



Universidad de las Ciencias Informáticas

Facultad 1

**Trabajo de Diploma para optar por el título de
Ingeniera en Ciencias Informáticas**

Extensión de Vista paralela para LibreOffice Writer.

Autora:

Yamilka Leyva Coss

Tutores:

Msc. Maidely Calderón Montero

Ing. Yordanis Cabreja Núñez

Ing. Adriam Delgado Rivero

La Habana, Junio de 2014

“Año 56 de la Revolución”

Declaración de autoría

Declaración de autoría

Declaro ser la autora principal del trabajo de diploma “Extensión de Vista paralela para LibreOffice Writer” y autorizo a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de Junio del 2014.

Yamilka Leyva Coss

Msc. Maidely Calderón Montero

Ing. Yordanis Cabreja Núñez

Pensamiento



"Cuando fui a la escuela, me preguntaron que quería ser de mayor. Yo respondí: "feliz". Me dijeron que yo no entendía la pregunta, y yo les respondí que ellos no entendían la vida. "

John Lennon

Dedicatoria

Dedicatoria

A mi hermano, no porque estés ausente, porque tu presencia está en cada uno de nosotros, a tí porque están tus actos, estás en lo que digo y en lo que soy. Te atesoro y te recuerdo en mis memorias, a tí, porque vivirás en mí y por permitirme sentir esa clase de amor por el que la gente muere sin dudarlo.

A mis padres, porque soy el fruto de su esfuerzo, por ayudarme a crecer, porque son mi razón de ser, la luz de mi mundo. Nadie ha recibido más afecto ni apoyo incondicional como el que ustedes me han dado. Yo... también los quiero.



Agradecimientos

Agradecimientos

A mis padres

Por ser los principales protagonistas de mi vida, les agradezco infinitamente por tanto amor, por enseñarme que los obstáculos son esas cosas espantosas que veo cuando aparto mis ojos de la meta.

A mi hermano Yoel

Por ser mi mayor fuente de inspiración y por enseñarme que no puedo elegir como me siento pero siempre puedo hacer algo.

A mi mejor amigo y compañero Yasiel

Por ser mi sol de medianoche y darme siempre su apoyo incondicional, eres tan especial que podrías encontrar el amor hasta en los rincones más oscuros del universo.

A mis personitas especiales

Dignay, Adrian, Isabel, Iye, Hanoi y Arlhey que me demostraron que existen los amigos, les enviaré luz y amor cada vez que piense en ustedes, los quiero tanto que el corazón se me hace pequeño.

A mis compañeros

Dainelys, Daniel, Gustavo, Javier, Dani, Yosvani, Juan Carlos, Katy, Iván, Jorgito, Marlene, Leandro, Eva y Carlos por permitirme compartir con ellos estos cinco años y por estar ahí en los momentos de alegría y dolor.

A mi familia

Que ha sido constante estímulo para seguir adelante ante cada dificultad.

A mi queridísima profe Matilde

Por la fuerza de sus palabras: “Llora si es necesario para ti manifestar una emoción, pero continúa con tu camino. No te detengas ante el dolor y convierte tu debilidad en fuerza”

A mis tutores

Maidely, Eduardo, Adriam y Yordanis por su tiempo y dedicación para que este trabajo saliera adelante

A todos los profesores

Agradecimientos

Que durante estos cinco años han contribuido de una forma u otra a mi formación profesional, en especial a la profe Damaris.

A aquellos que me preguntaban en los pasillos

¿Cómo va la tesis? ¿Cuándo discutes? ¿Qué te falta para terminar?

A todos

Gracias. ... sin ustedes este día no sería posible.



Resumen

Resumen

Debido a la necesidad de comparar dos versiones distintas de un mismo texto o consultar dos documentos diferentes a la vez, surge la idea de desarrollar una extensión para proporcionarle la funcionalidad de Vista paralela a la aplicación LibreOffice Writer, lo cual es el objetivo fundamental del presente trabajo; la misma favorece la labor de lectura, minimiza los esfuerzos y reduce el tiempo de revisión de los documentos. Para lograr el cumplimiento del objetivo propuesto fue imprescindible conocer fundamentos del manejo de archivos en el sistema operativo NOVA, se configuró y preparó el ambiente de trabajo de LibreOffice para el desarrollo de extensiones y fue necesario tener conocimientos del lenguaje LibreOffice *Basic*. Para apoyar la investigación se utilizó bibliografía actualizada sobre el desarrollo de componentes para dicha *suite* de oficina.

Palabras claves: extensión, LibreOffice Writer, Vista paralela.

Índice

Índice de contenido

Introducción	11
Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice.	16
1.1 Referentes teórico-metodológicos	16
1.1.1 Definiciones de extensión	16
1.1.2 Estándar ODF	18
1.1.3 Formato de compresión ZIP	18
1.1.4 UNO	19
1.1.5 El lenguaje IDL.....	19
1.1.6 Gestor de extensiones	20
1.1.7 Contenedor de Bibliotecas	20
1.1.8 Bibliotecas	20
1.1.9 Bibliotecas en tiempo de ejecución	21
1.1.10 Interfaz	21
1.1.11 Servicios	21
1.2 Estructura de las extensiones.....	21
1.2.1 El archivo <i>description.xml</i>	22
1.2.2 El archivo <i>manifest.xml</i>	22
1.2.3 El archivo Paths.xcu.....	22
1.2.4 Versión de una extensión.....	22
1.2.5 Licencia de una extensión.....	23
1.2.6 Nombre para mostrar	23
1.2.7 Descripción de una extensión	23
1.2.8 Ícono de una extensión.....	23
1.2.9 La carpeta descripción de una extensión	24
1.3 Alcance de una extensión	24
1.4 LibreOffice y sus extensiones	24
1.5 La herramienta Ver en paralelo de Microsoft Word.....	25
1.6 Metodología de desarrollo de software SXP	26
1.7 Herramientas y tecnologías a utilizar en el desarrollo de la extensión.	27
1.7.1 Herramienta de modelado Visual Paradigm 8.0.....	27
1.7.2 API UNO	27
1.7.3 Extensión BasicAddonBuilder 0.3.5	28
1.7.4 Entorno de Desarrollo Integrado	28

Índice

1.7.5 Lenguaje de programación LibreOffice <i>Basic</i>	29
1.7.6 El servidor Subversion 1.16.17 y la herramienta de desarrollo colaborativo RapidSVN 0.12.0	30
Conclusiones del capítulo	31
Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer	32
2.1 Propuesta de solución.....	32
2.2 Requisitos Funcionales y No Funcionales del producto.....	33
2.3 Historias de Usuario	35
2.4 Implementación de los requisitos	37
2.4.1 Tareas de ingeniería (TI).....	37
2.5 Conclusiones de la fase de Análisis	40
2.6. Arquitectura definida para la aplicación	40
2.7 Modelo de implementación	40
2.7.1 Diagrama de componentes del diseño.....	41
2.7.2 Diagrama de paquetes del diseño	45
Conclusiones del capítulo	51
Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer.....	52
3.1 Planificación de la implementación.....	52
3.2 Estándar de codificación.....	53
3.3 Modelo de pruebas.....	55
Conclusiones del capítulo	57
Conclusiones generales	58
Recomendaciones.....	59
Referencias	60
Bibliografía.....	62
Glosario de términos.....	64
Anexos	66
.....	89

Índice

Índice de tablas

Tabla 1: Requisitos Funcionales (RF).....	34
Tabla 2: Requisitos No Funcionales (RNF).....	34
Tabla 3: HU Gestionar documentos.....	35
Tabla 4: HU Mostrar barra de herramienta	36
Tabla 5: HU Activar Ver en paralelo.....	37
Tabla 6: TI 1 Implementar la macro para mostrar los documentos.....	37
Tabla 7: TI 2 Implementar la macro para redimensionar los documentos	38
Tabla 8: TI 3 Implementar la macro para situar los documentos en paralelo.....	38
Tabla 9: TI 4 Diseñar el icono de la extensión.....	39
Tabla 10: TI 5 Empaquetar las macros creadas.....	39
Tabla 11: Caso de prueba de aceptación 1	56
Tabla 12: Caso de prueba de aceptación 2	56
Tabla 13: Caso de prueba de aceptación 3	57

Índice de figuras

Figura 1: Documentos en paralelo al ejecutar la extensión.....	33
Figura 2: Diagrama de componentes del diseño parte 1	43
Figura 3: Diagrama de componentes del diseño parte 2	44
Figura 4: Diagrama de paquetes del diseño parte 1.....	46
Figura 5: Diagrama de paquetes del diseño parte 2.....	47

Introducción

Introducción

Organización y tecnología sería la forma más sencilla de expresar el significado que sugiere la palabra ofimática, pues justamente es la perspectiva que estudia el uso de tecnologías de la información en el trabajo de las oficinas. Esta es posibilitada por una combinación entre *hardware* y *software* que permite crear, manipular, almacenar y transmitir digitalmente la información que se necesita en una oficina para realizar las tareas cotidianas y alcanzar sus objetivos.

Debido a los inmensos volúmenes de información que deben procesar muchos organismos, y contando con los adelantos tecnológicos de la época, el uso intensivo de los equipos de cómputo se ha convertido en una necesidad para dar solución a los diversos problemas que involucra el manejo de la información. Por esta razón se ha de emplear una serie de herramientas ofimáticas que permitan a los usuarios acceder a la información de forma eficiente, garantizando así el desarrollo normal de las actividades y el cumplimiento de las tareas.

Existe un conjunto de técnicas y aplicaciones informáticas que simplifican la manera de acceder, compartir y analizar la información, tal es el caso de las *suites* de oficina, ejemplos de estas son LibreOffice, Microsoft Office, Apache OpenOffice, Microsoft Surface entre otras; su uso ha permitido optimizar, automatizar y mejorar los procedimientos relacionados con el procesamiento de documentos. Dichas herramientas pueden ser libres o privativas, desafortunadamente no todos pueden tener acceso a estas últimas, pues deben comprar la licencia a un alto precio, por esta y otras razones el *Software Libre* y el Código Abierto han surgido para ofrecer excelentes oportunidades a países e instituciones que no pueden acceder a estas costosas aplicaciones propietarias.

Migrar hacia soluciones libres facilitaría el desarrollo autónomo de Cuba en el campo de la informática y las comunicaciones, puesto que alcanzaría un estado de soberanía tecnológica, en el cual será capaz de decidir sobre la forma en que se desarrollen y usen sus tecnologías, de acuerdo a los principios en los que se basa el *Software Libre*, “libertad, soberanía e independencia”, poniendo a disposición de la sociedad cubana los conocimientos y avances

Introducción

científicos. Como parte de la estrategia orientada a la migración hacia *Software Libre* y aplicaciones de Código Abierto, el departamento de Servicios Integrales de Migración, Asesoría y Soporte (SIMAYS) cuenta con el proyecto Personalización Ofimática, cuyos miembros se encuentran desarrollando nuevas funcionalidades para LibreOffice, debido a la excelente recopilación de aplicaciones que esta *suite* brinda a los usuarios, las cuales contribuyen a la creación y modificación de archivos y documentos.

Los usuarios de la *suite*, que trabajan en las oficinas en las cuales se ha llevado a cabo el proceso de migración, han planteado la necesidad de trabajar con dos documentos que requieren ser revisados a la vez; esta tarea se puede realizar redimensionando los documentos manualmente para acomodarlos en pantalla o minimizando uno para acceder al otro, lo cual se convierte en una labor iterativa que dificulta la asimilación de la información que se analiza y representa una pérdida de tiempo y esfuerzo. Dicha tarea es sumamente importante en muchas áreas, puesto que en ocasiones es necesario comparar dos versiones distintas de un mismo documento; tal es el caso de un editor revisando sus observaciones; un autor comparando la versión anterior de su texto; un estudiante analizando los comentarios de sus lectores de tesis; o simplemente usuarios que tienen dos documentos, los cuales le interesa ir revisando a la vez, bien sea porque son parecidos o bien porque el contenido de uno de ellos puede serle útil para copiar partes de un texto hacia el otro.

Microsoft Office proporciona una serie de ventajas en este sentido que lo sitúan como la mayor competencia entre las *suites* de oficina, una de estas características es que Microsoft Word posibilita visualizar dos archivos para compararlos a la vez, así como observar sus diferencias simultáneamente, con lo que las personas ganan tiempo y satisfacción. En tanto LibreOffice Writer no permite revisar textos al mismo tiempo, de manera que favorezca la labor de lectura, minimice los esfuerzos y reduzca el tiempo de revisión de los documentos. Una de las premisas fundamentales para alcanzar éxito en el proceso de migración, es lograr la satisfacción de aquellos que van a interactuar directamente con las aplicaciones que brinda LibreOffice, pues es uno de los *software* que más se utilizan.

Introducción

Una vez planteado lo anterior, se identifica el siguiente **problema científico** ¿Cómo visualizar dos documentos, de forma paralela, en LibreOffice Writer para favorecer la labor de lectura, minimizar los esfuerzos y reducir el tiempo de revisión?

El **objeto de estudio** del presente trabajo de investigación se centra en el proceso de creación de extensiones para LibreOffice.

Por lo que el **objetivo general de la investigación** es desarrollar una extensión para LibreOffice Writer que proporcione la funcionalidad de Vista paralela.

Enmarcándose el **campo de acción** en la creación de extensiones para LibreOffice Writer.

Desglosándose la investigación en los siguientes **objetivos específicos**:

- ✚ Sintetizar el desarrollo de extensiones para LibreOffice centrándose en la Vista paralela.
- ✚ Analizar y diseñar la funcionalidad de Vista paralela para LibreOffice Writer como una extensión.
- ✚ Implementar la funcionalidad de Vista paralela para LibreOffice Writer como una extensión.
- ✚ Probar el funcionamiento de la extensión de Vista paralela para LibreOffice Writer.

Para darle cumplimiento al objetivo propuesto se plantean las siguientes **tareas de investigación**:

- ✚ Revisión de la bibliografía asociada al proceso de creación de extensiones para LibreOffice centrándose en la Vista paralela.
- ✚ Análisis de la herramienta Ver en paralelo de Microsoft Word.
- ✚ Análisis y diseño de la funcionalidad Vista paralela para LibreOffice Writer como una extensión.
- ✚ Implementación de la funcionalidad Vista paralela para LibreOffice como una extensión.

Introducción

✚ Diseño y ejecución de casos de prueba a la extensión desarrollada.

En el marco de esta investigación se plantea como **idea a defender** que el desarrollo de una extensión para el LibreOffice Writer, que proporcione la funcionalidad de Vista paralela, permitirá visualizar dos documentos en pantalla, uno al lado del otro, lo cual favorecerá la labor de lectura, minimizará los esfuerzos y reducirá el tiempo de revisión de los mismos.

Para abarcar todas las características del problema se utilizaron algunos métodos científicos en aras de establecer criterios, definiciones y herramientas necesarios para llegar a la solución del problema en cuestión. Como **métodos teóricos** se aplicaron el Analítico-Sintético, Inductivo-Deductivo y la Modelación. El primero permitió analizar documentos, teorías y otros materiales que posibilitaron la extracción de los elementos más importantes relacionados con el proceso de creación de extensiones para LibreOffice y que hicieron posible la elaboración de conclusiones relacionadas con el objeto de estudio. El segundo permitió inferir a partir de extensiones existentes la estructura y composición de la extensión de Vista paralela. Se empleó la Modelación en la confección de los diagramas para representar la propuesta de solución y a partir de estos la implementación de la herramienta.

Estructura capitular

Para una mejor comprensión del contenido, el presente trabajo de diploma está constituido por tres capítulos:

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice: En este capítulo se ofrecerá una visión general de los elementos fundamentales del proceso de creación de extensiones para la *suite* LibreOffice. Además, se expondrán las herramientas, el lenguaje de programación y la metodología de desarrollo de *software* a utilizar en el desarrollo de la solución.

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer: En este capítulo quedarán expuestos los elementos generados por la metodología, el diseño de la solución propuesta, así como su visión, alcance, objetivos y características.

Introducción

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer: En este capítulo se mostrará el plan de iteraciones que se llevará a cabo para darle cumplimiento a las Historias de Usuario, reflejadas en el capítulo anterior. Se describirán las tareas de ingeniería, además de exponer las pruebas realizadas a la extensión para garantizar su calidad.

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice.

Introducción

A raíz de las solicitudes de los usuarios, surgidas durante el proceso de migración hacia *Software Libre*, en algunos departamentos de la UCI, crece la necesidad de visualizar dos documentos a la vez, por lo que se inicia la presente investigación; para ello los primeros pasos están en función de analizar el conocimiento disponible para generar nuevos conocimientos frente al problema planteado. A continuación se realiza una caracterización de las extensiones, se expone la metodología, técnicas y herramientas a utilizar para alcanzar el objetivo propuesto.

Es necesario plantear que el mayor apoyo de la investigación estuvo centrado en la documentación relacionada con OpenOffice.org (OOo por sus siglas en inglés), ya que LibreOffice es una bifurcación de esta, razón por la cual en determinadas partes del presente documento se hace uso de términos que hacen alusión a dicha *suite*.

1.1 Referentes teórico-metodológicos

Para desarrollar extensiones para LibreOffice es importante conocer definiciones relacionadas con los elementos que componen este proceso, para tener una base teórica de soporte para la investigación en curso.

1.1.1 Definiciones de extensión

Extensión de archivo

Algunos autores plantean que una extensión de archivo o extensión de fichero, es una cadena de caracteres anexada al nombre de un archivo antecedida por un punto. Su función principal es diferenciar el contenido del archivo, de modo que el sistema operativo disponga del procedimiento necesario para ejecutarlo o interpretarlo, sin embargo, la extensión es solamente parte del nombre del archivo y no representa ningún tipo de obligación respecto al contenido del mismo. (1)

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

Extensiones para LibreOffice

Según un escrito publicado en el sitio oficial de LibreOffice, las extensiones son herramientas que se pueden agregar o quitar de forma independiente a partir de la instalación del programa principal. Son *plug-ins* de *software* que se instalan como extras al LibreOffice estándar y que agregan algún tipo de funcionalidad adicional a la *suite*, ya sea para un uso particular (*Writer*, *Calc*, *Impress*), o para todas las aplicaciones o pueden hacer que la funcionalidad existente sea más fácil de usar.

Una forma de referirse a las extensiones es por medio de la palabra complementos, en este caso se denomina así a las que proporcionan funciones adicionales en los navegadores. Cuando se hace alusión a complemento de LibreOffice se refiere a que en LibreOffice se puede observar que la barra Formato cambia tras realizar ciertas operaciones. Por ejemplo, si se inserta una fórmula en un documento de texto, se puede apreciar que los íconos para editar la fórmula, son los mismos que se muestran en los documentos de *Calc*. En este sentido, se suele decir que el documento de *Calc* es un complemento del documento de *Writer*. (2)

Extensiones para OpenOffice.org

Una extensión es un archivo destinado a la distribución de código y / o de datos, que ha de ser utilizado por OpenOffice.org. El archivo tiene la extensión "oxt" (anteriormente .uno.pkg y .Zip), y actúa como un contenedor para varios elementos, como: las bibliotecas, datos de configuración, bibliotecas básicas y diálogos básicos. Puede ser implementada en cualquiera de los lenguajes que soporten la Red de Objetos Universal (UNO por sus siglas en inglés). (3)

De acuerdo a lo expuesto con anterioridad, la autora de la presente investigación define extensión para LibreOffice desde un punto de vista más práctico, como un archivo comprimido en formato Zip que contiene una librería de macros y varios ficheros basados en XML. Al comprimido se le cambia la extensión .zip por .oxt de esta forma puede contener varios elementos, dentro

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

de los cuales se especifican los diferentes parámetros para crear una funcionalidad determinada.

Una vez definido el concepto de extensión se procede a exponer una serie de conceptos y definiciones de suma importancia para la correcta comprensión de la investigación en curso.

