

Universidad de las Ciencias Informáticas



Facultad 1

Título: Asistente para la elaboración de calendarios en LibreOffice Writer.

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autores: Maria del Carmen Rios Jiménez

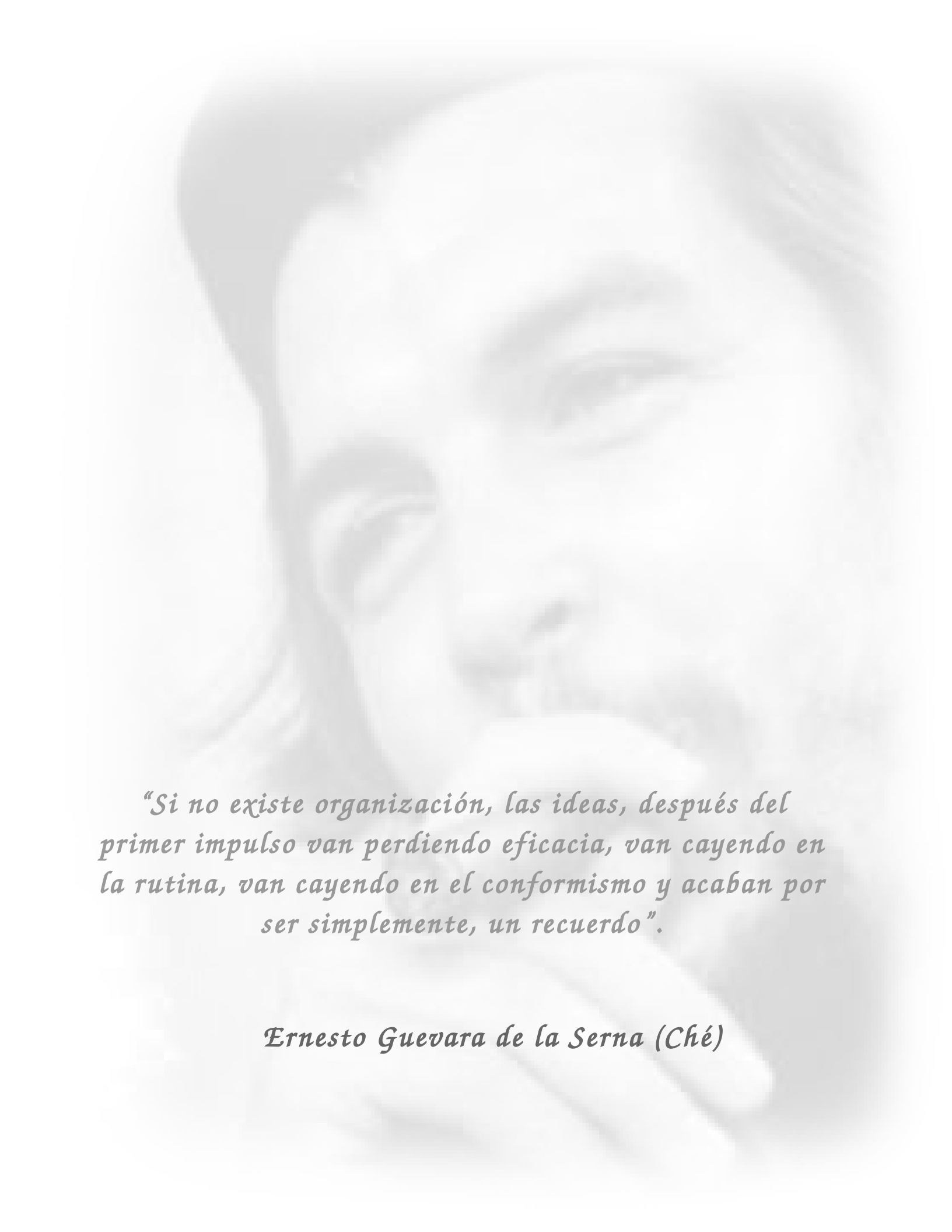
Adriam Delgado Rivero

Tutores: Msc. Angel Goñi Oramas

Ing. Jorge Luis Roque Alvarez

La Habana, Cuba, Junio 2013.

“Año 55 de la Revolución”



“Si no existe organización, las ideas, después del primer impulso van perdiendo eficacia, van cayendo en la rutina, van cayendo en el conformismo y acaban por ser simplemente, un recuerdo”.

