil integración con el lenguaje de programación elegido y la facilidad que ofrece para extenderlo para representar otras realidades. Así mismo, se consideró interesante el aprendizaje de este lenguaje debido a la aceptación que el mismo ha tenido en los últimos tiempos como herramienta para la transferencia de información entre sistemas.

El documento producido [Crispino-Rozza-3] contiene la estructura que deberá tener el archivo que se utiliza para comunicar los distintos módulos. Si se desea reemplazar alguno de los módulos del sistema, por ejemplo para diagramar otro tipo de información, el nuevo módulo deberá cumplir con la estructura especificada en este documento.

8.3.4 Lector de KBs

Este módulo se encarga de obtener la información de una base de conocimiento GeneXus™, para lo cual fue necesario comunicar un driver JDBC (Java™) con un proveedor ADO (GXPublic™). Inicialmente se planteó desarrollar este módulo en otro lenguaje, como por ejemplo Visual Basic, para evitar este tipo de problemas, pero finalmente se optó por utilizar un driver que permite desde Java™ conectarse a un proveedor ADO. El mismo pertenece a la empresa InfoZoom (<http://www.infozoom.de/IE/javaado.html>), y la versión utilizada fue la versión de evaluación que puede usarse por 30 días.

Este es el único de los tres módulos que interactúa con GeneXus™, logrando que la herramienta pueda ser adaptada a otras realidades sustituyendo un único módulo.

8.3.5 Optimizador

Este módulo es el encargado de tomar la estructura de un grafo y a partir de ésta construir una buena representación. Para esto, se desarrolló un módulo que cuenta con

una interfaz gráfica que permite al usuario seleccionar los algoritmos a utilizar, así como ver el progreso de la optimización.

Para desarrollar este módulo, fue necesario primero estudiar los algoritmos existentes y adaptarlos para mejorar una representación gráfica de un grafo, los cuales se detallan en la sección 5.3.4 de este documento.

El Optimizador se desarrolló de tal forma que resultara sencillo agregar nuevos algoritmos de optimización iterativos, sin necesidad de realizar modificaciones a los algoritmos ya implementados. El documento [Crispino-Rozza-5] detalla la forma de hacer ésto.

8.3.6 Visualizador

El módulo Visualizador es el encargado de mostrar al usuario la representación obtenida para el grafo, junto con la información asociada a cada nodo. Además, este programa permite modificar la estructura del grafo agregando o quitando nodos y aristas.

Cuenta con una interfaz gráfica que permite realizar operaciones como alejar o acercar la representación (zoom), ver la información asociada a un nodo, centrar el diagrama en un nodo, imprimir el diagrama, modificar los parámetros de visualización, etc.

8.3.7 Integración de Módulos

Si bien los tres módulos deben ser capaces de actuar por separado y la comunicación entre los mismos ya estaba definida, por razones de rendimiento se decidió implementar un cuarto módulo, de nombre Diagramador, que integra los otros tres. Este módulo invoca a los otros tres en orden, y obtiene los grafos y representaciones intermedias como datos básicos, y no como archivos con estructura XML. Esto hace que el pasaje de un módulo a otro sea mucho más rápido, ya que no es necesario escribir a disco y luego leer nuevamente.

8.4 Resultados y Evaluación

El resultado de esta etapa fue una herramienta que cumple con los requerimientos especificados en [Crispino-Rozza-1], y que está siendo utilizada dentro de la empresa Concepto S.R.L. para obtener diagramas de bases de conocimiento de tamaño mediano (de unas 200 tablas) con un rendimiento comparable con el esperado al comienzo del proyecto.

Una primer versión de esta herramienta, la cual no contaba aún con todas las funcionalidades, fue presentada en el "XI Encuentro de Usuarios GeneXus" con muy buenos resultados, ya que despertó el interés de varios de los asistentes a la conferencia.

9 Etapa de Documentación

En este capítulo se describe la etapa de documentación. Se detallan los objetivos de la etapa, así como el cronograma planteado y el realizado. Además, se presenta el desarrollo de la etapa, para finalmente concluir con un resumen de los productos obtenidos y una evaluación de la etapa.

9.1 Objetivos

El objetivo de esta etapa es producir todos los documentos necesarios para completar el proyecto, entre los cuales se encuentra el Informe Final (este documento) y el Manual de Usuario [Crispino-Rozza-4].

9.2 Cronograma

Inicialmente se había planteado el mes de noviembre como fecha para realizar la documentación final del sistema, pero debido a la huelga que afectó a la Facultad de Ingeniería, los plazos no pudieron ser mantenidos. Por este mismo motivo, no tiene sentido comentar el cronograma real de esta etapa.

9.3 Desarrollo

La documentación del proyecto se llevó a cabo en su mayor parte sobre el final del mismo, por parte de los dos integrantes. Sin embargo, se intentó documentar los aspectos más relevantes del proyecto a la largo del mismo, para facilitar la última etapa.

9.4 Resultados y Evaluación

El Informe Final será utilizado como base para la defensa del proyecto, y se utiliza como nexo entre los demás documentos cumpliendo así su objetivo.

El Manual de Usuario [Crispino-Rozza-4], no ha sido utilizado aún debido a que las personas que utilizan esta herramienta han acompañado el desarrollo de la misma y por lo tanto no necesitan dicho documento. Sin embargo, se prevé que en el corto plazo, más personas comiencen a utilizar esta herramienta, para lo cual será útil contar con el Manual de Usuario.