1.1.2 Estándar ODF

El formato de documento abierto para aplicaciones ofimáticas conocido como *OpenDocument* u ODF, es un formato de fichero estándar para almacenar documentos ofimáticos. Se aplica a los documentos en soporte electrónico, sean textos, hojas de cálculos, presentaciones o gráficos. Fue diseñado originalmente para OpenOffice.org, pensado como una especificación de formato de documento que hiciese posible el intercambio entre OpenOffice.org y otras aplicaciones compatibles. Tal y como lo describió la *OpenDocument Format Alliance*, es un formato abierto, gratuito, documentado, implementado en XML y mantenido por un organismo internacional. Un documento en formato ODF es un archivo comprimido en ZIP que contiene múltiples ficheros XML con la descripción del documento. (4)

OpenDocument ofrece un formato no-binario, multiplataforma, ampliamente adoptado, modificable y procesable por diversas herramientas. Su desarrollo se ha llevado a cabo en un proceso abierto, público, visible y neutral. Es sometido a un proceso formal de control de cambios y adopción, además está libre de restricciones legales (licencias, patentes). La estandarización asegura un mercado competitivo, la interoperabilidad, la competencia y la innovación. Es un estándar abierto que fue aprobado por *Advancing Open Standards for the Information Society* (OASIS) el 1 de mayo de 2005 y luego por la ISO/IEC JTC1 como un estándar internacional ISO/IEC 26300:2006. (5)

1.1.3 Formato de compresión ZIP

Zip es un concepto que se utiliza en la informática para nombrar a ciertos formatos de almacenamiento, ya sean físicos o virtuales. A nivel de *software*, el zip se utiliza para la compresión de datos (documentos de texto, imágenes, programas) sin pérdida de calidad. El formato (cuya extensión es .zip) es

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

utilizado por diversos programas, tanto comerciales como libres. (6)

Dicho formato de fichero comprime cada uno de los archivos de forma separada, lo que permite recuperar cada uno de ellos sin tener que leer el resto, aumentando el rendimiento. La especificación de ZIP indica que cada archivo puede ser almacenado, o bien sin comprimir, o utilizando una amplia variedad de algoritmos de compresión. Este, soporta un sistema de cifrado simétrico basado en una clave única, además, es muy usado para la creación de archivos con múltiples componentes, estando todos ellos comprimidos en un archivo en formato Zip. El formato *OpenDocument* usado por LibreOffice y OpenXML por Microsoft Office 2007 son archivos XML comprimidos en formato Zip.

1.1.4 UNO

La Red de Objetos Universal es el modelo de componentes utilizado en las *suites* OpenOffice.org y LibreOffice. El objetivo de UNO es proporcionar un entorno para los objetos de red a través de límites de lenguaje de programación y plataforma. Los objetos UNO se comunican por todas partes, ya que este ofrece puentes para enviar llamadas a métodos y recibir valores de retorno entre los procesos y entre los objetos escritos en diferentes lenguajes de programación.

La tecnología de UNO proporciona un marco para las plataformas y lenguajes de programación independientes. Todos los componentes de LibreOffice se pueden desarrollar en cualquier lenguaje que implemente dicha tecnología, mientras que sólo se comunican con otros componentes a través de sus interfaces IDL. LibreOffice provee una interfaz de programación en la forma de UNO, es orientada a objetos y es sub-dividida en varios de ellos por medio de los cuales asegura el acceso controlado por programa al paquete de aplicaciones. (7)

1.1.5 El lenguaje IDL

El lenguaje de definición de interfaz (IDL por sus siglas en inglés) es una manera de describir las interfaces de componentes de *software*, tarea que se realiza de carácter neutral, lo cual permite la comunicación entre ellos

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

independientemente de si están o no desarrollados en diferentes lenguajes, como por ejemplo entre componentes escritos en C++ y otros en *Java*.

IDL permite definir una serie de interfaces para la comunicación entre dos o más aplicaciones. El código se escribe en un fichero, normalmente con la extensión *.idl*, en el cual se definen las interfaces y los métodos. IDL se maneja en entornos donde las interfaces se establecen para la comunicación entre objetos. Por ejemplo, CORBA y OLE tienen sus propios lenguajes de definición de interfaz. UNO no se sostiene detrás de estas tecnologías de componentes y especifica su propio IDL el cual se llama UNO IDL (8)

1.1.6 Gestor de extensiones

Es una herramienta para administrar las extensiones y otros elementos desplegados, como las bibliotecas independientes y los archivos de datos de configuración. Administrar incluye agregar, quitar, habilitar, deshabilitar y actualizar las extensiones de LibreOffice. El Gestor de extensiones se inicia desde dentro de la *suite* pulsando la opción del menú Herramientas>Gestor de extensiones o ejecutando la herramienta *unopkg*, que se encuentra en el directorio del programa de la instalación del paquete ofimático. (9)

1.1.7 Contenedor de Bibliotecas

Un gestor de bibliotecas contiene las bibliotecas de macros. La *suite* LibreOffice presenta dos contenedores, “Macros y diálogos de LibreOffice” y “Mis macros y diálogos”. Puede ver los contenedores de bibliotecas disponibles en la siguiente localización: Herramientas> Macros> Organizar macros> Macros de LibreOffice *Basic*. El contenedor Macros y diálogos de LibreOffice, no debe ser modificado, todas las macros que realice el usuario deben ser almacenadas en Mis macros y diálogos de LibreOffice. (10)

1.1.8 Bibliotecas

Una biblioteca contiene módulos y se utiliza para un alto nivel de agrupación de funcionalidades. Por ejemplo, si se desea escribir un conjunto de macros relacionadas y luego ejecutarlas, entonces se deben guardar todas en la misma biblioteca. No se puede ejecutar una macro a menos que la biblioteca en que

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

está contenida esté cargada. Puede cargar una biblioteca mediante la Interfaz Gráfica de Usuario (GUI por sus siglas en inglés) o desde dentro de una macro. Cada contenedor tiene automáticamente una biblioteca llamada *standard*, esta siempre está cargada para guardar macros específicas y garantizar que estas siempre se puedan ejecutar. (10)

1.1.9 Bibliotecas en tiempo de ejecución

Una biblioteca en tiempo de ejecución es una colección de funciones de utilidad, las cuales soporta un programa mientras se está ejecutando; estas evitan a los programadores el trabajo de reescribir continuamente las capacidades básicas especificadas en un lenguaje de programación o suministradas por el sistema operativo. Las extensiones deben contener sus propias bibliotecas de tiempo de ejecución, por lo general tienen que residir en el mismo directorio que las bibliotecas que contienen los componentes UNO. (8)

1.1.10 Interfaz

En OOo, una interfaz define cero o más métodos y puede heredar de cero o más interfaces. Un objeto que implementa una interfaz debe ser compatible con los métodos definidos por la interfaz y todas las interfaces que hereda. (10)

1.1.11 Servicios

En OOo, un servicio define cero o más propiedades, puede heredar de cero o más servicios, y cero o más interfaces. Un objeto que soporta un servicio debe ser compatible con los métodos y propiedades definidas por el servicio y todas las interfaces y servicios heredados. (10)

1.2 Estructura de las extensiones

Las extensiones deben contener el archivo *description.xml* y un directorio *META-INF* (todo en mayúsculas). El directorio *META-INF* contiene un fichero denominado *manifest.xml* que enumera todos los elementos de la extensión y su tipo de dato.

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

1.2.1 El archivo *description.xml*

El archivo *description.xml* actúa como medio para proporcionar informaciones adicionales, tales como: dependencias, licencias y actualizaciones de la información. El expediente se encuentra en la raíz de la extensión y el nombre distingue entre mayúsculas y minúsculas. Este archivo contiene en su interior la definición de los elementos que se mostrarán al usuario al instalar la extensión en el Gestor de extensiones. (3)

1.2.2 El archivo *manifest.xml*

Si el archivo *description.xml* declara las opciones que se presentarán al usuario al instalar la extensión, *manifest.xml* declara los archivos que definen que nuevos elementos se agregarán al perfil personal en la instalación de OOo. El archivo *manifest.xml* debe estar contenido en una carpeta llamada *META-INF* de forma obligatoria. El Gestor de extensiones busca dicho archivo en esa carpeta, y si no lo encuentra, automáticamente se bloquea o no instala correctamente la extensión. (3)

1.2.3 El archivo *Paths.xcu*

Este archivo se suele situar en la raíz de la carpeta que almacena la extensión, pero realmente puede estar almacenado en cualquier subcarpeta, y no es obligatorio utilizar este nombre, pero sí utilizar la extensión *.xcu*. El nombre de "*Paths*", es porque lo que define este archivo es una modificación a realizar en las carpetas del perfil del usuario. Por ello, lo mejor es dejarlo como está, ya que también es lo más habitual. En caso de cambiar su nombre y localización se debe declarar en el archivo *manifest.xml*. (3)

1.2.4 Versión de una extensión

La versión de una extensión no tiene ninguna influencia en la decisión de si se utiliza la extensión por parte del usuario o de la capa compartida. La utilización de dicho elemento es muy importante, puesto que permite saber la versión actual del producto que se está desarrollando. Además, la identifica cuando en

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

la aplicación ya existe una versión anterior de la misma o para saber si esta ya está instalada. (3)

1.2.5 Licencia de una extensión

Este elemento es importante, pero opcional, el nombre de la carpeta puede ser cualquiera, siendo habitual utilizar una carpeta llamada “*license*”. El nombre de los documentos que contienen las licencias de texto también son de libre elección, lo habitual es utilizar el nombre siguiendo la plantilla *license_xx.txt*, donde *xx* es el código del idioma. En esta sección hay que declarar correctamente cualquier cambio que aparezca. (3)

1.2.6 Nombre para mostrar

Una extensión puede tener un nombre para mostrar, el cual se define en el fichero *description.xml*. Es un dato opcional; si se proporciona, el Gestor de extensiones lo mostrará en el diálogo, en caso contrario se mostrará el nombre del archivo. Dicho fichero puede proporcionar nombres adicionales en diferentes idiomas. El Gestor de extensiones elige el idioma que más se acerque a la configuración regional de la *suite*. (3)

1.2.7 Descripción de una extensión

Se declara el nombre y carpeta donde se encuentran los archivos con la descripción de la extensión. Esta declaración es opcional, pero resulta muy adecuada, ya que se presentará tanto al instalar la extensión como una vez instalada, al hacer clic sobre ella en el Gestor de extensiones. El uso de una descripción es similar al uso de una licencia. (3)

1.2.8 Ícono de una extensión

El Gestor de extensiones muestra un ícono predeterminado para cada extensión, a menos que esta proporcione su propio ícono, el cual se puede almacenar en una subcarpeta cualquiera dentro de la carpeta de la extensión. En ese caso, hay que indicar en el archivo *description.xml* la ruta relativa hasta el ícono, este debe ser una imagen de tipo png o jpeg con un tamaño de 42x42 píxeles. Es opcional, pero incluirlo hace que se muestre en el Gestor de extensiones, lo que le confiere un mejor acabado a la extensión. (3)

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

1.2.9 La carpeta descripción de una extensión

En ella se incluyen uno o más archivos que contienen la descripción de la extensión, que se presenta en el momento de la instalación y luego de hacer clic sobre la entrada en el Gestor de extensiones, cada uno de estos archivos contiene la descripción en un idioma distinto. Es aconsejable que contengan una descripción no muy extensa. Tanto el nombre de la carpeta como el de los archivos es completamente libre, pero para estos se recomienda utilizar la plantilla *description xx.txt*. Sean cuales sean los nombres elegidos, deben estar correctamente declarados en la descripción de la extensión que aparece en el archivo *description.xml*. (3)

1.3 Alcance de una extensión

Los usuarios con privilegios de administrador pueden observar un diálogo donde serán capaces de elegir la opción instalar extensiones "para todos los usuarios" o "solo para mí".

- ✚ Un usuario como administrador o con privilegios de administrador puede instalar una extensión para ser compartida y disponible entre todos los usuarios. Después de seleccionar una extensión, se muestra un diálogo y le solicita el lugar donde se instalará, para el usuario actual o todos los usuarios.
- ✚ Un usuario sin privilegios de administrador solamente podrá instalar extensiones para su propio uso, lo cual se conoce como una extensión de usuario. (3)

1.4 LibreOffice y sus extensiones






LibreOffice es una *suite* ofimática libre, gratuita y multiplataforma, posee un procesador de texto (LibreOffice Writer), un editor de hojas de cálculo (LibreOffice Calc), un creador de presentaciones (LibreOffice Impress), un gestor de bases de datos (LibreOffice Base), un editor de gráficos vectoriales (LibreOffice Draw) y un editor de fórmulas matemáticas (LibreOffice Math). Fue creada por *The Document Foundation* como una bifurcación de la *suite*

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

OpenOffice.org en octubre de 2010 y se ha convertido en la mejor alternativa hoy en día a Microsoft Office. (2)

LibreOffice ofrece la posibilidad de desarrollar extensiones independientes por parte de terceros. Las categorías de extensiones alcanzan desde librerías *Basic*, paquetes con macros *Java/JavaScript* o *Python*, hasta extensiones más sofisticadas en forma de componentes UNO implementados en *Java*, *C++*, *Python*. Incluso, la última categoría de extensiones se subdivide en componentes generales UNO, que proporcionan nuevas implementaciones para áreas funcionales existentes o componentes especializados como: *Add-Ons*, *Add-Ins* de *Calc*, *Add-Ins* de *Chart* o componentes lingüísticos; por ejemplo, comprobador de ortografía, separador de palabras con guiones, o diccionario de sinónimos. (3)

Algunos ejemplos de extensiones de LibreOffice gestionadas por el Gestor de extensiones son:

-  Componentes UNO (módulos de *software* compilados).
-  Datos de configuración (para comandos de menú).
-  Bibliotecas de LibreOffice *Basic*.
-  Bibliotecas de diálogos de LibreOffice.
-  Archivos de extensiones (archivos *.oxt que contengan una extensión).

Aprovechando la particularidad de LibreOffice de poder crear sus extensiones, se desea implementar la solución de forma tal que incluya algunas características propias de una herramienta que ofrece la *suite* Microsoft Office, que resuelven el problema planteado en la presente investigación, para ello se procede a realizar un breve análisis de esta herramienta.

1.5 La herramienta Ver en paralelo de Microsoft Word.

Cada vez se incrementan más las aplicaciones que facilitan la manera de realizar las tareas en las oficinas, las mismas cuentan con una serie de características que logran la preferencia y satisfacción de los usuarios, tal es el caso de la *suite* de oficina Microsoft Office.

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

Una de las aplicaciones de esta *suite* es Microsoft Word, la cual proporciona una herramienta cuyo uso facilita la revisión de documentos. La misma se encuentra bajo el menú Vista y permite acomodar en pantalla dos documentos, estos se pueden situar uno en la parte superior y otro en la parte inferior o en posición vertical, según lo prefiera el usuario. Para utilizar esta opción, se deben abrir los dos archivos que se desean comparar con anterioridad.

Sin duda la herramienta analizada ayuda a incrementar las prestaciones de la aplicación Microsoft Word, es factible y resuelve el problema de visualizar ambos documentos en pantalla, al mismo tiempo, pero no es una funcionalidad reutilizable ya que pertenece a una *suite* privativa, por lo que no se tiene acceso al código para adecuarla a LibreOffice Writer, por esta razón es necesario desarrollar una extensión que adicione al paquete ofimático de las distribuciones libres la siguiente característica :

- Situar dos documentos en pantalla, en posición vertical, uno al lado del otro.

A partir del análisis realizado se empleará la metodología de desarrollo SXP para guiar la realización de la solución.

1.6 Metodología de desarrollo de software SXP

Para definir los procedimientos y técnicas así como proporcionar soporte documental para el desarrollo de la solución propuesta, el departamento al cual pertenece la autora de la presente investigación seleccionó como metodología de desarrollo SXP ya que esta fue desarrollada en la UCI, agrupa las buenas prácticas de XP y SCRUM, ambas metodologías ágiles y sobre todo porque se ajusta a las características del proyecto.

El uso de la metodología SXP contribuye a que los miembros del equipo aumenten su responsabilidad de manera consciente, además, permite que el líder del proyecto tenga un excelente control del mismo. Una ventaja de suma importancia, es que el usuario final es un miembro más del equipo de desarrollo, lo cual representa un éxito para el proyecto.

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

El desarrollo con SXP se realiza en iteraciones cortas (*sprints*) a lo largo de tres fases, dándole cumplimiento a un grupo de actividades, de las que se generan una serie de artefactos que documentan el proceso de desarrollo, obteniendo una liberación del producto con nuevas funcionalidades. (11)

Seguidamente se proponen las tecnologías y herramientas a utilizar en el proceso de creación de la extensión de Vista paralela para LibreOffice Writer, de modo que se contribuya a la conformación del perfil tecnológico para la realización de la aplicación en cuestión.

1.7 Herramientas y tecnologías a utilizar en el desarrollo de la extensión.

Para el desarrollo de la extensión se utilizarán una serie de tecnologías y herramientas, que serán la base fundamental para el correcto diseño e implementación de la solución, a continuación se muestran las mismas, así como el por qué de su selección.

1.7.1 Herramienta de modelado Visual Paradigm 8.0

Para representar los diagramas UML de las diferentes fases por las que transitó el desarrollo de la solución, se decide utilizar la herramienta *Visual Paradigm*, dichas fases son la captura de requisitos, análisis, diseño e implementación. Esta herramienta está disponible en múltiples plataformas y permite realizar el modelado de requisitos, el modelado del proceso de negocio, la interoperabilidad y la generación de documentación. (12)

Se escogió porque, a pesar de que no es libre, es la mejor alternativa para realizar diversas tareas, por ejemplo, modelar los artefactos generados durante el desarrollo de la extensión de Vista paralela para LibreOffice Writer. La misma será utilizada en su versión para *GNU/Linux* con lo cual permitirá dibujar los diagramas de clases, de componentes y de paquetes, suministrando elementos a la documentación, que servirá de apoyo para futuros trabajos relacionados con la propuesta.

1.7.2 API UNO

La API UNO (del inglés *Application Programming Interface of Universal Network Objects*), es una interfaz de programación en la forma de UNO. Está

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

basada en componentes de interfaz y diseñada para ofrecer interoperabilidad entre diferentes lenguajes de programación, modelos de objetos y arquitecturas de *hardware*, ya sea en una computadora, dentro de una red LAN o a través de Internet. (7) La misma es orientada a objetos por lo que puede ser utilizada para crear, abrir, modificar e imprimir documentos de LibreOffice, facilitando la manera de extender el rango personal de la *suite* a través de la implementación de macros y diálogos personalizados.

1.7.3 Extensión BasicAddonBuilder 0.3.5

BasicAddonBuilder es una útil extensión para OpenOffice.org, diseñada para exportar una biblioteca de *StarBasic* en el formato estándar. Esta no requiere de un conocimiento profundo de las especificaciones de extensiones; muestra un cuadro de diálogo asistente que guía el proceso, permitiéndolo definir de manera gráfica todos los menús y barras de herramientas que se agregarán a la interfaz de usuario de OpenOffice.org, a fin de lanzar las macros desde la biblioteca de *StarBasic*. (13)

Se utilizará dicha herramienta para el desarrollo de la propuesta, ya que facilita la creación de extensiones siguiendo el asistente que las genera. Como *BasicAddonBuilder* 0.3.5 es en sí misma una extensión de OpenOffice.org, se puede emplear en LibreOffice, una vez instalada agregará una nueva barra de herramientas con un único botón que al pulsarlo desplegará el asistente, el cual conducirá el proceso completo de creación de la extensión de Vista paralela.

1.7.4 Entorno de Desarrollo Integrado

El entorno de desarrollo integrado (IDE por sus siglas en inglés) es la herramienta que se utiliza para el desarrollo de aplicaciones, este integra la edición, depuración, diseño de la interfaz gráfica, la ayuda en línea y las características avanzadas, tales como: el control de versiones y la gestión de proyectos en una interfaz de usuario unificada. Para la implementación de la extensión de Vista paralela se decide emplear el editor de código *Basic* que incorpora la *suite* LibreOffice, su nombre es el acrónimo del inglés *Basic Integrated Debugging Environment* (Entorno Integrado de Depuración); el mismo permite escribir, ejecutar y depurar macros. (14)

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

Se decide utilizar dicho IDE porque posibilita a los desarrolladores construir fácilmente LibreOffice, puesto que comprende las funciones más requeridas, incluyendo módulos de construcción, la creación de componentes para la *suite*, así como las pruebas de funcionamiento de la misma, por lo que a continuación se plantea el lenguaje de programación compatible con dicho entorno de desarrollo.