Ernesto Guevara de la Serna (Ché)

Declaración de Autoría

Declaramos ser los autores principales del trabajo de diploma “Asistente para la elaboración de calendarios en LibreOffice Writer” y autorizamos al departamento de Servicios Integrales para la Migración, Asesoría y Soporte (SIMAYS) de la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los ____ días del mes de junio del año 2013.

Maria del Carmen Rios Jiménez

Firma del Autor

Adriam Delgado Rivero

Firma del Autor

Msc. Angel Goñi Oramas

Firma del Tutor

Ing. Jorge Luis Roque Alvarez

Firma del Tutor

Agradecimientos

Quiero agradecer primeramente a mi mamá porque desde pequeña me ha dado su apoyo incondicional, por estar siempre tan preocupada por mis estudios y enseñarme a ser sacrificada y persistente. Por quererme tanto y darme siempre lo mejor dentro de lo posible. Esta es la manera de retribuirle tantos años de sacrificio.

A mi hermanito Bradley (Bray), por ser la lucecita que alumbró la vida de mi mamá, mi padrastro y la mía con su nacimiento. A Leonel por ser como un padre para mí del cual nunca me ha faltado el apoyo y ayuda incondicional.

A mi abuela Ramona, mis tíos Eugenito, Marilín, Virgilio, Maida, Frank y Yoli por apoyarme en cada momento de mi vida. A todos mis primos(as) a los que les deseo éxitos en la vida.

Agradezco a mi papá Ernesto por su cariño y dedicación durante todos estos años. A mis abuelos Humberto y Ofelia por brindarme siempre su cariño y apoyo incondicional. A Arismayda y Ernesto Jesús por cuidar de mi papá y brindarme todo su cariño y apoyo. A mis tías Diameli y Diana, así como mi prima Dayi...muchísimas gracias.

Agradezco a mi novio y compañero de tesis Adriam (mi Tata), por tener tanta paciencia, por brindarme su amor y cariño durante toda la carrera, por hacer que mis días en la Universidad fueran lo más lindo posible. También a mis suegros Magnolia y Vladimir, así como a Carlitos y Cinthia por el cariño y la atención que me han dado.

A los tutores Angel Goñi (Goñi) y Jorge Luis (Jorgito) por dedicarnos parte de su escaso tiempo y por contribuir en la realización de la tesis.

A todos mis compañeros(as) de los grupos por los que pasé (102, 207, 309, 410 y 510), en especial a los compañeros del proyecto por ayudarme cuando los necesité: Alexander (el flaco), Yasiel, Nuricel, Yosle,...Agradecer especialmente por el tiempo compartido durante esta experiencia a los profesores: Ing. Jailen García González, Msc. Yoandy Pérez Villazón, Ing. Abel García Vitier, Ing. Gladis Marsi Peñalver, en fin:

A todas las personas que de una forma u otra me ayudaron no solo en la realización de la tesis sino también en el transcurso de mi carrera como profesional.

Maria del Carmen Rios Jiménez

Agradecimientos

La vida es un largo camino, donde surgen numerosos obstáculos; y por cada uno, una meta que superar. Metas tan difíciles, que es necesaria la ayuda de amigos y familiares para poderlas alcanzar.

Quiero agradecer todo el apoyo y la influencia que he recibido, principalmente la de mis padres Magnolia y Vladimir. Ellos, ejemplos de superación, que no escatimaron esfuerzos, amor ni dedicación en mis estudios, son los responsables de todos mis logros. A ellos que tanto apoyo me han dado, que orgullosos siempre han estado, quiero extenderles mi eterno agradecimiento.

A mi abuela Maria, que la vida no la dejó disfrutar de este momento a mi lado, agradecerle todo su cariño y amor donde quiera que esté.

A mi hermano Carlos Manuel (Charlitin), que siempre ha sido mi mentor, mi guía, mi profesor, mi consejero. También a su novia Cinthia, por preocuparse tanto por mi y brindarme todo su apoyo.

A mis tíos Carlito y Juanito, a mis tías Cari y Belki. A mis primos (as), a toda la familia mi eterno agradecimiento.