1.7.5 Lenguaje de programación LibreOffice *Basic*

LibreOffice *Basic* es el lenguaje de programación desarrollado e integrado directamente en LibreOffice. Actualmente ofrece la mejor integración con esta, porque se puede insertar código en documentos, adjuntar eventos arbitrarios de la *suite*, tales como: cargar documentos, atajos de teclado o entradas de menú. En LibreOffice *Basic*, las llamadas se invocan en un objeto en lugar de en una interfaz específica. Las Interfaces se integran como propiedades de los objetos.

Dicho lenguaje de programación siempre se ejecuta durante los procesos de la *suite* y por lo tanto evita la ocurrencia de llamadas entre procesos costosos; es inseguro, debido a que sólo un número mínimo de errores se encuentran durante la compilación. La mayoría de los errores aparecen en tiempo de ejecución, por lo que no es la mejor opción para grandes proyectos. El lenguaje es específico de LibreOffice y sólo ofrece un pequeño conjunto de funcionalidades en tiempo de ejecución, con poco apoyo de otros lenguajes. (15)

Estructuración del lenguaje

Subrutinas

Las subrutinas se utilizan para manejar tareas concretas dentro de un programa, para identificarlas se emplea la contracción Sub. Estas dividen una tarea en procedimientos individuales; dividir un programa en procedimientos y subprocedimientos mejora su legibilidad y reduce la posibilidad de errores. Una subrutina puede tomar algunos argumentos como parámetros, pero no devuelve ningún valor a la subrutina o función que la ha llamado. (13)

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

Función

Una función es esencialmente una subrutina que devuelve un valor. Las funciones se pueden usar en el lado derecho de una declaración de variable o en otros sitios en que normalmente se usarían valores. Después de separar el programa en procedimientos y funciones, estas se pueden guardar como archivos, para reutilizarlas en otros proyectos.

LibreOffice *Basic* admite módulos y bibliotecas. Tanto las subrutinas como las funciones siempre se incluyen en los módulos, estos pueden definirse para que sean globales o formen parte de un documento. Varios módulos pueden combinarse en una biblioteca. (13)

El éxito de la extensión depende en gran medida del nivel de organización que se tenga durante el período de creación de la misma, para ello es necesario emplear algunas herramientas que faciliten manejar las actualizaciones generadas en cada iteración, seguidamente se explican las que se seleccionaron para el desarrollo de la propuesta.

1.7.6 El servidor Subversion 1.16.17 y la herramienta de desarrollo colaborativo RapidSVN 0.12.0

Para mantener el control de las versiones de la solución se hará uso del servidor *Subversion* y su plataforma visual *RapidSVN*, ya que así fue definido por el departamento SIMAYS, al cual pertenece la autora de la presente investigación.

Subversion 1.16.17

Subversion es un sistema de control de versiones, se le conoce también como SVN por ser el nombre de la herramienta utilizada en la línea de commando. (16)

Algunas de sus principales características son:

- ✚ Permite administrar cualquier tipo de ficheros.
- ✚ Las diferentes versiones de un mismo archivo no tienen un número de revisión independiente.

Capítulo 1: Fundamentación teórica del proceso de creación de extensiones para LibreOffice

- ✚ Verifica las versiones de los proyectos almacenados.
- ✚ Mantiene las versiones de los directorios y archivos.

RapidSVN 0.12.0

RapidSVN es una plataforma visual para el sistema *Subversion* escrito en el lenguaje C++, con Código Abierto y *Software Libre* bajo la licencia *GNU General Public License (GPL)*. Está disponible en varios idiomas y actúa de cliente gráfico para el acceso al repositorio SVN, tanto si este es remoto como si es local. Es de fácil manejo para los usuarios principiantes, pero lo suficientemente potente y con herramientas interesantes para los usuarios avanzados. El programa se puede ejecutar en *Linux, Windows, Mac OS / X, Solaris*, etc. (17)

Se utilizarán ambas aplicaciones, lo cual permitirá explotar sus características, en función de garantizar la seguridad de las actualizaciones de los documentos relacionados con la presente investigación. La primera de ellas es muy utilizada en la actualidad por los beneficios que aporta, en tanto la segunda posee la particularidad de ser una de las plataformas más completas con las que se cuenta hoy en día.

Conclusiones del capítulo

Para llevar a cabo el desarrollo de la propuesta, se concreta el perfil tecnológico a partir de las valoraciones realizadas de la metodología, tecnologías y herramientas a utilizar. El mismo es el punto de partida para realizar un correcto análisis y diseño de la extensión que proporcione la funcionalidad de Vista paralela para LibreOffice Writer, de modo tal que los usuarios de la *suite* encuentren en esta todo lo que necesitan.

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Introducción

Una vez definida la metodología, lenguaje, herramientas y tecnologías a emplear para el desarrollo del *plug-in*, se procede a realizar el análisis y diseño de la solución, para ello se definirán los requisitos funcionales y no funcionales en la Lista de Reserva del Producto (LRP). En las HU se reflejará la especificación de los requisitos definidos para la extensión, así como las funcionalidades a implementar; se definirá la arquitectura a utilizar y se realizará el modelo de diseño del *plug-in*, donde se elaborarán los diagramas, de componentes y de paquetes.

2.1 Propuesta de solución

La extensión contará con funcionalidades que en conjunto, permitirán abrir dos documentos para revisarlos a la vez; situándolos en posición vertical, uno al lado del otro. Para ejecutar la extensión se debe acceder al ícono que la representa, el mismo aparece en la barra de menu. Una vez realizado este proceso el usuario tendrá la posibilidad de seleccionar en un diálogo, cual de los documentos desea ver en paralelo con el documento desde donde ejecutó la extensión, o podrá abrir un documento desde un directorio determinado. De esta forma la herramienta favorecerá la labor de lectura, minimizará los esfuerzos y reducirá el tiempo de revisión de los textos, por lo que brindará mayor comodidad al usuario.

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

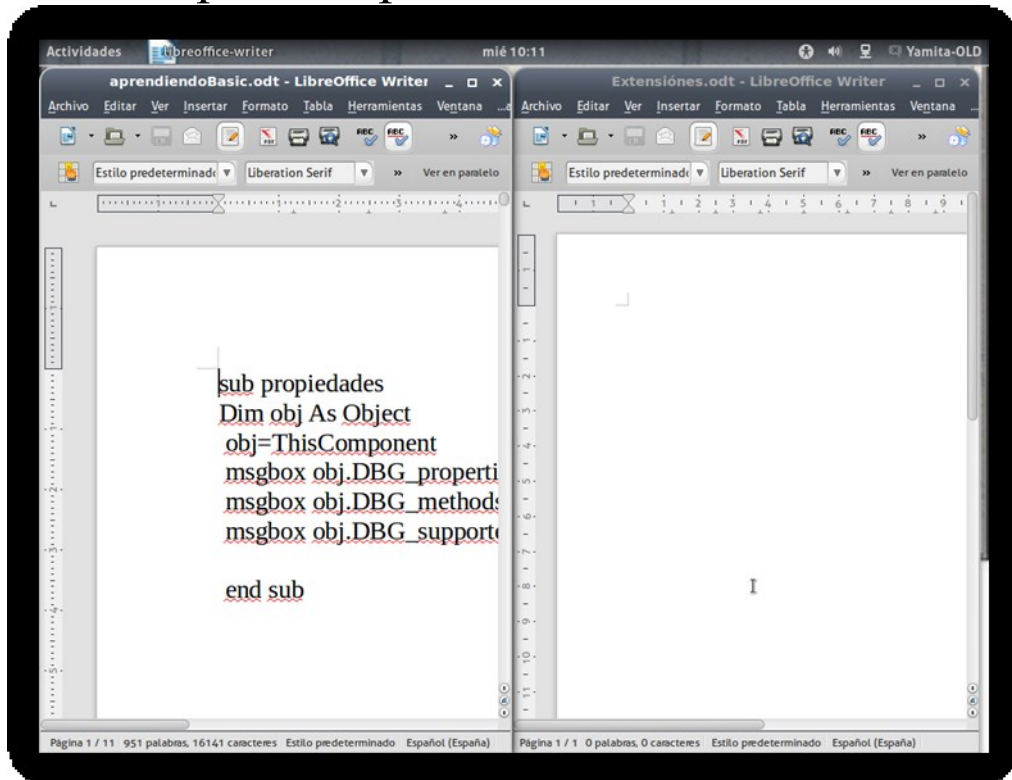


Figura 1: Documentos en paralelo al ejecutar la extensión

2.2 Requisitos Funcionales y No Funcionales del producto

Después de analizar el problema, es preciso definir qué es lo que debe hacer la extensión. Para ello, deben ser estudiadas todas las ideas que los usuarios y miembros del equipo de proyecto tengan sobre lo que debe hacer el *plug-in*, como candidatas a requisitos. Los requisitos son las condiciones o capacidades que tienen que ser alcanzadas por la solución para satisfacer las necesidades del cliente.

La LRP es una recopilación organizada de Requisitos Funcionales y No Funcionales del producto, se centra en especificar las características que debe tener el *software* y define el orden en que se les irá dando cumplimiento a cada uno de los requerimientos recogidos, de acuerdo a la prioridad establecida para estos. A continuación se muestra cada requisito funcional y no funcional identificado, la prioridad con la que se les dará solución, entre otras informaciones pertinentes.

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Tabla 1: Requisitos Funcionales (RF)

Asignado a	Ítem *	Descripción	Estimación	Estimado por
Prioridad		Alta		
Yamilka Leyva Coss	1	Activar Vista paralela	2	Programador
Yamilka Leyva Coss	2	Mostrar documentos abiertos	1	Programador
Yamilka Leyva Coss	3	Seleccionar un documento abierto	1	
Prioridad		Media		
Yamilka Leyva Coss	4	Mostrar barra de herramienta	1	Programador
Yamilka Leyva Coss	5	Abrir documento desde un directorio	2	Programador

Tabla 2: Requisitos No Funcionales (RNF)

Asignado a	Ítem*	Descripción	Estimación	Estimado por
Yamilka Leyva Coss	1	Apariencia (La interfaz para acceder a la funcionalidad de la extensión debe ser sencilla, para que los usuarios puedan utilizarla fácilmente.)	0,2	Programador

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Yamilka Leyva Coss	2	Implementación (La extensión deberá ser implementada usando los lenguajes que soporten UNO.)	0,2	Programador
Yamilka Leyva Coss	3	Legales (La plataforma a utilizar para el desarrollo de la aplicación debe estar basada en la licencia GNU/GPL.)	0,2	Programador

2.3 Historias de Usuario

Para describir los requisitos de la extensión y priorizar las necesidades del cliente, se realizarán las Historias de Usuarios (HU), que servirán para sentar las bases para las pruebas funcionales. Se elaboraron tres HU, en correspondencia con los Requisitos Funcionales expuestos anteriormente. Estas permitirán especificar el modo en que se desarrollará la funcionalidad de Vista paralela.

Tabla 3: HU Gestionar documentos.

Historia de Usuario	
Número: HU01	Nombre Historia de Usuario: Gestionar documentos.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Yamilka Leyva Coss	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 4 semanas
Riesgo en Desarrollo: Medio	Puntos Reales: 3 semanas

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

<p>Descripción: La HU muestra un diálogo, con tantos botones de opción como documentos de LibreOffice haya abiertos, cuenta además con tres botones de control: Aceptar, Buscar y Cancelar; los cuales permiten realizar las acciones indicadas.</p>
<p>Observaciones:</p> <p>Mostrar los documentos abiertos: Debe mostrar los documentos de LibreOffice Writer, que el usuario tenga abiertos en el momento de ejecutar la extensión.</p> <p>Seleccionar un documento abierto: Debe permitir seleccionar uno de los documentos que se muestran en el diálogo.</p> <p>Abrir documento desde un directorio: Debe permitir abrir un documento desde un directorio determinado por el usuario.</p>
<p>Prototipo de interfaz:</p> <p>(Ver Anexo 4)</p>

Tabla 4: HU Mostrar barra de herramienta

Historia de Usuario	
Número: HU02	Nombre Historia de Usuario: Mostrar barra de herramienta.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Yamilka Leyva Coss	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 1 semana
Riesgo en Desarrollo: Alta	Puntos Reales: 1 semana
Descripción: La HU permite mostrar una nueva barra de herramienta con la opción Ver en paralelo.	
Observaciones:	
Mostrar Barra de herramienta debe permitir el acceso a la opción antes descrita, para activarla a través del ícono de la extensión que aparecerá como una barra de herramienta adicional.	
Prototipo de interfaz:	
(Ver Anexo 5)	

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Tabla 5: HU Activar Ver en paralelo

Historia de Usuario	
Número: HU03	Nombre Historia de Usuario: Activar Ver en paralelo
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Yamilka Leyva Coss	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 2 semanas
Riesgo en Desarrollo: Alta	Puntos Reales: 2 semanas
Descripción: La HU permite activar la opción Ver en paralelo.	
Observaciones: El usuario puede activar la opción Ver en paralelo. Para ello accede al menú de herramientas y presiona el ícono Ver en paralelo. Se debe mostrar el documento seleccionado y el documento desde donde ejecutó la extensión, en posición vertical, uno al lado del otro.	
Prototipo de interfaz: (Ver Anexo 6)	

2.4 Implementación de los requisitos

2.4.1 Tareas de ingeniería (TI)

Tabla 6: TI 1 Implementar la macro para mostrar los documentos abiertos

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 01
Nombre Tarea: Implementar la macro para mostrar los documentos abiertos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 17/03/2014	Fecha Fin: 21/03/2014

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Programador Responsable: Yamilka Leyva Coss
Descripción: Se realiza la implementación de la macro que permitirá mostrar los documentos de LibreOffice que el usuario tiene abiertos

Tabla 7: TI 2 Implementar la macro para redimensionar los documentos

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 03
Nombre Tarea: Implementar la macro para redimensionar los documentos	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 semana
Fecha Inicio: 31/03/2014	Fecha Fin: 4/04/2014
Programador Responsable: Yamilka Leyva Coss	
Descripción: Se realiza la implementación de la macro que permitirá redimensionar los documentos de LibreOffice para adaptarlos a las necesidades del usuario.	

Tabla 8: TI 3 Implementar la macro para situar los documentos en paralelo

Tarea de Ingeniería	
Número Tarea: 3	Número Historia de Usuario: 03
Nombre Tarea: Implementar la macro para situar los documentos en paralelo	
Tipo de Tarea : Desarrollo	Puntos Estimados: 2 semana
Fecha Inicio: 31/03/2014	Fecha Fin: 4/04/2014
Programador Responsable: Yamilka Leyva Coss	
Descripción: Se realiza la implementación de la macro que permitirá situar en pantalla dos documentos, uno al lado del otro.	

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Tabla 9: TI 4 Diseñar el icono de la extensión

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 03
Nombre Tarea: Diseñar el icono de la extensión	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 23/04/2014	Fecha Fin: 25/04/2014
Programador Responsable: Yamilka Leyva Coss	
Descripción: Se realiza el diseño del ícono que mostrará la extensión, una vez instalada.	

Tabla 10: TI 5 Empaquetar las macros creadas

Tarea de Ingeniería	
Número Tarea: 5	Número Historia de Usuario: 01
Nombre Tarea: Empaquetar las macros creadas	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio: 23/04/2014	Fecha Fin: 25/04/2014
Programador Responsable: Yamilka Leyva Coss	
Descripción: Se crea la extensión, agrupando para ello, las funcionalidades implementadas, el ícono y la licencia; una vez creada se procederá a su instalación	

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

2.5 Conclusiones de la fase de Análisis

En la fase de análisis se dio prioridad al conocimiento de los requerimientos, los conceptos y las operaciones relacionadas con la propuesta. La investigación y el análisis se caracterizaron por centrarse en cuestiones concernientes al proceso de creación de extensiones para LibreOffice, en particular para la aplicación LibreOffice Writer. Durante el desarrollo iterativo se procede a pasar a la fase de diseño, de modo tal que se logre una solución lógica que se funda en el paradigma orientado a objetos. Su esencia es la elaboración de diagramas, que muestran gráficamente como los objetos se comunicarán entre sí a fin de lograr el cumplimiento de los requerimientos.

2.6. Arquitectura definida para la aplicación

En el proceso de desarrollo de una aplicación se emplea una arquitectura determinada, de modo que su utilización contribuya a dar una mejor solución, producto de las ventajas que esta proporciona. En el desarrollo de la extensión de Vista paralela para LibreOffice Writer se utiliza la Arquitectura en Capas (N-Capas o N-Layer).

Se emplea el estilo arquitectónico en dos capas, de manera que permita dividir el problema de forma efectiva, utilizando para ello el diseño de datos a nivel de componentes, este se concentra en la representación de las estructuras de datos a las que se tiene acceso de forma directa, mediante uno o más componentes de *software*. La primera es la capa de Presentación o Interfaz gráfica, con la cual interactúa directamente el usuario, en este caso en particular se trata del ícono de acceso a la extensión, una vez ejecutado el *plug-in*, se genera un diálogo para la captura de información. La segunda capa es la de Lógica de Negocio, esta recibe las peticiones del usuario, ejecuta las acciones y establece la forma en que estas se van a desarrollar, tal es el caso de situar los dos textos en pantalla, uno al lado del otro.

2.7 Modelo de implementación

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

“El modelo de implementación describe como los elementos del modelo de diseño se implementan en términos de componentes como las clases, ficheros de código fuente, ejecutables, entre otros. El modelo de implementación describe también como se organizan los componentes, de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados y como dependen los componentes unos de otros.” (18)

Para el desarrollo de la solución no fue necesario diseñar un diagrama de clases específico, solo se utilizaron servicios e interfaces definidos. El trabajo con la API se hizo a través de la creación de objetos para el acceso a cada uno de los elementos disponibles. Toda la implementación se realizó en un módulo, el cual está compuesto por una serie de macros, las cuales definen las diferentes funcionalidades de la extensión. Los módulos se guardan y organizan en bibliotecas, estas están contenidas dentro de documentos. (Ver Anexo 7)

Todos los documentos tienen una biblioteca especial llamada *Standard*, a esta se le pueden agregar y eliminar módulos, pero no puede ser eliminada; es importante mencionar que también pueden contener diálogos, tal es el caso de la solución que se propone, dicho diálogo será explicado más adelante.

2.7.1 Diagrama de componentes del diseño

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre los componentes de *software*, sean estos de código fuente, binarios o ejecutables. A partir de este diagrama se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión de *software*, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

Para dar soporte a las funcionalidades de la extensión se utilizó la librería de ejecución de LibreOffice *Basic*, esta provee funciones estándares que no tienen referencia con LibreOffice. Se emplearon además, una serie de servicios e

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

interfaces del módulo *Star*, de los cuales a continuación se muestra una selección, así como una breve explicación de los que más influyeron en el desarrollo de la solución propuesta.