A mi novia Maria del Carmen (mi tata), quien ha sido durante toda la carrera mi compañera de barco, dispuesta a transitar conmigo hacia cualquier rumbo, sin importar el costo.

Agradecer a todos mis compañeros con los cuales he compartido, a aquellos que han reído y disfrutado algún momento de mi compañía. A mis compañeros de proyecto, en especial a Jailen y Jorgito que tanta ayuda me brindaron.

*A mis tutores, por toda la paciencia, dedicación y ayuda que han puesto en este trabajo.
A todos, gracias...*

Adriam Delgado Rivero

Dedicatoria

Dedicamos este triunfo especialmente a nuestros padres, a todos aquellos que estuvieron junto a nosotros brindando apoyo y seguridad en nuestros pasos, aquellos que ofrecieron su aporte y preocupación, aquellos que nos acompañaron en el trayecto y que siempre estuvieron física y espiritualmente a nuestro lado así como a la Revolución por la oportunidad que nos ha dado de formarnos como profesionales.

Resumen

En Cuba, desde abril de 2004 se está llevando a cabo una paulatina migración a aplicaciones de código abierto y software libre. El departamento de Servicios Integrales para la Migración, Asesoría y Soporte (SIMAYS), perteneciente al Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas (UCI), en su labor de llevar a cabo la migración en los Organismos de la Administración Central del Estado (OACE), ha detectado que muchas de las empresas elaboran calendarios de forma manual o mediante herramientas informáticas como por ejemplo, el asistente para calendarios de la herramienta privativa Microsoft Word. LibreOffice Writer, alternativa libre, carece de un asistente para dicha funcionalidad.

La presente investigación que lleva por título “Asistente para la elaboración de calendarios en LibreOffice Writer”, propone el desarrollo de un asistente para la elaboración de calendarios en la herramienta LibreOffice Writer con el objetivo de contribuir a mejorar la aceptación de los usuarios que están inmersos en la migración, a la nueva tecnología ofimática. Para ello se realiza un estudio del estado del arte de los diferentes sistemas y herramientas que permiten la elaboración de calendarios, incursionando en la forma en que ellos realizan dicho proceso con el objetivo de identificar elementos útiles para desarrollar el asistente. Se define la metodología de desarrollo de software SXP y se exponen los elementos fundamentales de la misma. Por otro lado se seleccionan las herramientas y tecnologías, así como el lenguaje de programación que hizo posible el desarrollo de la solución.

Finalmente se obtuvo como resultado práctico de la investigación un asistente completamente libre y multiplataforma, capaz de permitir la elaboración de calendarios en la herramienta LibreOffice Writer.

Palabras claves: Asistente, Calendarios, LibreOffice, LibreOffice Writer.

Índice

Declaración de Autoría.....	I
Agradecimientos.....	II
Agradecimientos.....	III
Dedicatoria.....	IV
Resumen.....	V
Índice.....	VI
Introducción.....	1
Capítulo #1: Fundamentación teórica.....	5
1.1 Introducción.....	5
1.2 LibreOffice. Writer y sus asistentes.....	5
1.3 Sistemas y herramientas para la elaboración de calendarios.....	6
1.3.1 Asistente para calendarios de la herramienta privativa Microsoft Word.....	7
1.3.2 Microsoft Office Outlook 2007.....	7
1.3.3 Zimbra Calendar.....	8
1.3.4 iCal.....	9
1.3.5 Google Calendar.....	10
1.3.6 Yahoo Calendar.....	11
1.3.7 Sunbird.....	12
1.5 Estándares para el intercambio de información de calendarios.....	13
1.4.1 vCalendar.....	13
1.4.2 iCalendar.....	14
1.4.3 hCalendar.....	15
1.6 Metodología de desarrollo de software SXP.....	16
1.7 Herramientas y tecnologías a utilizar en el desarrollo del asistente.....	17
1.7.1 API UNO.....	17
1.7.2 Lenguaje de programación Java.....	17
1.7.3 Entorno de Desarrollo Integrado (IDE) Netbeans.....	18
1.7.4 Herramienta de modelado Visual Paradigm UML.....	19
1.7.5 Herramientas de desarrollo colaborativo.....	19
1.7.5.1 Subversion.....	19