Para ello se elaboró el diagrama de los componentes que intervienen durante el proceso de creación de la extensión de Vista paralela para LibreOffice Writer, el mismo muestra las opciones de realización, incluyendo Código fuente y ejecutable; fue necesario dividir en dos partes la ilustración que muestra dicho diagrama de componentes, para su correcta visualización en el presente documento. Algunos de estos componentes existen en tiempo de compilación, algunos en tiempo de ejecución y otros en ambos casos. (Ver el Anexo 8)

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

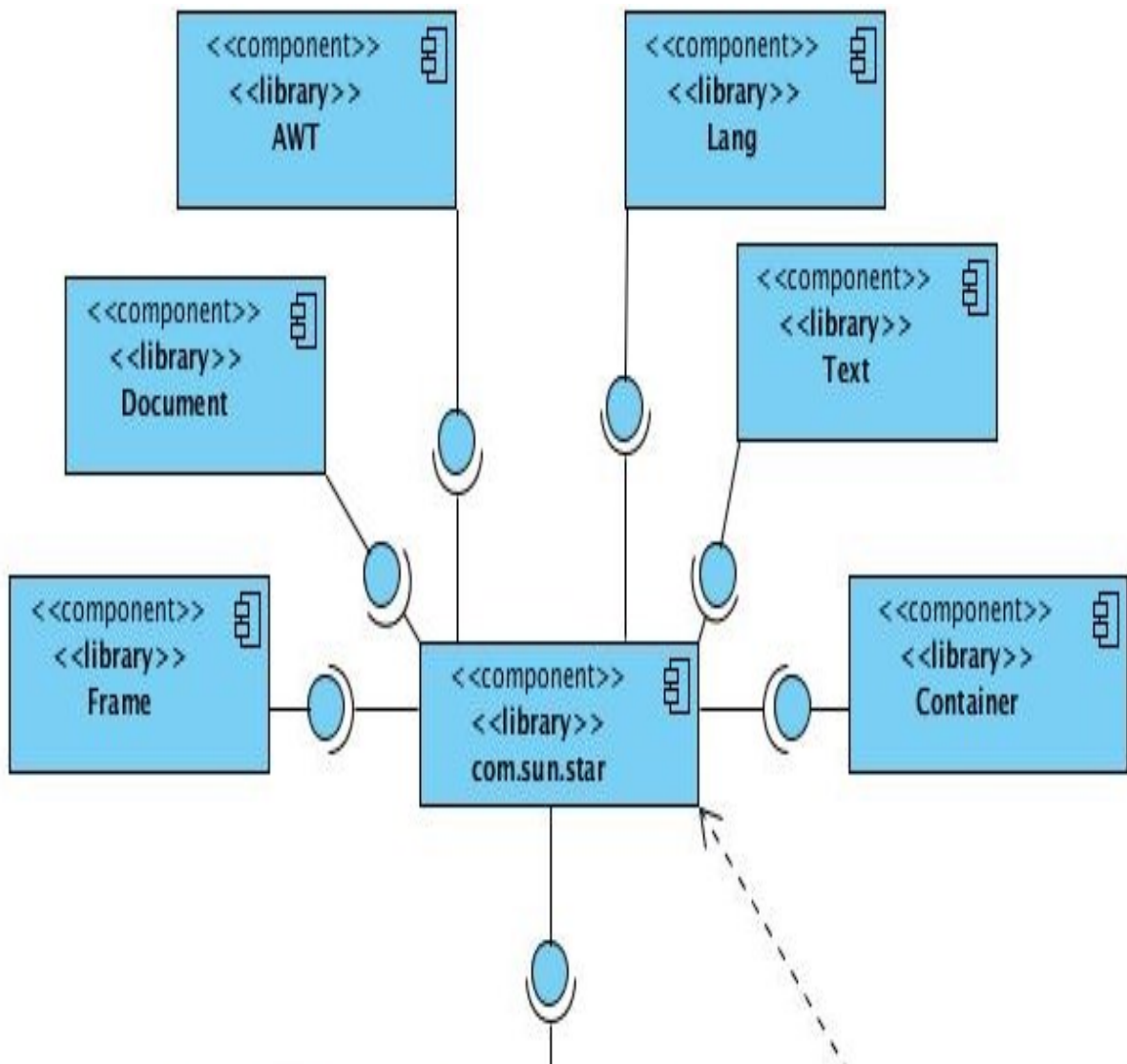


Figura 2: Diagrama de componentes del diseño parte 1

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

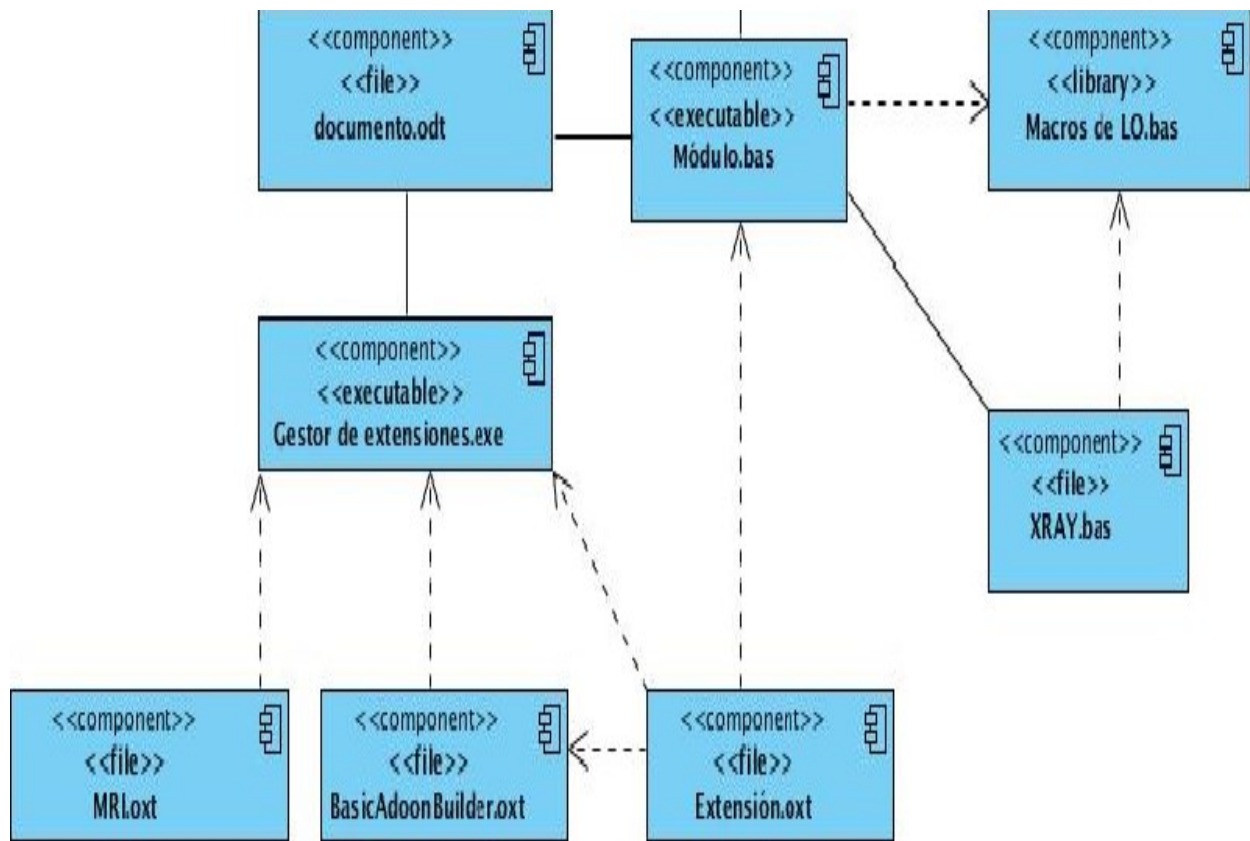


Figura 3: Diagrama de componentes del diseño parte 2

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

2.7.2 Diagrama de paquetes del diseño

Para la representación de la extensión basada en la organización de paquetes y sus elementos, se diseñó el diagrama de paquetes tomando como marco de referencia el estilo arquitectónico propuesto. Este especifica la forma en que está dividido el proceso de creación de la extensión en agrupaciones lógicas, mostrando las relaciones entre ellas, de este modo se brinda una descomposición de la jerarquía lógica de la aplicación. El diagrama de paquetes se muestra en dos partes para su correcta visualización en el presente documento. (Ver Anexo 9)

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

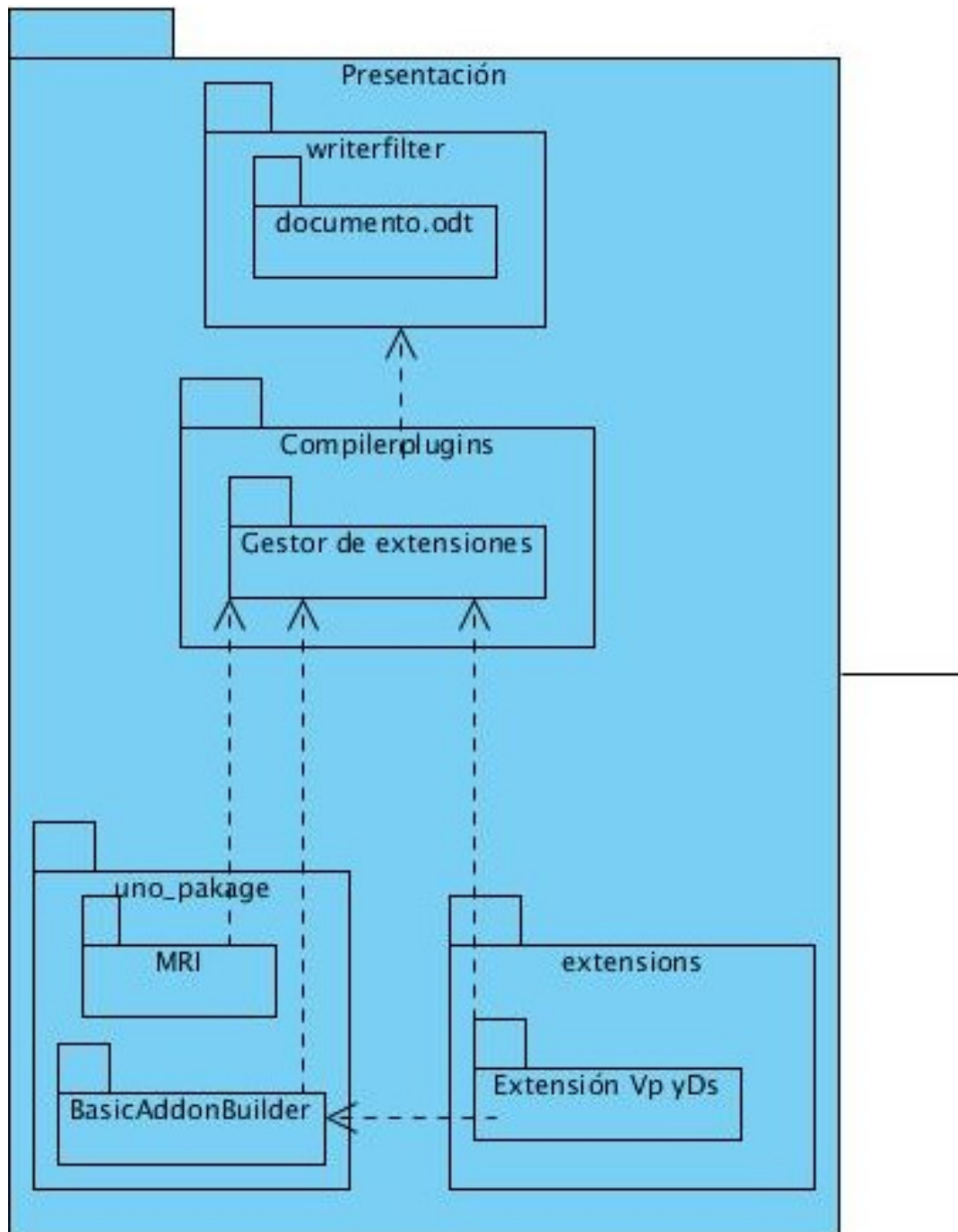


Figura 4: Diagrama de paquetes del diseño parte 1

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

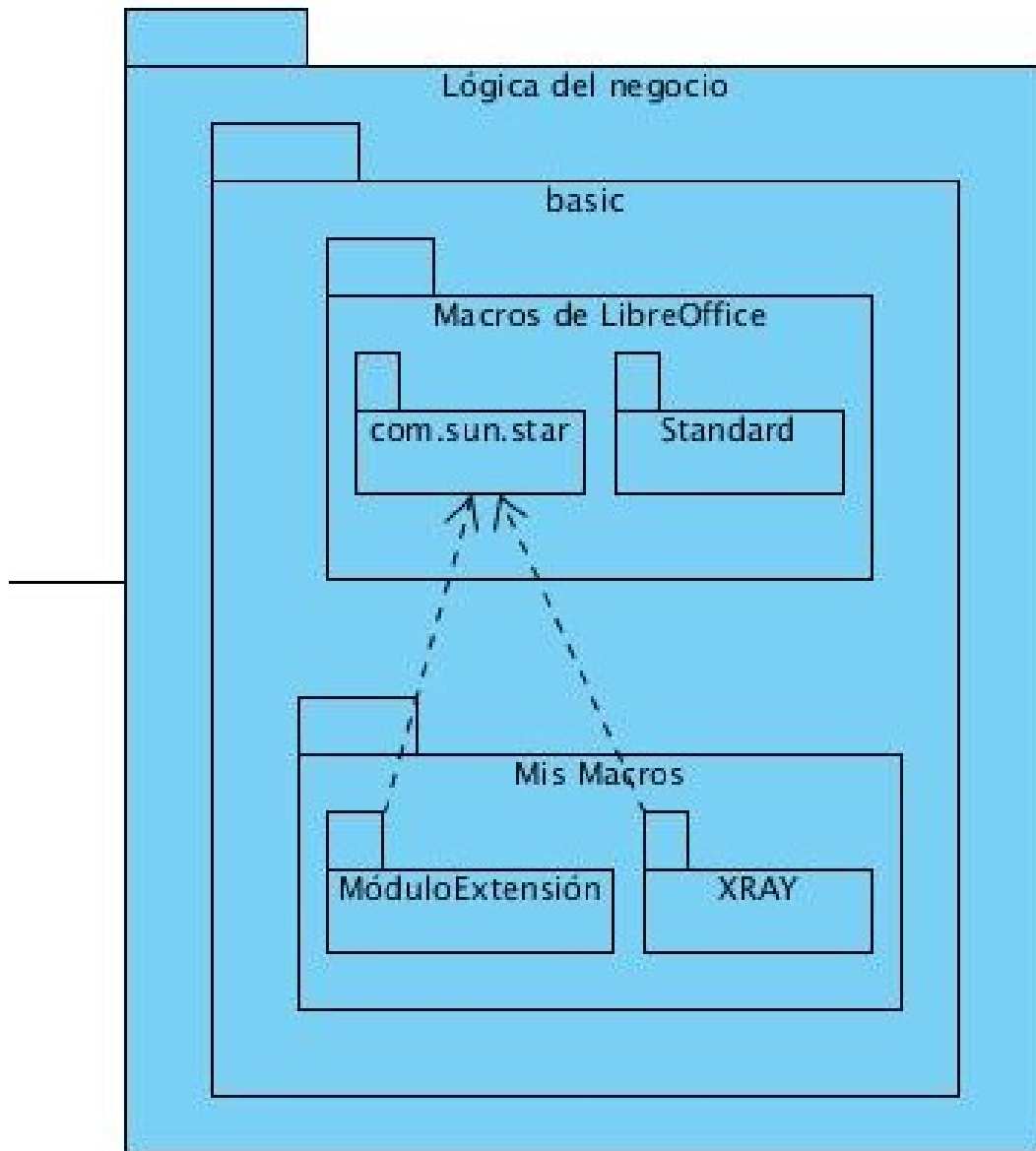


Figura 5: Diagrama de paquetes del diseño parte 2

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Los módulos más usados del servicio *Star* (19) para la implementación de las funcionalidades son:

- ✚ *AWT*: Proporciona interfaces para el trabajo con herramientas y objetos en cualquiera de los lenguajes que soportan UNO. (Ver Anexo 10, Anexo 11)
- ✚ *Container*: Proporciona las interfaces para colecciones y contenedores.(Ver Anexo 12, Anexo 13)
- ✚ *Document*: Proporciona las interfaces relacionadas con los documentos del paquete *Office*. (Ver Anexo 14, Anexo 15)
- ✚ *Frame*: Provee interfaces relacionadas con el entorno de escritorio.(Ver Anexo 16, Anexo 17)
- ✚ *Lang*: Proporciona conceptos generales del trabajo con el modelo de componentes UNO.(Ver Anexo 18, Anexo 19)
- ✚ *Text*: Especifica las interfaces para el trabajo con textos.(Ver Anexo 20, Anexo 21)
- ✚ *Uno*: Provee las interfaces básicas para el trabajo con objetos.(Ver Anexo 22, Anexo 23)

Para mayor comprensión de los módulos antes expuestos ver los anexos que se indican en cada caso. A continuación se procede a explicar determinados métodos, pertenecientes a algunas interfaces que no aparecen explicadas en los anexos y que tuvieron relevancia a la hora de implementar el *plug-in*.

En algunas instancias, un objeto puede contener una lista de objetos subordinados que no pueden ser accedidos por un nombre o por un índice, por ejemplo los documentos que el usuario tiene abiertos. En estas situaciones son apropiadas las interfaces *XEnumeration* y *XEnumerationAccess*, estas proveen un mecanismo por medio del cual todos los elementos subordinados de un

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

objeto pueden ser pasados, paso a paso, sin tener que usar un direccionamiento directo. El acceso a la interfaz *XenumerationAccess* se realiza a través de un objeto básico, dicha interface contiene el método *CreateEnumeration* que crea una enumeración con los datos almacenados en el objeto básico. Esto retorna un objeto auxiliar, el cual proporciona la interfaz *XEnumeration* con los métodos *hasMoreElements* y *NextElement*. Por medio de estos, se tendrá acceso a los objetos subordinados para trabajar con cada uno de ellos.

Para el desarrollo de la solución fue necesario emplear la función *CreateUnoService*, esta crea un objeto que puede ser usado universalmente, dicho objeto también es llamado servicio independiente del contexto. Adicionalmente a los servicios independientes del contexto están los servicios dependientes del contexto, estos objetos solamente son útiles cuando son usados conjuntamente con otro objeto. El método *createInstance*, el cual está definido en la interfaz *XmultiServiceFactory*, es usado en particular en estos objetos de documentos.

El servicio *com.sun.star.frame.Desktop* provee las funciones para el módulo *Frame* de LibreOffice, dentro del cual están clasificadas todas las ventanas de documentos, los cuales pueden ser creados, abiertos e importados usando este servicio, el mismo abre automáticamente cuando LibreOffice es iniciada. Para lograr esto, se crea un objeto con el nombre global de *StarDesktop*. La interfaz más importante que soporta el *StarDesktop* es *com.sun.star.frame.XcomponentLoader*. El módulo *Xcomponent* contiene el método *loadComponentFromURL*, el cual es el responsable de la creación, importación y apertura de documentos. (15)

Para el desarrollo de la funcionalidad que muestra los documentos abiertos, se empleó un diálogo; estos de manera general consisten en una ventana de diálogo que puede contener campos de texto, cajas de listas, botones de radio y otros elementos de control. El diálogo de la propuesta contiene botones de radio, los cuales serán visibles en tiempo de ejecución de acuerdo a la cantidad de documentos que tenga abiertos el usuario, debido a que no se tiene certeza

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

de la cantidad exacta de documentos, que un usuario determinado pueda tener abiertos, en el momento de ejecutar la extensión. Se empleó el método *CreateUnoDialog*, este crea un objeto que está referenciado al diálogo asociado, para ello es preciso construir el diálogo antes de usar dicho método.

Algunos objetos soportan el método *supportsService*, el cual puede establecer cuando un objeto soporta un servicio particular. Además de los elementos anteriormente expuestos, se empleó una macro de OpenOffice.org, esta es denominada *XRAY* en su versión 6.0, la cual fue sumamente importante para la implementación de las funcionalidades de la extensión, ya que el IDE de depuración de macros no tiene auto-realización, como es usual en otros entornos de desarrollo.

Esto se debe al diferente concepto de diseño de LibreOffice de no tener objetos predefinidos, todos los objetos necesarios se crean a través de instancias en el momento que sea preciso, por lo que no es posible conocer de forma estática (en tiempo de compilación) las propiedades y métodos disponibles de una variable de objeto de tipo UNO.