1.7.5.2 RapidSVN.....	20
1.12 Conclusiones del capítulo.....	20
Capítulo #2: Análisis y diseño de la solución.....	22
2.1 Introducción.....	22
2.2 Propuesta de solución.....	22
2.3 Lista de Reserva de Producto (LRP).....	23
2.4 Historias de Usuario (HU).....	24
2.4.1 HU Crear calendario semanal.....	25
2.4.2 HU Crear calendario mensual.....	25
2.4.3 HU Crear calendario anual.....	26
2.4.4 HU Gestionar tarea.....	26
2.4.5 HU Añadir tareas al calendario semanal.....	27
2.4.6 Añadir tareas al calendario mensual.....	28
2.4.7 HU Añadir tareas al calendario anual.....	28
2.4.8 HU Mostrar vista previa del calendario.....	29
2.6 Descripción de la arquitectura.....	29
2.7 Patrones de diseño empleados.....	30
2.7.1 Patrones GRASP.....	31
2.7.2 Patrones GOF.....	31
2.8 Modelo de Diseño.....	33
2.8.1 Diagrama de clases del diseño.....	33
2.8.2 Diagrama de paquetes.....	34
2.9 Conclusiones del capítulo.....	35
Capítulo #3: Implementación y pruebas de la solución.....	36
3.1 Introducción.....	36
3.2 Plan de liberaciones.....	36
3.3 Tareas de Ingeniería (TI).....	37
3.3.1 TI asociadas a la HU Crear calendario semanal.....	37
3.3.2 TI asociada a la HU Crear calendario mensual.....	38
3.3.3 TI asociada a la HU Crear calendario anual.....	38
3.3.4 TI asociadas a la HU Gestionar tarea.....	39

3.3.5 TI asociada a la HU Añadir tareas al calendario semanal.....	41
3.4 Estándar de codificación.....	41
3.5 Integración del asistente con LibreOffice.....	42
3.5.1 Uso de la clase CallWizard.....	42
3.5.2 Uso de API UNO.....	43
3.6 Pruebas.....	46
3.7 Conclusiones del capítulo.....	54
Conclusiones generales.....	55
Recomendaciones.....	56
Referencias bibliográficas.....	57
Bibliografía.....	61
Glosario de términos.....	65
Anexos.....	68
Anexo 1: Tendencia al uso de herramientas gestoras de calendario.....	68
Anexo 2: Tendencia al uso de herramientas de control de versiones.....	68
Anexo 3: Interfaz correspondiente a la HU-01.....	69
Anexo 4: Interfaz correspondiente a la HU-02.....	69
Anexo 5: Interfaz correspondiente a la HU-03.....	70
Anexo 6: Interfaz correspondiente a la HU-04.....	70
Anexo 7: Interfaz correspondiente a la HU-05.....	71
Anexo 8: Interfaz correspondiente a la HU-06.....	72
Anexo 9: Interfaz correspondiente a la HU-07.....	73
Anexo 10: Interfaces correspondientes a la HU-08.....	74
Anexo 11: Estándar de codificación utilizado.....	77

Introducción

A medida que surgen nuevas tecnologías en el mundo, países como Alemania, Francia, Inglaterra, España, Argentina, Brasil y Venezuela [1], optan por la migración parcial o total a tecnologías y estándares abiertos. Cuba no está ajena a los sucesos del mundo, y en abril del 2004 el Consejo de Ministros adoptó el Acuerdo 084/2004, donde indicaba al Ministerio de la Informática y las Comunicaciones (MIC) ordenar la paulatina migración a aplicaciones de código abierto y software libre, en las empresas cubanas [2].

En la actualidad el 95% de las computadoras de la Aduana General de la República de Cuba prestan servicio sobre plataforma de software libre [3]. Han sido migradas otras empresas entre las que se encuentran el Ministerio de Salud Pública (MINSAP), el Ministerio de la Informática y las Comunicaciones (MIC), el Ministerio de Educación Superior (MES), la Empresa de Telecomunicaciones de Cuba (ETECSA), el Ministerio de Cultura (MINCULT), el Centro de Cibernética Aplicada a la Medicina (CECAM), entre otras.