Las propiedades, métodos y servicios sólo se pueden encontrar en tiempo de ejecución, por lo tanto, fue necesario llamar a la macro *XRAY* 6.0, con el objeto a ser analizado como argumento, como en el siguiente ejemplo:

```
GlobalScope.BasicLibraries.loadLibrary("XrayTool")
```

```
xray docActual
```

La variable *docActual* hace referencia a un objeto determinado. La ejecución de dicha herramienta devuelve en un diálogo los elementos antes mencionados. (Ver Anexo 24)

Con un propósito similar se utilizó la extensión Resonancia Magnética, en su versión 1.2.1 (MRI, por sus siglas en inglés), esta cuenta con un amplio sistema de contribución que se integra con la ayuda en línea de OpenOffice.org durante su instalación. (Ver Anexo 25)

Capítulo 2: Análisis y diseño de la extensión de Vista paralela para LibreOffice Writer

Conclusiones del capítulo

Se identificaron cinco Requisitos Funcionales y tres No Funcionales, con lo cual se distribuye la implementación de las funcionalidades de la extensión en tres historias de usuarios. El uso de la arquitectura permitió dividir el problema de forma efectiva en dos capas. Con la representación de los elementos que componen el Modelo de Diseño, se obtuvo la correcta composición de un Modelo de implementación que contribuyó al desarrollo del producto.

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

Introducción

Realizados el análisis y diseño de la solución, se procede a implementar la extensión de Vista paralela, teniendo como punto de partida los diagramas destinados al ciclo de desarrollo actual de la solución; para ello es necesario elaborar la descripción del plan de iteraciones, las tareas de ingeniería y el estándar de codificación a utilizar en la implementación de la solución propuesta. De forma adicional quedarán expuestos los casos de prueba, enfocados en la verificación del correcto desarrollo de las funcionalidades que provee el *plug-in*.

3.1 Planificación de la implementación

La planificación de las etapas, en las cuales se desarrollará cada requisito funcional, será de acuerdo a la prioridad de los mismos, por esta razón se define en que iteración se dará cumplimiento a cada HU, lo cual queda expuesto en el plan de iteraciones. La implementación de la extensión se realizará en dos iteraciones, como se muestra a continuación:

Iteración 1.

En esta iteración se desarrollarán requisitos de prioridad Alta para la investigación, tal es el caso de las HU 01 y 03 ya que constituyen la base fundamental para la extensión, los mismos le brindarán un alto nivel de aceptación por parte del usuario, ya que le permite a este visualizar los documentos de LibreOffice Writer que están abiertos, así como seleccionar entre estos últimos cual desea ver en paralelo con el documento actual (desde donde ejecuta la extensión) o podrá buscar un documento desde un directorio específico.

Iteración 2.

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

En esta última parte del desarrollo se implementará el requisito de prioridad Media, este está recogido en la HU 02, el mismo permite mostrar el ícono de acceso a la funcionalidad Vista paralela. Una vez terminada esta iteración se tendrá la solución lista para ser instalada, para ello se hace uso del Gestor de extensiones que ofrece la *suite* LibreOffice.

3.2 Estándar de codificación

Se decide utilizar el estándar de codificación de caracteres universal (UNICODE, por sus siglas en inglés) de manera que cuando sea necesario consultar la implementación de la extensión, reutilizarla o modificarla todos los programadores del proyecto Personalización Ofimática comprendan el código mediante el cual se desarrolló la misma, contribuyendo al empleo de una de las buenas prácticas propuesta por la metodología escogida. De esta forma se garantiza la legibilidad del código, así como su rápida comprensión.

Siguiendo las convenciones de codificación que propone dicho estándar se pueden obtener las siguientes ventajas:

- ✚ El código tendrá un aspecto coherente, de modo que los lectores pueden centrar mejor su atención en el contenido y no en el diseño.
- ✚ Los lectores entenderán el código con más rapidez porque podrán hacer suposiciones basadas en su experiencia previa.
- ✚ Se podrá copiar, cambiar y mantener el código con mayor facilidad.

Convenciones de comentarios

- ✚ Se deben poner los comentarios en una línea independiente, no al final de una línea de código.
- ✚ Se debe comenzar el texto del comentario con una letra en mayúscula y finalizar con un punto.
- ✚ Se debe insertar un espacio entre el delimitador de comentario (') o la palabra clave REM y el texto del comentario.

Convenciones del lenguaje

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

- ✚ La estructura básica para una macro es

```
Sub nombre-de-la-macro
```

```
    REM Aquí se pone todo el código que se desee
```

```
End Sub
```

- ✚ La palabra clave `Dim` se utiliza para declarar las variables.

```
Dim variable1 As Integer
```

- ✚ Para concatenar cadenas, se debe utilizar una “y” comercial (&).

```
MsgBox("hola" & vbCrLf & "hello")
```

- ✚ Es conveniente usar una sintaxis breve al inicializar matrices en la línea de declaración. Por ejemplo, se puede usar la siguiente:

```
Dim letras1 As String() = {"a", "b", "c"}
```

- ✚ Se debe utilizar la sintaxis `{}` al declarar e inicializar matrices de tipos de datos básicos. Por ejemplo, se puede usar la siguiente sintaxis:

```
Dim letras2() As String = {"a", "b", "c"}
```

- ✚ Cuando se realicen una serie de llamadas a un objeto, se debe considerar la posibilidad de utilizar la palabra clave `With`.

```
With orderLog
```

```
    .Log = "Application"
```

```
    .Source = "Application Name"
```

```
    .MachineName = "Computer Name"
```

```
End With
```

- ✚ Se deben usar las instrucciones `Try...Catch` y `Using` cuando emplee el Control de excepciones. No se debe utilizar `On Error Goto`.

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

✚ Para la toma de decisiones se recomiendan, entre otras, las instrucciones `If...Then...Else, Select case.... End select`.

Una vez implementada la solución propuesta se procede a realizar una serie de pruebas de aceptación relacionadas con las HU, con el objetivo de comprobar que la extensión creada cumple con las características esperadas.

3.3 Modelo de pruebas

“El Modelo de Pruebas describe la forma en que se probarán los componentes ejecutables del sistema. Este modelo contiene una colección de Casos de Pruebas (CP) que detallan como han de ser probados aspectos específicos del sistema. Constituye una descripción de lo que se probará, y cómo se hará.”
(18)

La realización de pruebas es una tarea importante para el control de la calidad de *software*, por esta razón se realizaron algunas pruebas a nivel unitario durante el transcurso de las iteraciones, para verificar la factibilidad de la solución en cuanto a discrepancias entre los requerimientos y los resultados de la ejecución de la misma.

Las pruebas brindan confianza en la calidad del *software* al encontrar defectos y a su vez esta es incrementada a medida que los defectos encontrados por las pruebas son reparados. En el presente trabajo se llevaron a cabo dos métodos de prueba en aras de diseñar aquellas que descubran diferentes tipos de errores en el menor tiempo y esfuerzo posibles.

Dichos métodos son caja negra y caja blanca, el primero valida la interfaz del *software*, en tanto el segundo asegura selectivamente que el funcionamiento interno del mismo sea correcto. Se realizaron pruebas a nivel de programador y a nivel de cliente con las que se pudo confirmar que la extensión cumple con los requisitos planteados. Una vez terminadas las pruebas se obtuvo un producto con calidad aceptado por los usuarios.

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

Se generó la plantilla de Casos de prueba de aceptación (CPA) correspondiente a las HU implementadas.

Tabla 11: Caso de prueba de aceptación 1

Caso de Prueba de Aceptación	
Código Caso de Prueba: 01	Nombre Historia de Usuario: Gestionar documentos
Nombre de la persona que realiza la prueba: Yamilka Leyva Coss	
Descripción de la Prueba: Prueba la funcionalidad mostrar los documentos abiertos de LibreOffice.	
Condiciones de Ejecución: Deben haber documentos de LibreOffice Writer abiertos.	
Entrada / Pasos de ejecución: 1. Acceder al ícono de la extensión.	
Resultado Esperado: Deben mostrarse todos los documentos de LibreOffice que estén abiertos, en un diálogo.	
Evaluación de la Prueba: Satisfactoria	

Tabla 12: Caso de prueba de aceptación 2

Caso de Prueba de Aceptación	
Código Caso de Prueba: 02	Nombre Historia de Usuario: Gestionar documentos
Nombre de la persona que realiza la prueba: Yamilka Leyva Coss	
Descripción de la Prueba: Prueba la funcionalidad de abrir un documento desde un directorio existente.	
Condiciones de Ejecución: Debe ser un documento de LibreOffice Writer	
Entrada / Pasos de ejecución: 1. Acceder al ícono de la extensión. 2. Clic en el botón Buscar. 3. Acceder al directorio deseado. 4. Seleccionar el documento.	

Capítulo 3: Implementación y pruebas de la extensión de Vista paralela para LibreOffice Writer

5. Clic en el botón Aceptar
Resultado Esperado: Debe permitir abrir un documento desde un directorio determinado.
Evaluación de la Prueba: Satisfactoria

Tabla 13: Caso de prueba de aceptación 3

Caso de Prueba de Aceptación	
Código Caso de Prueba: 03	Nombre Historia de Usuario: Gestionar documentos
Nombre de la persona que realiza la prueba: Yamilka Leyva Coss	
Descripción de la Prueba: Prueba la funcionalidad de seleccionar uno de los documentos abiertos de LibreOffice.	
Condiciones de Ejecución: Deben haber documentos abiertos de LibreOffice Writer.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Acceder al ícono de la extensión. 2. Seleccionar el documento deseado. 3. Clic en el botón Aceptar. 	
Resultado Esperado: Solo debe permitir seleccionar uno de los documentos mostrados.	
Evaluación de la Prueba: Satisfactoria	

Conclusiones del capítulo

Con el desarrollo de las tareas de ingeniería y el plan de iteraciones se logró mayor organización en el proceso de creación de la extensión, lo cual fue apoyado por la utilización del estándar de codificación en el perfeccionamiento de las líneas de implementación, las pruebas que se realizaron a la aplicación arrojaron resultados satisfactorios, de acuerdo a dichos resultados se logra la culminación exitosa de la propuesta, lo cual se constató con la solución de los problemas.

Conclusiones generales

Conclusiones generales

Con la realización de la presente investigación se obtuvo un producto totalmente libre, que automatiza el proceso de situar dos documentos de LibreOffice Writer en pantalla, uno al lado del otro.

La extensión favorece la labor de lectura, minimiza los esfuerzos y reduce el tiempo de revisión de los documentos, lo cual representa un éxito para el proyecto Personalización Ofimática ya que contribuye a la satisfacción de los usuarios de la *suite*.

La documentación de este trabajo constituye una guía para futuros trabajos relacionados con la creación de extensiones para LibreOffice.

Recomendaciones

Recomendaciones

Comenzar la explotación de la extensión dentro de la Universidad de las Ciencias Informáticas, tal que se puedan identificar posibles mejoras para esta.

Implementar la funcionalidad “Desplazamiento sincrónico”, para poder sincronizar el desplazamiento de ambos documentos, una vez situados en paralelo.

Integrar los resultados a la personalización cubana de LibreOffice.

Referencias bibliográficas

Referencias

1. Definicion.de. [En línea] [Citado el: 22 de 1 de 2014.] <http://definicion.de/extension/>.
2. The Document Foundation. LibreOffice.org. [En línea] 2013. [Citado el: 11 de 2 de 2013.] <http://www.libreoffice.org>.
3. Wiki Apache OpenOffice.org. [En línea] [Citado el: 25 de 2 de 2014.] <http://wiki.open-office.es/Extensiones>.
4. Marcich, Marino. ODF: el Formato de documento emergente a elección de los gobiernos. [En línea] 12 de 2006. <http://www.ati.es/novatica>.
5. Pereira de la Rosa, Adisleidy. *Propuesta de solución para la conversión de formatos de archivos a estándares abiertos*. 2013.
6. Definición. de. [En línea] [Citado el: 2 de 4 de 2014.] <http://definicion.de/zip/>.
7. Wiki Apache OpenOffice.org. [En línea] [Citado el: 8 de 2 de 2014.] <http://wiki.openoffice.org/wiki/Uno>.
8. *OpenOffice.org Developer's Guide*. s.l. : Sun Microsystems. 3.1.
9. The Document Foundation. LibreOffice.org. [En línea] 2014. https://help.libreoffice.org/Common/Extension_Manager/es.
10. Pitonyak, Andrew. www.pitonyak.org. [En línea] 3 de 3 de 2014. <http://www.pitonyak.org/oo.php>.
11. Peñalver Romero, Gladys Marsi y Meneses Abad, Abel. *SXP, metodología ágil para proyectos de software libre*. 2009.
12. freedownloadmanager.org. [En línea] [Citado el: 5 de 3 de 2014.] www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D.
13. softpedia.es. [En línea] 5 de 4 de 2014. <http://www.softpedia.es/programa-BasicAddonBuilder-105982.html>.
14. Wiki Apache OpenOffice.org. [En línea] 25 de 2 de 2014. http://wiki.open-office.es/El_IDE_de_OpenOffice_Basic.
15. Garro, Arturo. *Guia del programador basic*. [En línea] 2003. <http://ggyma.geo.ucm.es/docencia/documentos/informatica/Guia%20del%20programador%20BASIC.pdf>.
16. Wikipedia, la enciclopedia libre. [En línea] [Citado el: 6 de 3 de 2014.] [http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software)).
17. [doc.ubuntu-es](http://doc.ubuntu-es.org). [En línea] [Citado el: 6 de 3 de 2014.] http://doc.ubuntu-es.org/Rapid_SVN.

Referencias bibliográficas

18. Jacobson, I. y Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2004.

19. OpenOffice.org. [En línea] 4 de 3 de 2014.

<http://www.openoffice.org/api/docs/common/ref/com/sun/star/module-ix.html>.

20. Reynoso, Carlos y Kicillof, Nicolás. Entorno Virtual de Aprendizaje. [En línea] 2004. [Citado el: 28 de 4 de 2014.]

http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Arquitectura_de_Software/Estilos_y_Patrones_en_la_Estrategia_de_Microsoft.pdf.

Bibliografía

Bibliografía

1. Definicion.de. [En línea] [Citado el: 22 de 1 de 2014.] <http://definicion.de/extension/>.
2. The Document Foundation. LibreOffice.org. [En línea] 2013. [Citado el: 11 de 2 de 2013.] <http://www.libreoffice.org>.
3. Wiki Apache OpenOffice.org. [En línea] [Citado el: 25 de 2 de 2014.] <http://wiki.openoffice.es/Extensiones>.
4. Marcich, Marino. ODF: el Formato de documento emergente a elección de los gobiernos. [En línea] 12 de 2006. <http://www.ati.es/novatica>.
5. Pereira de la Rosa, Adisleidy. *Propuesta de solución para la conversión de formatos de archivos a estándares abiertos*. 2013.
6. Definición. de. [En línea] [Citado el: 2 de 4 de 2014.] <http://definicion.de/zip/>.
7. Wiki Apache OpenOffice.org. [En línea] [Citado el: 8 de 2 de 2014.] <http://wiki.openoffice.org/wiki/Uno>.
8. *OpenOffice.org Developer's Guide*. s.l. : Sun Microsystems. 3.1.
9. The Document Foundation. LibreOffice.org. [En línea] 2014. https://help.libreoffice.org/Common/Extension_Manager/es.
10. Pitonyak, Andrew. www.pitonyak.org. [En línea] 3 de 3 de 2014. <http://www.pitonyak.org/oo.php>.
11. Peñalver Romero, Gladys Marsi y Meneses Abad, Abel. *SXP, metodología ágil para proyectos de software libre*. 2009.
12. freedownloadmanager.org. [En línea] [Citado el: 5 de 3 de 2014.] www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D.
13. softpedia.es. [En línea] 5 de 4 de 2014. <http://www.softpedia.es/programa-BasicAddonBuilder-105982.html>.
14. Wiki Apache OpenOffice.org. [En línea] 25 de 2 de 2014. http://wiki.openoffice.es/El_IDE_de_OpenOffice_Basic.
15. Garro, Arturo. Guía del programador basic. [En línea] 2003. <http://ggyma.geo.ucm.es/docencia/documentos/informatica/Guia%20del%20programador%20BASIC.pdf>.
16. Wikipedia, la enciclopedia libre. [En línea] [Citado el: 6 de 3 de 2014.] [http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software)).
17. [doc.ubuntu-es](http://doc.ubuntu-es.org). [En línea] [Citado el: 6 de 3 de 2014.] http://doc.ubuntu-es.org/Rapid_SVN.
18. Jacobson, I. y Jacobson, I. *El Proceso Unificado de Desarrollo de Software*. Madrid : Addison Wesley, 2004.
19. OpenOffice.org. [En línea] 4 de 3 de 2014. <http://www.openoffice.org/api/docs/common/ref/com/sun/star/module-ix.html>.

Bibliografía

20. EcuRed. [En línea] [Citado el: 6 de 6 de 2014.]
http://www.ecured.cu/index.php/Suite_ofim%C3%A1tica.
21. Reynoso, Carlos y Kicillof, Nicolás. Entorno Virtual de Aprendizaje. [En línea] 2004. [Citado el: 28 de 4 de 2014.]
http://eva.uci.cu/file.php/158/Documentos/Recursos_bibliograficos/Libros_y_articulos_UD_1/Arquitectura_de_Software/Estilos_y_Patrones_en_la_Estrategia_de_Microsoft.pdf.
22. Pressman R, Roger. *Ingeniería del software: un enfoque práctico. Quinta edición*. Madrid : McGraw-Hill, 2002. ISBN 84.
23. Popov, Dimitri. linux-magazine.es. [En línea] 25 de Febrero de 2014. http://www.linux-magazine.es/issue/69/069-072_WorkspaceLM69.pdf.
24. F.P., María José. *Libertad en el escritorio: estándares y el trabajo del traductor. Facultad de filosofía*.
25. Baeza Servín, Mauricio. *Aprendiendo OOO Basic*. México : s.n., 2011.
26. Free Software Foundation. *The GNU General Public License v3.0 - GNU Project - Free Software Foundation*. [En línea] 6 de Mayo de 2014. <http://www.gnu.org/licenses/gpl-3.0.html>.
27. Wiki Apache OpenOffice. [En línea] 10 de 3 de 2014.
https://wiki.openoffice.org/wiki/Extensions_development_basic#Xray_tool.
28. Wiki Apache OpenOffice.org. [En línea] 3 de Marzo de 2014.
<https://wiki.openoffice.org/wiki/Documentation/DevGuide/OfficeDev/Windows>.
29. Wiki Apache OpenOffice.org. [En línea] 25 de 2 de 2014. http://wiki.openoffice.es/Contenedores_librerias_y_modulos_en_OpenOffice_Basic.
30. Mora F., Walter. *Programación con OpenOffice.org Basic y OOO Calc*. Costa Rica : s.n. ISBN.

Glosario de términos

Glosario de términos

API: *Application Programming Interface*, en español Interfaz de Programación de Aplicaciones.

CORBA: *Common Object Request Broker Architecture* es un estándar que permite que diversos componentes de *software* escritos en múltiples lenguajes de programación y que corren en diferentes computadoras, puedan trabajar juntos, facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos.

Estándar: En el campo de la informática un estándar es una especificación que regula como se realizan ciertos procesos o se fabrican determinados componentes.

GUI: *Graphical User Interface*, en español Interfaz Gráfica de Usuario. Es un conjunto de formas y métodos que posibilitan la interacción de un sistema con los usuarios utilizando formas gráficas e imágenes. Con formas gráficas se refiere a botones, íconos, ventanas, fuentes, etc. los cuales representan funciones, acciones e información.

IDE: *Integrated Development Environment*, en español Entorno de desarrollo integrado.

Interoperabilidad: Capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas, para funcionar con otros productos o sistemas existentes o futuros y eso sin restricción de acceso o de implementación.

JTC1: Es el entorno de desarrollo de las normas que usan los expertos en todo el mundo para desarrollar las tecnologías de la Informática y la Comunicaciones (TIC).

Macros: Instrucciones que se guardan en módulos para decirle al programa qué hacer, cómo, cuándo, el orden y la secuencia en que se desea que lo haga.

Microsoft Office: *Suite* ofimática de la empresa Microsoft.

Glosario de términos

Módulos: Elementos contenedores de las macros y funciones de LibreOffice *Basic*.

OASIS: Consorcio sin fines de lucro que impulsa el desarrollo, la convergencia y la adopción de estándares abiertos para la sociedad de la información mundial.

Object: Tipo de dato, que puede contener propiedades y / o métodos.

Ofimática: Hace referencia a la automatización de las actividades que se realizan en una oficina.

OLE: *Object Linking and Embedding*, en español Incrustación y enlazado de objetos. Es el nombre de un sistema de objetos distribuido y un protocolo desarrollado por Microsoft.

OOo: OpenOffice.org es una *suite* ofimática libre (código abierto y distribución gratuita).

SDK: *Software Development Kit*, en español *Kit* de Desarrollo de *Software*. Es un conjunto de herramientas de desarrollo de *software* para crear aplicaciones para un sistema concreto.

Suite Ofimática: Conjunto de programas de ordenador diseñados para el trabajo de oficina. Las mismas incluyen procesador de texto, hoja de cálculo, gestión de base de datos, cliente de correo electrónico, agenda y administrador de presentaciones o diapositivas.

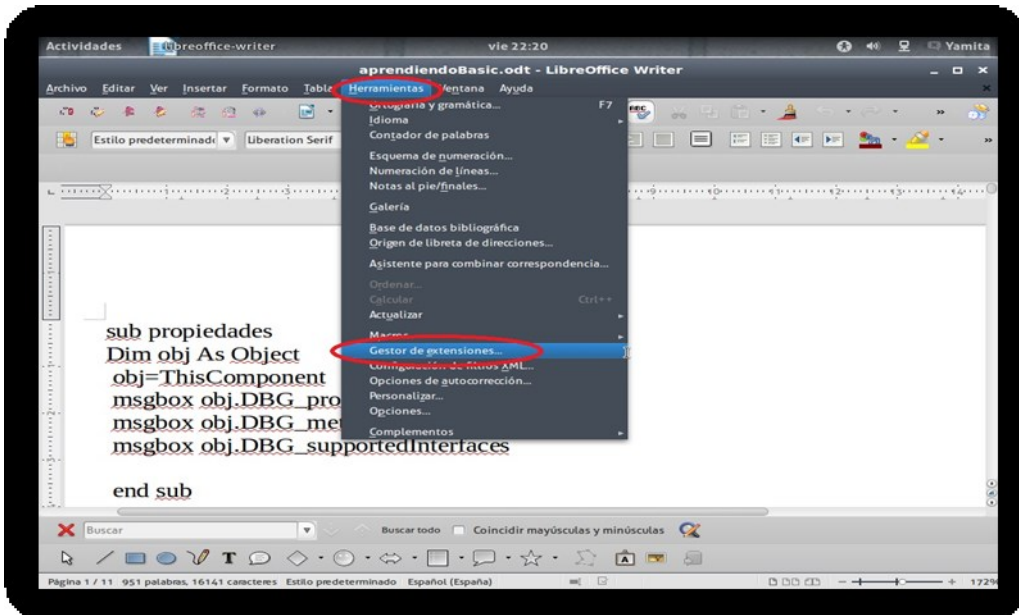
UNO: *Universal Network Objects*, en español Red de Objetos Universal.

XML: *Extensible Markup Language*, en español Lenguaje de Mercado Extensible.

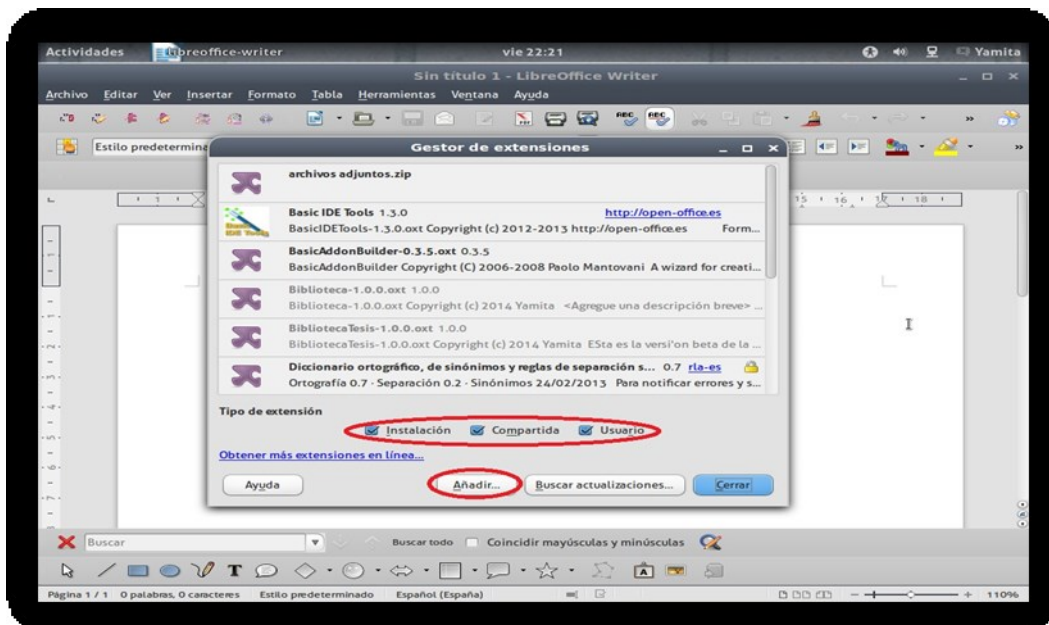
Anexos

Anexos

Anexo 1: Muestra como acceder al Gestor de extensiones

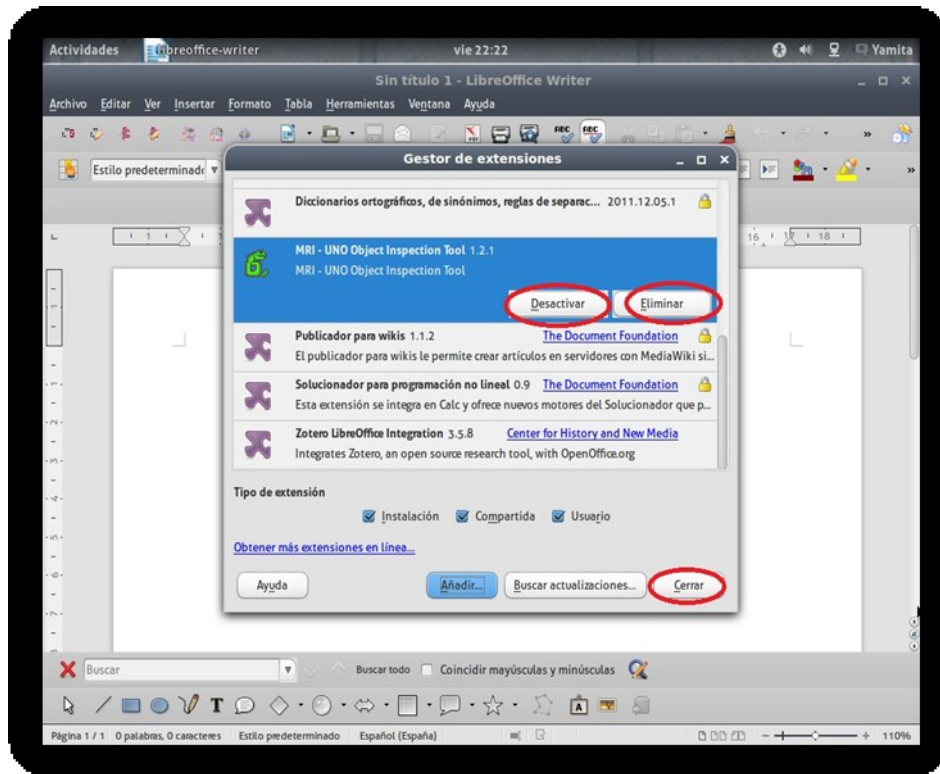


Anexo 2: Instalar una extensión para LibreOffice usando el Gestor de extensiones



Anexos

Anexo 3: Desinstalar una extensión de LibreOffice a través del Gestor de extensiones