En las experiencias obtenidas por el departamento SIMAYS, en su labor de llevar a cabo la migración en los OACE, ha detectado que muchas de las empresas antes mencionadas elaboran los calendarios de forma manual o mediante la utilización de herramientas informáticas como por ejemplo, el asistente para calendarios de la herramienta privativa Microsoft Word. LibreOffice Writer, su alternativa libre, carece de un asistente para dicha funcionalidad, siendo esto un elemento que trae como consecuencia que los usuarios que comienzan a utilizar LibreOffice tengan que modificar la manera en que realizaban la planificación y por ende se resistan al cambio a la nueva tecnología ofimática.

Partiendo de la problemática anterior, se llega al siguiente **problema científico**: ¿Cómo proveer a la herramienta LibreOffice Writer de un asistente para la elaboración de calendarios? Definiendo como **objeto de estudio** de la investigación “El proceso de elaboración de calendarios mediante herramientas informáticas” enmarcado en el **campo de acción** “Proceso de

desarrollo de asistentes en la herramienta LibreOffice Writer”. Para dar solución al problema se define como **objetivo general**, “Desarrollar un asistente para la elaboración de calendarios en la herramienta ofimática LibreOffice Writer”.

Se pretende cumplir el objetivo general a partir los siguientes **objetivos específicos**:

- Realizar un estudio de la bibliografía actual relacionada con los sistemas y herramientas que se utilizan para la elaboración de calendarios.
- Identificar la metodología de desarrollo de software que guiará el proceso de desarrollo del asistente así como las herramientas y tecnologías para su implementación.
- Desarrollar el asistente para la elaboración de calendarios en LibreOffice Writer.
- Probar el asistente desarrollado.

Los objetivos específicos anteriores se concretan en las siguientes **tareas de investigación**:

- Estudio del estado del arte acerca de la elaboración de calendarios.
- Análisis de la metodología, lenguaje y herramientas a utilizar en el desarrollo del asistente.
- Definición de la arquitectura y las funcionalidades para el asistente.
- Diseño de la solución propuesta.
- Implementación de las funcionalidades del asistente.
- Diseño y ejecución de pruebas al asistente.

Se plantea la siguiente **idea a defender**: “El desarrollo de un asistente para la elaboración de calendarios en la herramienta LibreOffice Writer puede contribuir a mejorar la aceptación por parte de los usuarios inmersos en el proceso de migración, a la nueva tecnología ofimática”.

Los métodos teóricos utilizados fueron el Analítico - Sintético y el Inductivo - Deductivo. El primero de estos, se enfoca en determinar la esencia de los fenómenos, los rasgos que los caracterizan y los distinguen. El mismo se empleó en el estudio de los sistemas y herramientas que permiten la elaboración de calendarios; lo que permitió comprender su funcionamiento e identificar elementos que resultaron útiles para desarrollar el asistente.

Por otra parte, el método Inductivo – Deductivo permite llegar a proposiciones generales a partir de hechos aislados que confirman la teoría, o a partir de estas teorías arribar a conclusiones sobre casos particulares que se verifican en la práctica. Este método permitió inferir a partir de los sistemas y herramientas gestoras de calendario existentes, la forma en que ellos llevan a cabo el proceso de elaboración de los mismos.

La técnica empleada para la recogida de la información, fue la tormenta de ideas. La misma fue realizada a especialistas del departamento SIMAYS; donde se obtuvieron diferentes puntos de vista e ideas que contribuyeron al desarrollo de la aplicación.

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

Capítulo 1: Fundamentación teórica: Se ofrecerá una visión general de los sistemas y herramientas gestoras de calendario de mayor demanda en el mundo. Se detallarán los estándares definidos a nivel internacional para lograr el intercambio de información entre calendarios. Además, se describirán las herramientas, el lenguaje de programación y la metodología de desarrollo de software a utilizar en el desarrollo de la solución propuesta.

Capítulo 2: Análisis y diseño de la solución: Se definirán los requisitos funcionales y no funcionales en la Lista de Reserva de Producto (LRP). Se reflejarán en las Historias de Usuarios (HU) las funcionalidades del asistente, con sus prototipos de interfaces gráficas de usuario. Se definirá la arquitectura y los patrones de diseño a utilizar. Además se realizará el modelo de diseño de la aplicación donde se elaborará el diagrama de clases del diseño y el diagrama de paquetes.