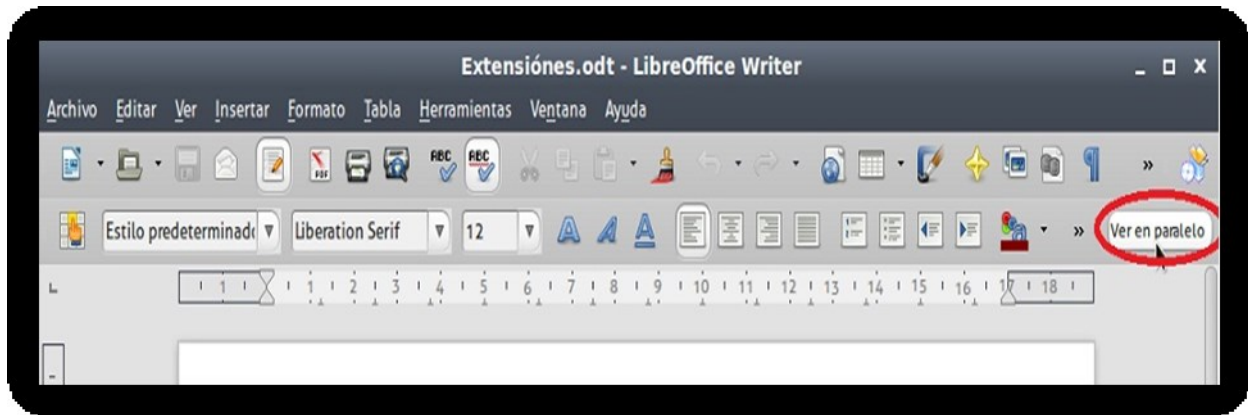


Anexo 4: Muestra los documentos abiertos de LibreOffice Writer

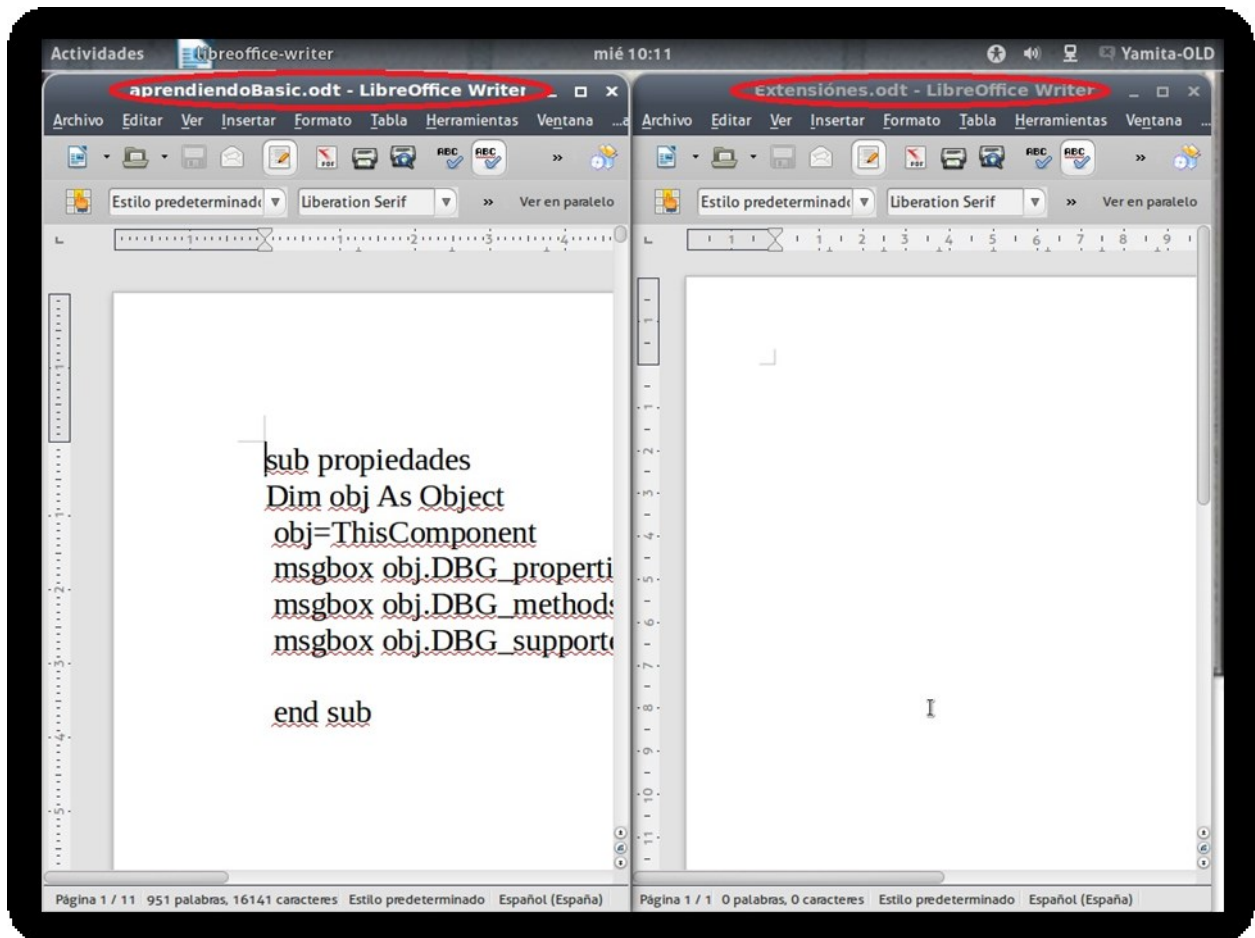


Anexos

Anexo 5: Barra de herramientas Ver en paralelo

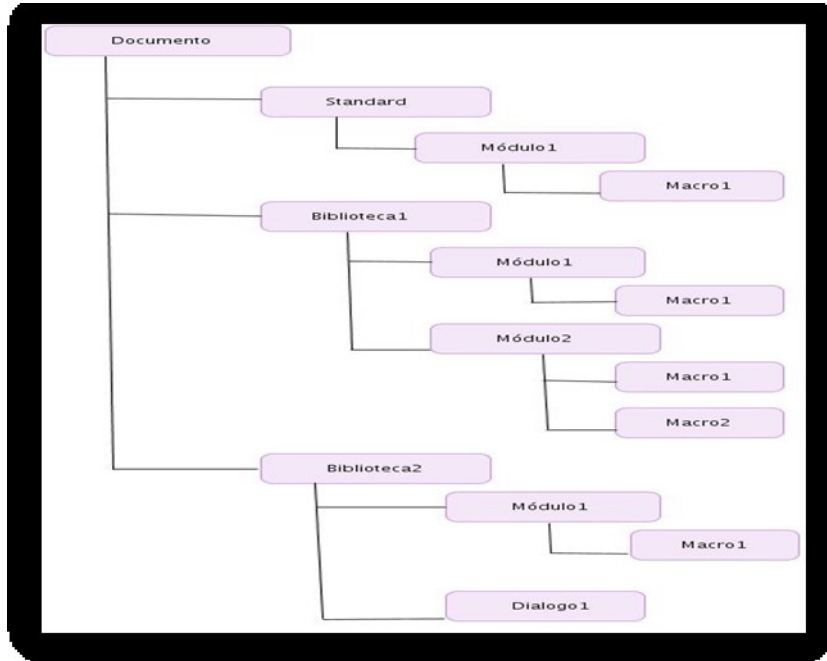


Anexo 6: Documentos en paralelo



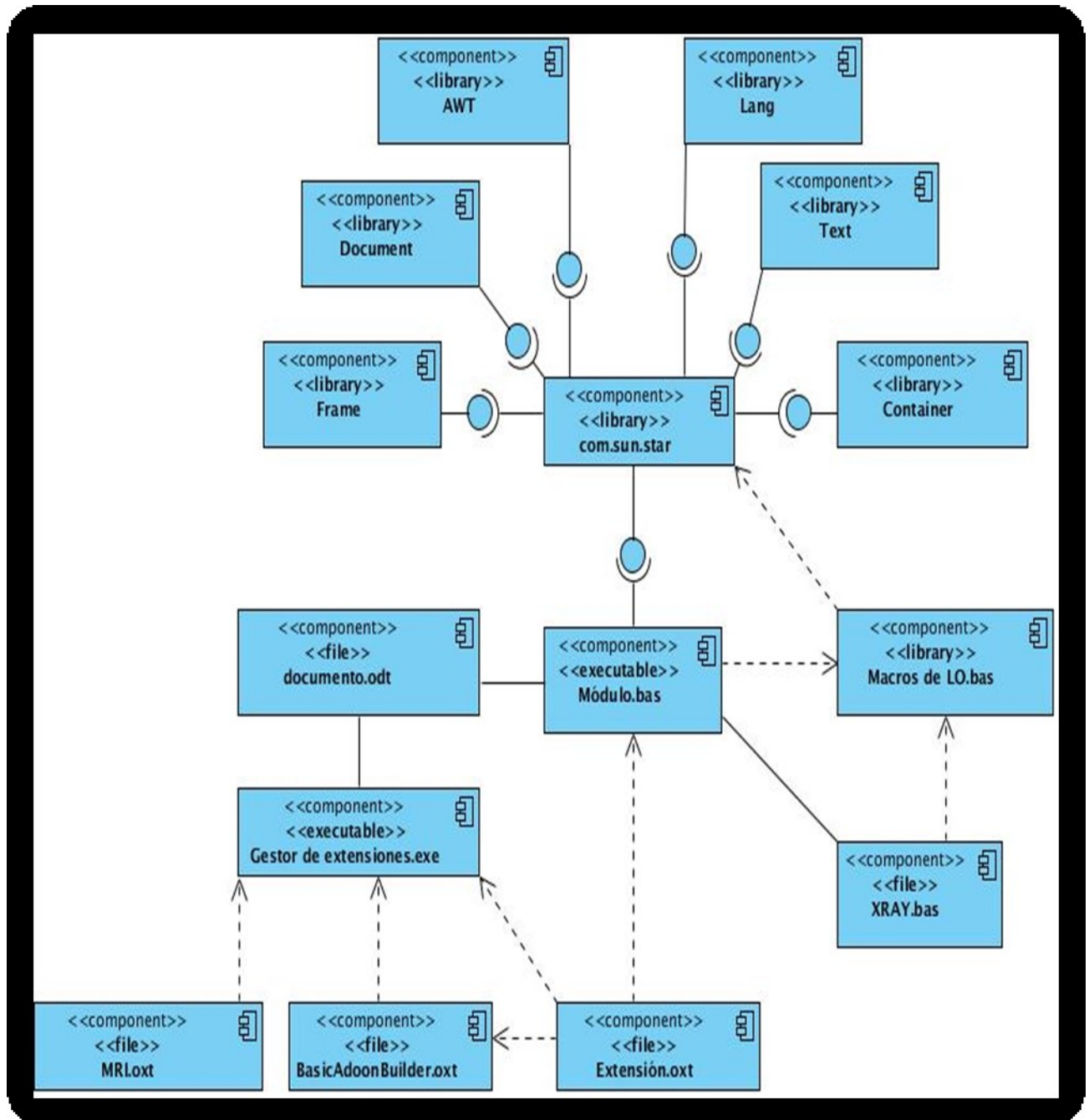
Anexos

Anexo 7: Estructura de un documento de LibreOffice Writer



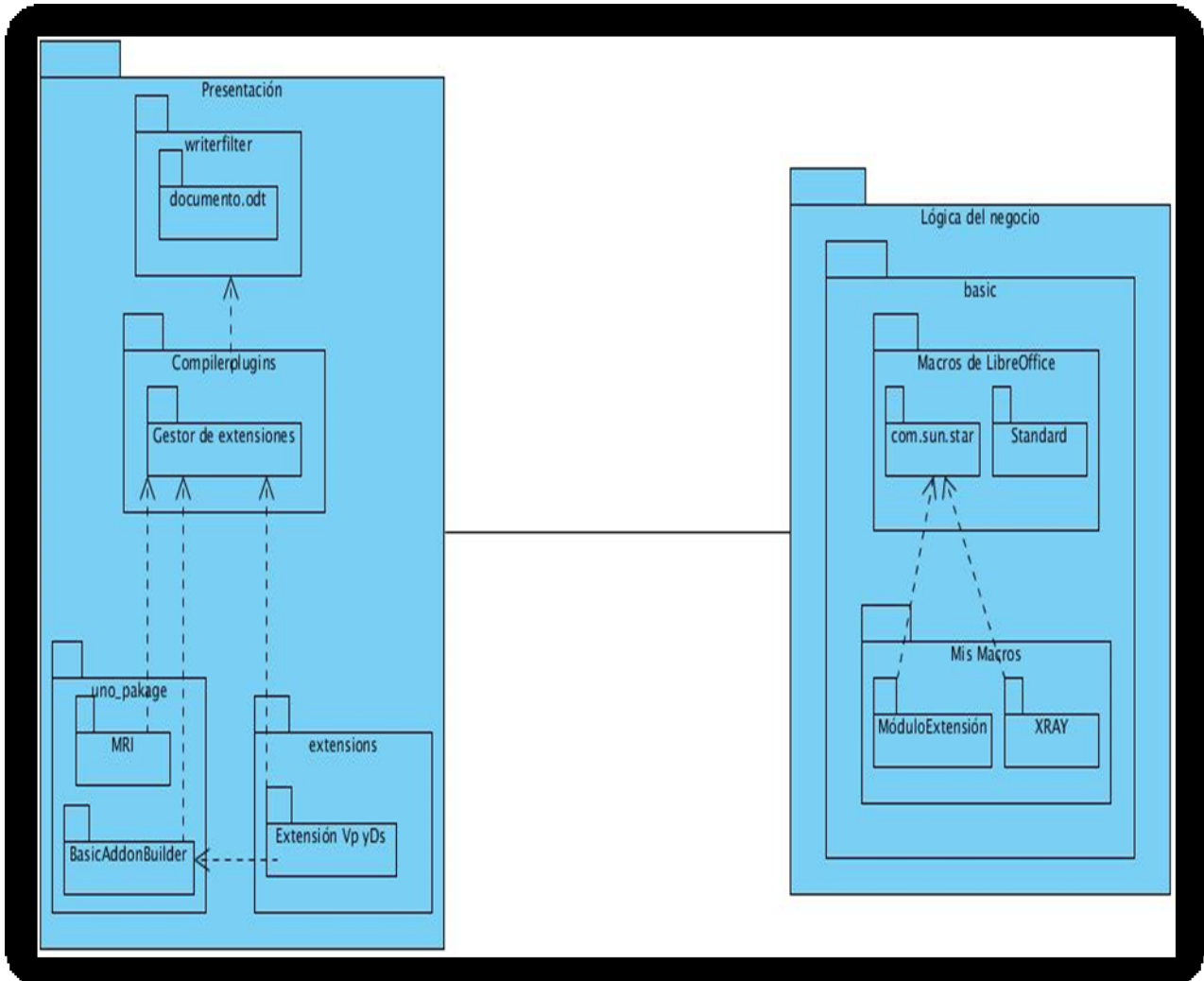
Anexos

Anexo 8: Diagrama de componentes del diseño



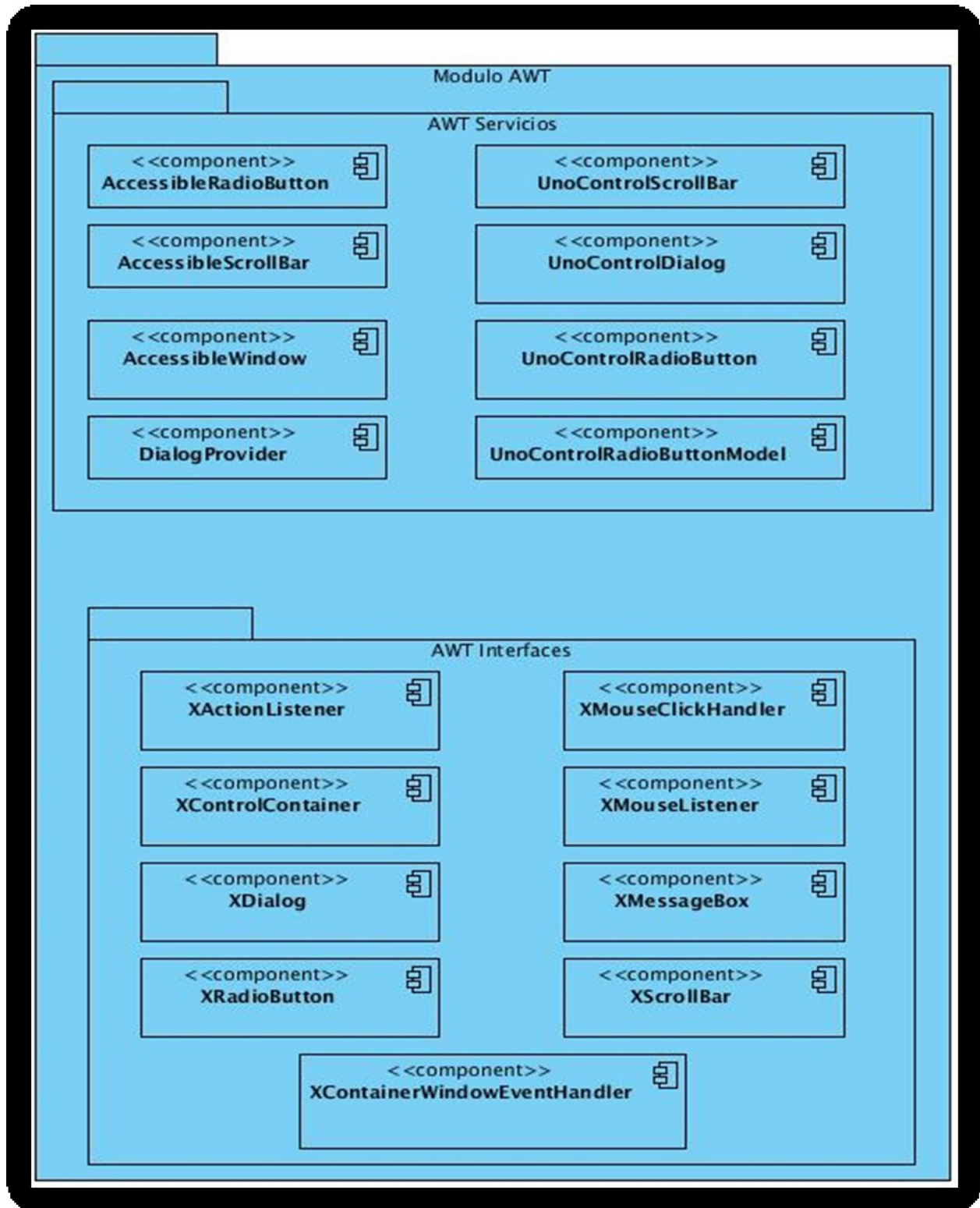
Anexos

Anexo 9: Diagrama de paquetes del diseño



Anexos

Anexo 10: Diagrama de componentes del módulo AWT



Anexos

Anexo 11: Descripción de los servicios e interfaces del módulo AWT

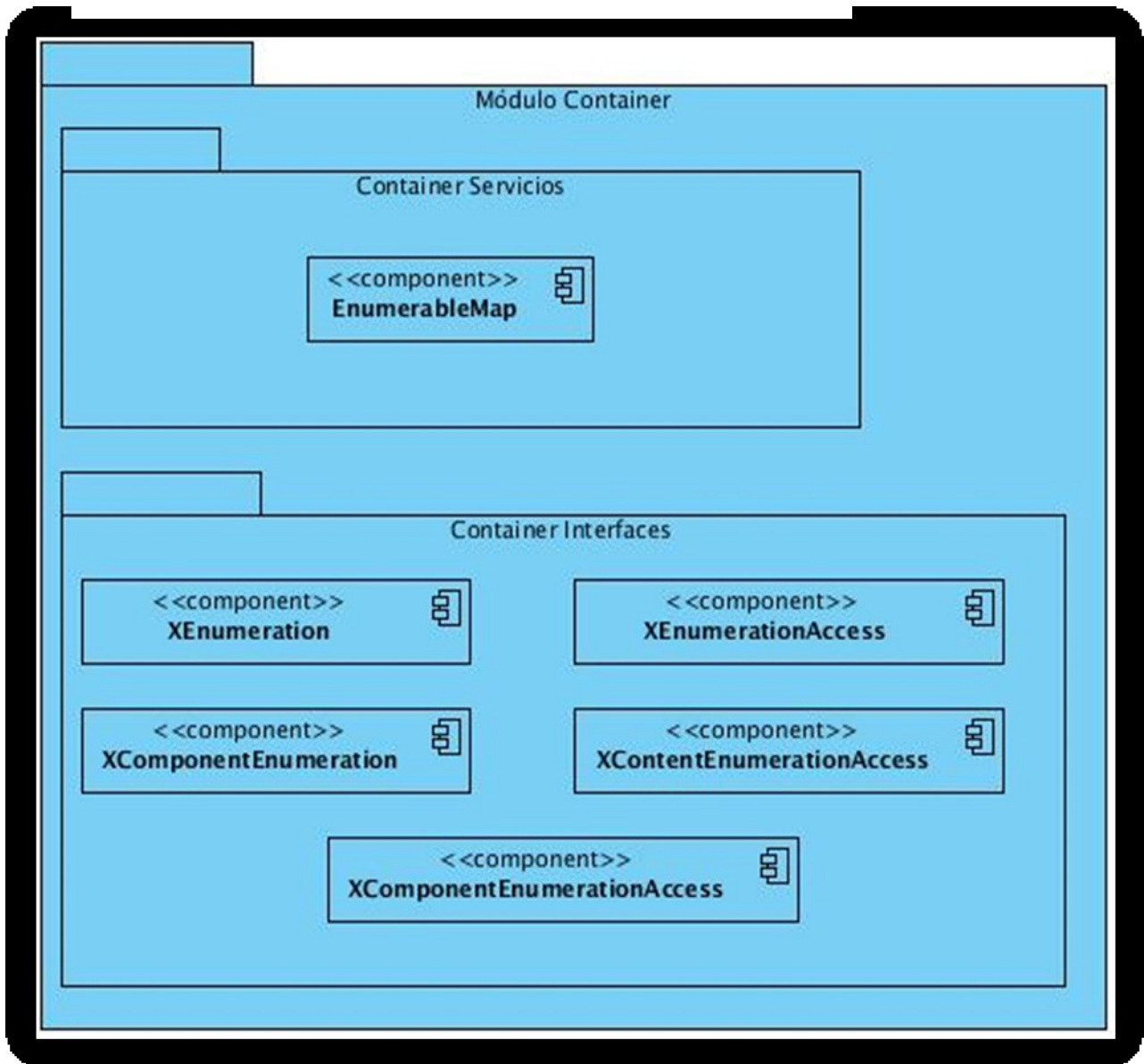
Módulo: awt	
Servicios	Descripción
AccessibleRadioButton	Especifica el acceso a un botón de radio.
AccessibleScrollBar	Especifica el acceso a una barra de desplazamiento.
AccessibleWindow	Especifica el acceso a una ventana.
DialogProvider	Proporciona diálogos que implementan la interfaz Xdialog.
UnoControlRadioButton	Especifica el control de los botones de radio.
UnoControlRadioButtonModel	Especifica el modelo estándar del servicio UnoControlRadioButton.
UnoControlScrollBar	Especifica el control de la barra de desplazamiento.
UnoControlDialog	Especifica el control del diálogo.
Interfaces	Descripción
XActionListener	Hace que sea posible la recepción de eventos.
XScrollBar	Proporciona el acceso al valor y la configuración de una barra de desplazamiento y permite registrar detectores de eventos.
	Facilita el acceso al estado de un botón de

Anexos

XRadioButton	opción y hace que sea posible registrar detectores de eventos de elementos.
XControlContainer	Proporciona acceso a los controles dentro de un objeto de tipo UnoControlContainer.
XDialog	Hace posible mostrar y ocultar un cuadro de diálogo y le da acceso al título del cuadro de diálogo.
XMessageBox	Permite el acceso a un cuadro de mensaje.
XMouseClickedHandler	Hace que sea posible la recepción de eventos desde el ratón a una ventana.

Anexos

Anexo 12: Diagrama de componentes del módulo container



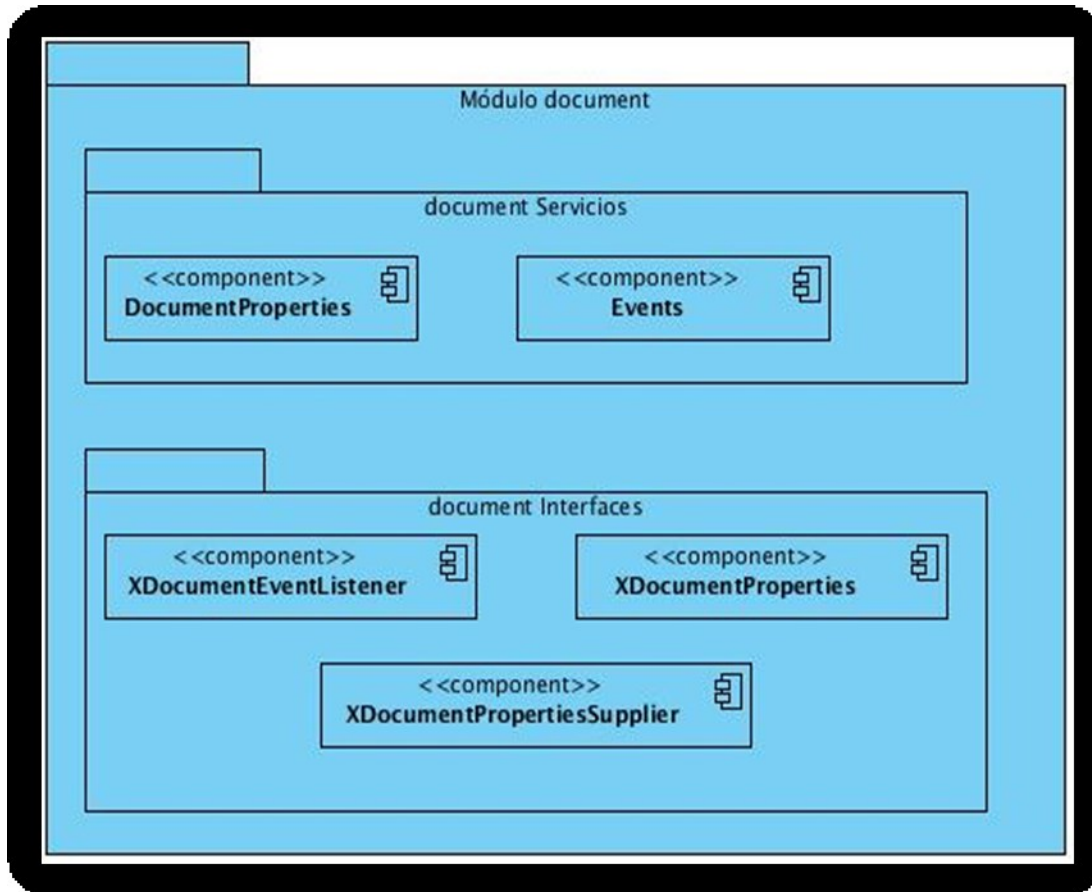
Anexos

Anexo 13: Descripción de los servicios e interfaces del módulo container

Módulo: container	
Servicios	Descripción
EnumerableMap	Proporciona una implementación por defecto XEnumerableMap
Interfaces	Descripción
XEnumeration	Proporciona la funcionalidad para enumerar el contenido de un objeto.
XComponentEnumeration	Proporciona una enumeración tipificada a través de componentes.
XEnumerationAccess	Utilizado para enumerar los objetos y almacenarlos en otro objeto para acceder a cada uno de ellos, similar a un arreglo de objetos.
XContentEnumerationAccess	Permite el acceso a las colecciones de todos los tipos de datos dentro del objeto.
XComponentEnumerationAccess	Proporciona una interfaz para una enumeración tipificada a través de una colección de componentes.

Anexos

Anexo 14: Diagrama de componentes del módulo document



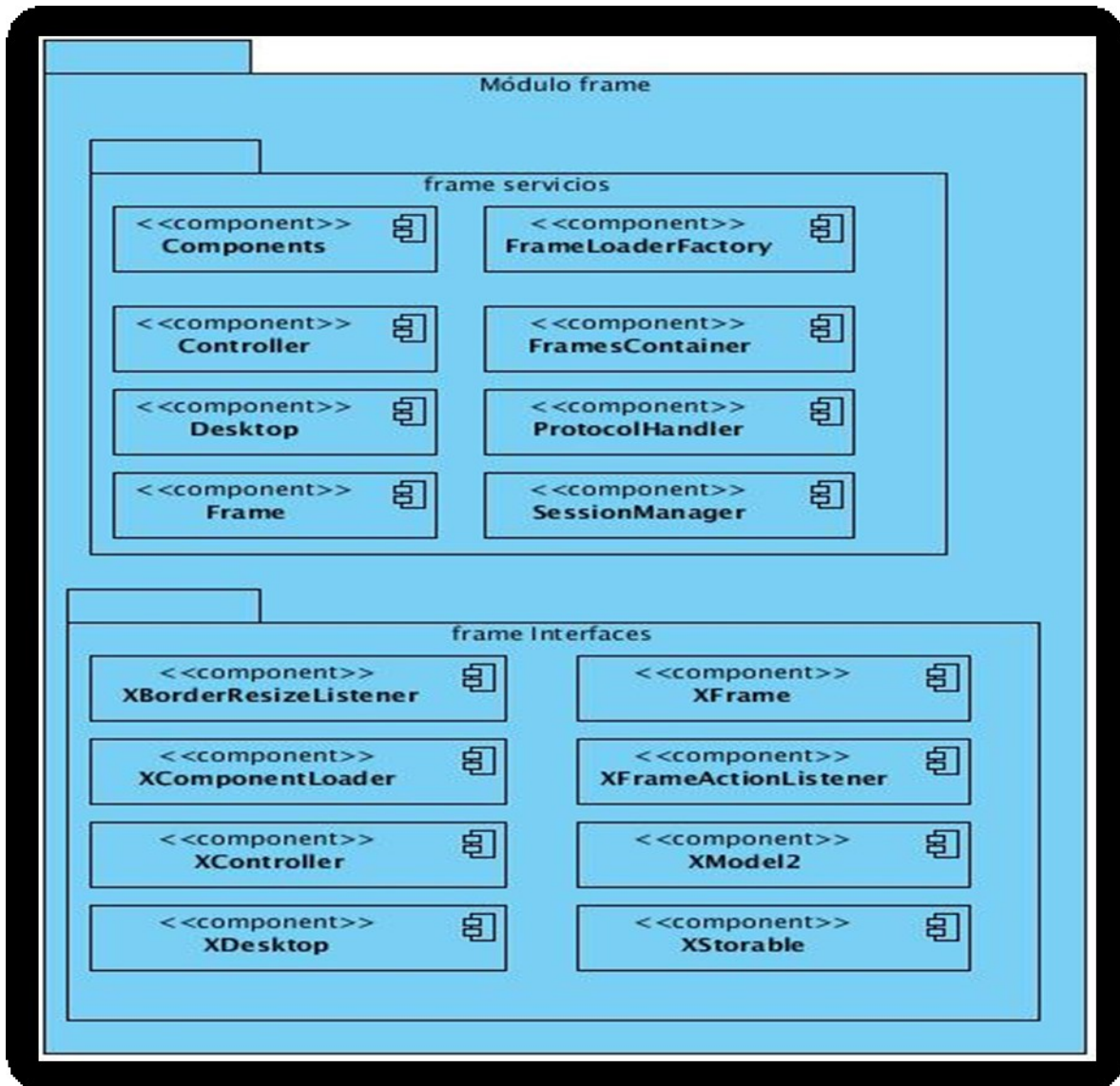
Anexos

Anexo 15: Descripción de los servicios e interfaces del módulo document

Módulo: document	
Servicios	Descripción
DocumentProperties	Proporciona información específica del documento, como el autor, fecha de creación, y los campos definidos por el usuario.
Events	Es una colección de todos los eventos, con el apoyo de un documento o el contenido de un documento
Interfaces	Descripción
XDocumentEventListener	Permite recibir notificaciones de los eventos que suceden en un documento.
XDocumentProperties	Proporciona información específica del documento, como el autor, fecha de creación, y los campos definidos por el usuario.
XDocumentPropertiesSupplier	Facilita el acceso a la instancia XDocumentProperties de un documento.

Anexos

Anexo 16: Diagrama de componentes del módulo frame



Anexo 17: Descripción de los servicios e interfaces del módulo frame

Módulo: frame	
Servicios	Descripción
Components	Especifica una colección de componentes.

Anexos

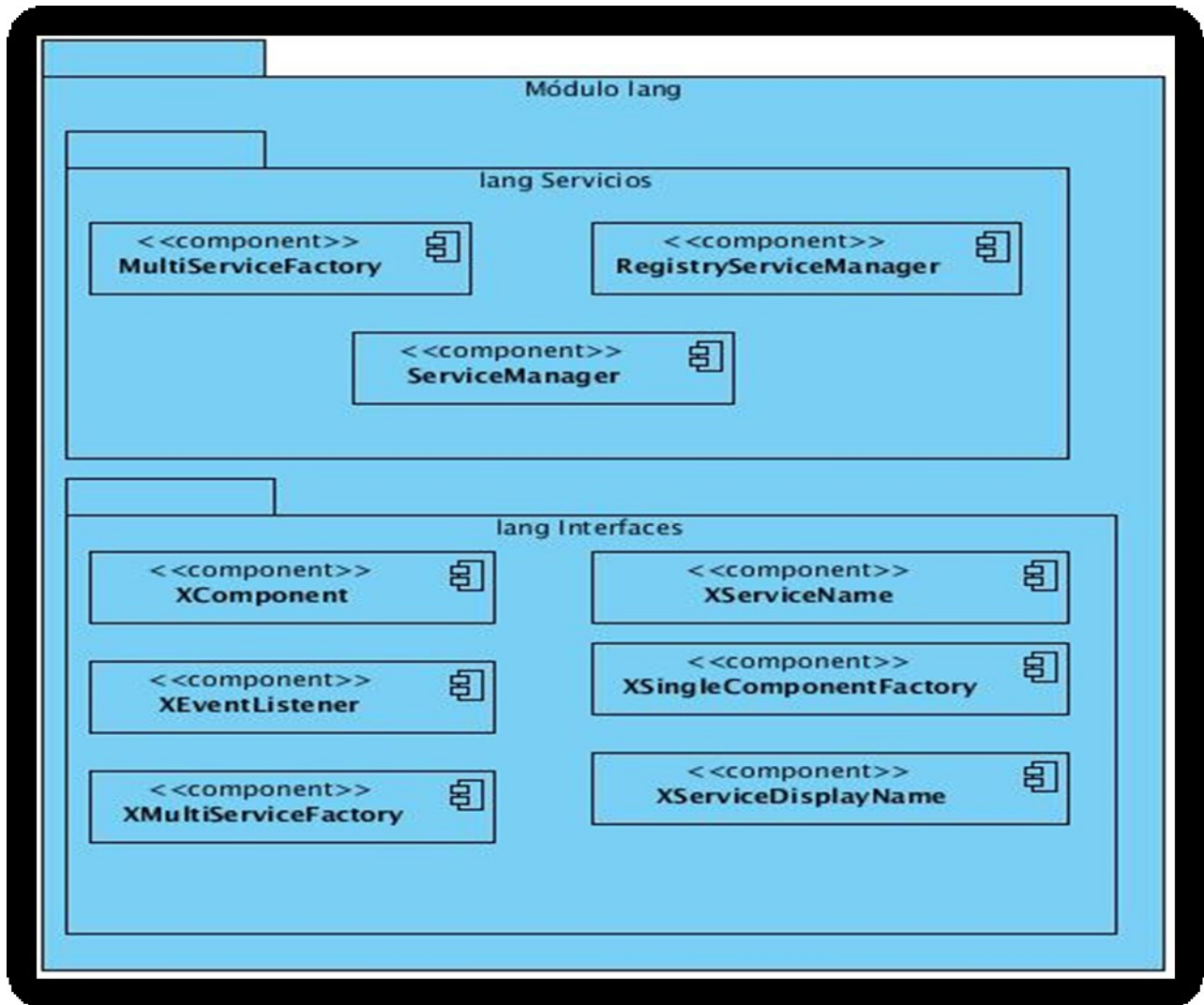
Controller	Es un servicio abstracto para un componente que ofrece una integración más profunda con los componentes de escritorio, tal es el caso de la interfaz AWTXWindow.
Desktop	Es el entorno para crear instancias de componentes.
Frame	Representa el entorno para un componente de escritorio
FramesContainer	Funciona como un recipiente especial que puede contener marcos.
ProtocolHandler	Proveedor especial registrado para protocolos URL.
SessionManager	El servicio SessionManager proporciona una interfaz para el gestor de sesiones de escritorio. Un gestor de sesiones realiza un seguimiento de las aplicaciones que se ejecutan cuando el escritorio se apaga y las vuelve a empezar en el mismo estado en que quedaron cuando el escritorio se pone en marcha la próxima vez. Para poder hacer esto, el gestor de sesiones necesita la cooperación de las aplicaciones; estas tienen que proporcionar información suficiente para volverse a iniciar, así como restaurar el estado en que se quedaron.
FrameLoaderFactory	Crea objetos para cargar los marcos.
Interfaces	Descripción
XBorderResizeListener	Permite analizar los eventos de cambio de tamaño de la frontera de un controlador.
XComponentLoader	Es la interfaz para cargar los componentes de una URL en un entorno de dado.

Anexos

XController	Con esta interfaz, los componentes de un marco pueden funcionar con el uso de eventos.
XDesktop	Esta es la interfaz principal de un servicio de escritorio.
XFrame	Un objeto que soporta un marco puede ser considerado como un objeto "ancla", al cual un componente se puede conectar.
XFrameActionListener	Es usado para que un objeto pueda recibir eventos, cuando se ejecutan varias acciones con los componentes del marco de escritorio.
XModel	Representa un componente que se crea a partir de una URL.
XStorable	Ofrece una forma sencilla de guardar un componente en una URL.

Anexos

Anexo 18: Diagrama de componentes del módulo lang



Anexos

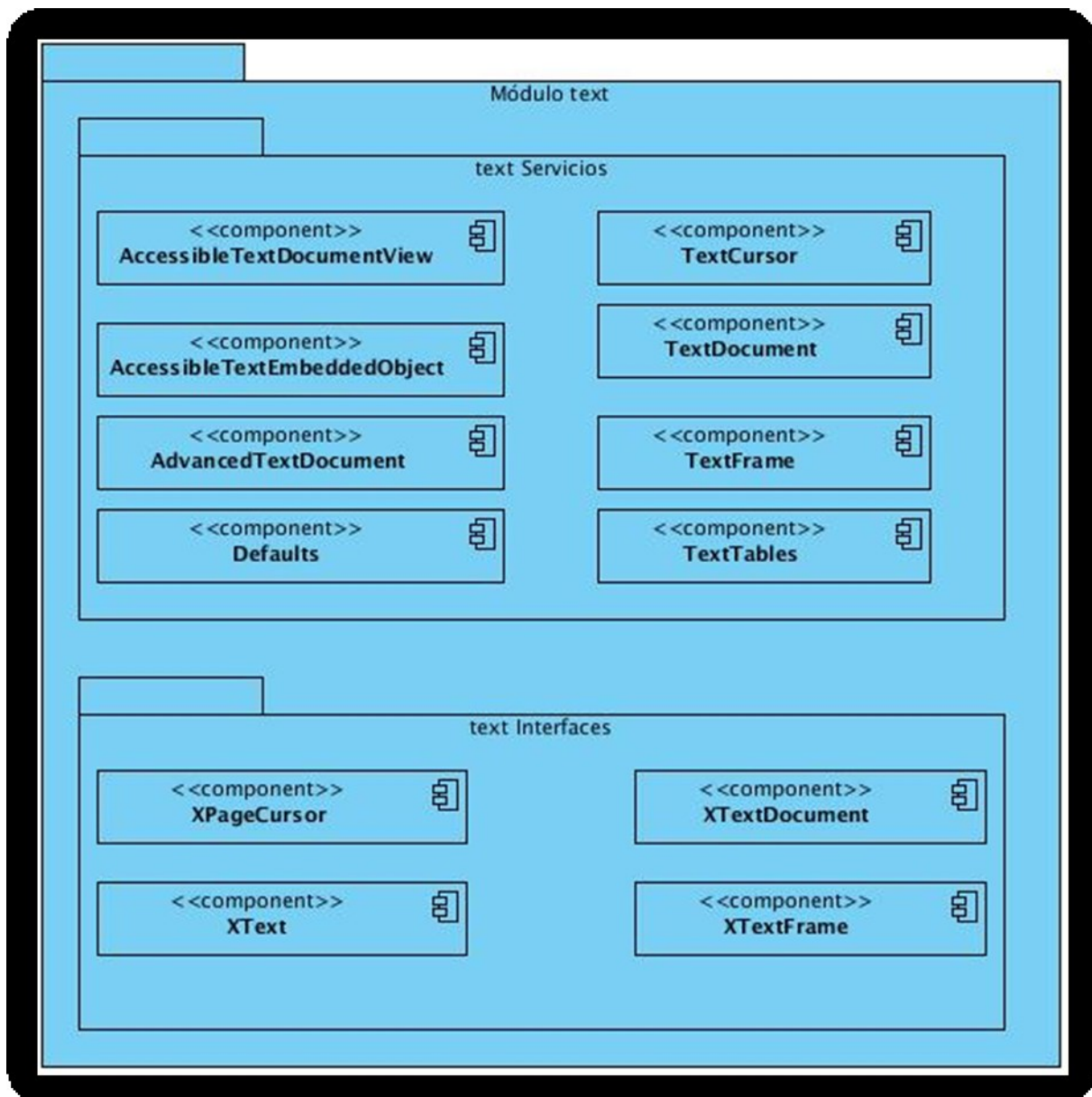
Anexo 19: Descripción de los servicios e interfaces del módulo lang

Módulo: lang	
Servicios	Descripción
MultiServiceFactory	Proporciona una colección de implementaciones de los servicios.
RegistryServiceManager	Proporciona una colección de implementaciones de servicios de lectura a partir de un registro de almacenamiento persistente.
ServiceManager	Proporciona una colección de implementaciones de los servicios. Este es un producto único que comúnmente se encuentra como <code>com.sun.star.lang.theServiceManager</code> .
Interfaces	Descripción
XComponent	Permite usar recursos libres y romper las referencias cíclicas.
XEventListener	Interfaz base para todas las interfaces de objetos detectores de eventos.
XMultiServiceFactory XServiceName	Interfaz de creación de instancias para dar un nombre, por ejemplo, un nombre de servicio. Identifica el objeto con un nombre de servicio que se puede utilizar para crear otro objeto.
XServiceDisplayName	Proporciona un nombre a un servicio para ser utilizado en la pantalla.

Anexos

XSingleComponentFactory Interfaz para crear instancias de la implementación de un servicio específico.

Anexo 20: Diagrama de componentes del módulo text



Anexos

Anexo 21: Descripción de los servicios e interfaces del módulo text

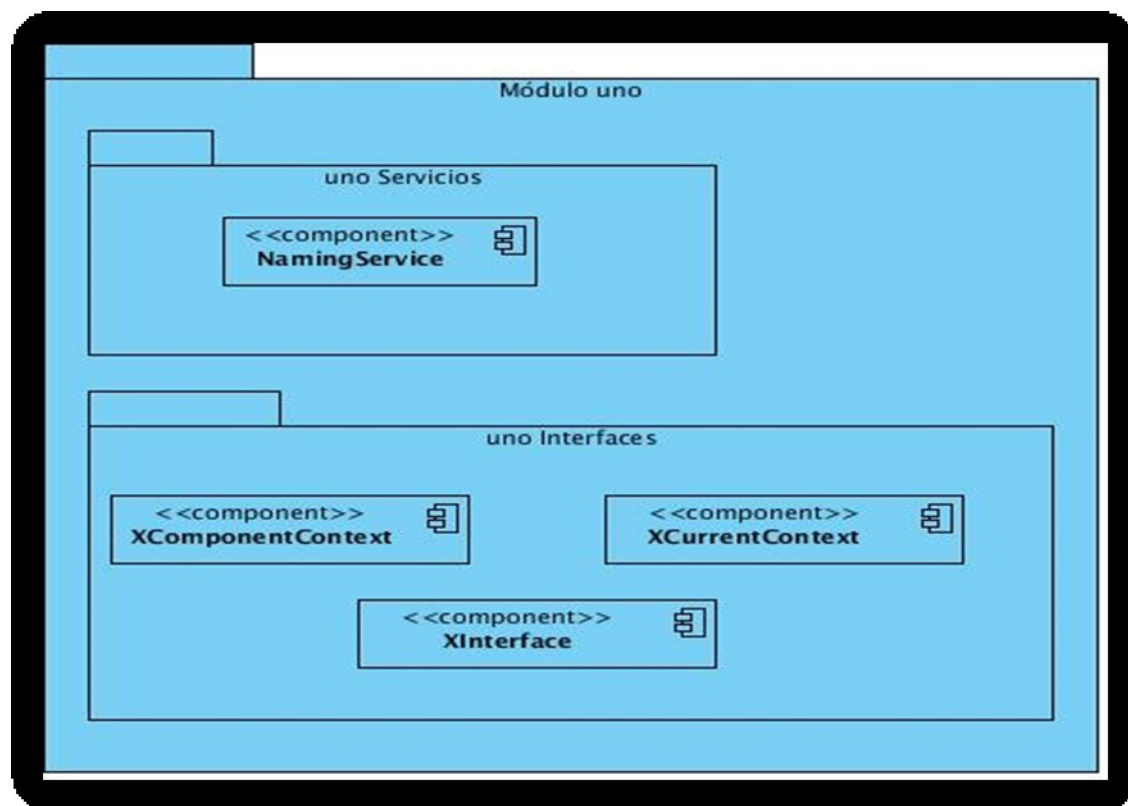
Módulo: text	
Servicios	Descripción
AccessibleTextDocumentView	La vista accesible de un documento de texto.
AccessibleTextEmbeddedObject	Proporciona la vista accesible de un objeto incrustado inactivo. Si un objeto incrustado se activa, el documento activo obtiene un árbol de objetos accesibles propio. El árbol sólo existe mientras que el objeto está desactivado.
AdvancedTextDocument	Un documento de texto avanzado es una extensión de un documento de hipertexto, que también puede contener texto-frames, los pies de notas finales, y algunos otros contenidos avanzados.
Defaults	Proporciona la configuración predeterminada de un componente de texto, párrafo y las propiedades del carácter
TextDocument	Un TextCursor es un TextRange que se puede mover dentro de un objeto de texto.
TextTable	Es una tabla de celdas de texto que se anclan a un texto que lo rodea.
TextRange	Apunta a una secuencia de caracteres dentro de un texto.

Anexos

Interfaces	Descripción
XPageCursor	Hace que sea posible llevar a cabo los movimientos del cursor entre páginas.
XTextContent	Permite que los objetos sean insertados en un texto y proporciona su ubicación una vez que se insertan.
XTextDocument	Es la interfaz principal de un documento de texto.
XTextFrame	Proporciona acceso al texto dentro de un marco de texto.

Anexos

Anexo 22: Diagrama de componentes del módulo uno

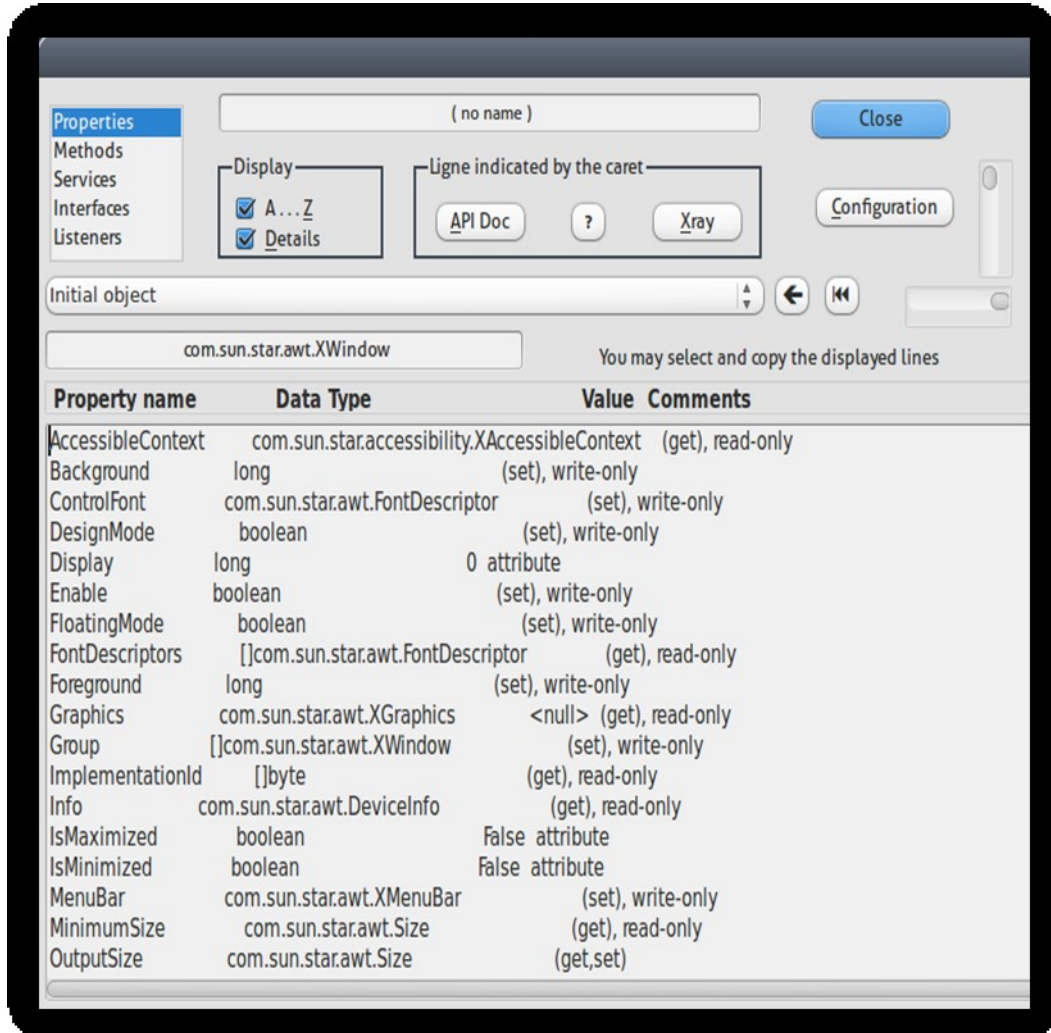


Anexo 23: Descripción de los servicios e interfaces del módulo uno

Módulo: uno	
Servicios	Descripción
NamingService	Contenedor de objetos por su nombre.
Interfaces	Descripción
XComponentContext	Provee el contexto de componentes que pasa a través de un componente del módulo lang a la interfaz XsingleComponentFactory. Por este se pueden recuperar valores arbitrarios.
XCurrentContext	Elementos del contexto de ejecución local para UNO.
XInterface	Interfaz base de todas las interfaces de UNO.

Anexos

Anexo 24: Ejecución de la macro XRAY



Anexos

Anexo 25: Ejecución de la extensión MRI