Capítulo 3: Implementación y pruebas de la solución: Se mostrará el plan de liberaciones que se llevará a cabo para darle cumplimiento a las HU reflejadas en el capítulo anterior. Se describirán las tareas de ingeniería. Se definirá el estándar de código a utilizar en el desarrollo de la solución, además de exponer las pruebas realizadas al asistente para garantizar su calidad.

Capítulo #1: Fundamentación teórica.

1.1 Introducción

El presente capítulo tiene como objetivo, presentar los elementos teóricos que sustentan la investigación, comenzando por describir elementos generales relacionados con el objeto de estudio. Además se presenta un resumen de las características de algunos sistemas y herramientas gestoras de calendario, así como de los estándares definidos a nivel internacional para lograr el intercambio de información entre calendarios. Por último, se describen las herramientas, el lenguaje de programación y la metodología de desarrollo de software a utilizar en el desarrollo de la solución propuesta.

1.2 LibreOffice. Writer y sus asistentes

LibreOffice es una suite ofimática libre y gratuita, creada por la fundación The Document Foundation como una bifurcación de la suite OpenOffice.org en octubre de 2010 [4]. Está disponible en más de 30 idiomas y para los principales sistemas operativos, incluyendo Microsoft Windows, Mac OS X y Linux (Debian, Ubuntu, Fedora, Mandriva, Suse,...) [5].

Incluye aplicaciones tales como: Calc, Impress, Base, Math, Draw y Writer. Esta última constituye el procesador de texto de LibreOffice. Permite proteger documentos con contraseña, guardar versiones del mismo documento, insertar imágenes, tablas, símbolos, fórmulas, gráficos, hiperenlaces, marcadores, formularios, etc. Admite firmas digitales y permite exportar archivos de texto a los formatos PDF y HTML.

LibreOffice Writer incluye varios asistentes que le brindan al usuario una vía fácil e intuitiva para la elaboración de cartas, faxes, agendas, presentaciones interactivas. El asistente para páginas web permite mantener un sitio web en un servidor de Internet. Contiene un asistente que permite convertir las cantidades monetarias de los documentos de LibreOffice Calc y de los campos y tablas de documentos de LibreOffice Writer en euros. También incluye un asistente para convertir documentos de Microsoft Word, Excel o PowerPoint en documentos OpenDocument (ODF).

El Formato de Documento Abierto para Aplicaciones Ofimáticas de OASIS (*en inglés, OASIS Open Document Format for Office Applications*), también referido como formato OpenDocument (ODF), es un formato de archivo abierto y estándar para el almacenamiento de documentos ofimáticos tales como hojas de cálculo, textos, gráficas y presentaciones. Fue publicado como estándar OASIS el 1 de mayo de 2005 aunque no fue hasta el 30 de noviembre de 2006 que fue aprobado por las organizaciones ISO/IEC como estándar internacional ISO/IEC 26300:2006 Open Document Format for Office Applications (OpenDocument) [6].

OpenDocument está basado en un lenguaje de esquema XML, inicialmente implementado en la suite ofimática OpenOffice.org. Las extensiones al nombre de archivo identificativas de los archivos OpenDocument incluye: odt para documentos de texto, ods para hojas de cálculo, odp para presentaciones, odg para gráficos y odb para bases de datos. Es utilizado tanto en aplicaciones de software libre como propietario. También lo utilizan varias suites ofimáticas como LibreOffice, OpenOffice.org y Microsoft Office (2007 y 2010), además de aplicaciones individuales como procesadores de texto, programas de manejo de hojas de cálculo, presentaciones y datos.

Con la presente investigación se pretende desarrollar un asistente para la elaboración de calendarios en LibreOffice Writer, lo que facilitará la planificación de las actividades que se realizan en empresas o entidades.

Para enriquecer el presente trabajo de diploma se hace necesario el estudio de los sistemas y herramientas gestoras de calendario más utilizadas en el mundo. Para la selección de estas herramientas se utilizó el Google Trends mediante el cual se obtuvo un diagrama de los grados de utilización de las diferentes herramientas gestoras de calendario a nivel mundial desde el año 2004 hasta la fecha (**Ver Anexo I**). Además se tendrán en cuenta otras herramientas que se utilizan en Cuba para gestionar calendarios.

1.3 Sistemas y herramientas para la elaboración de calendarios

En el mundo existen varios sistemas y herramientas para elaborar calendarios que pueden estar relacionadas o tributando a sistemas de planificación para la gestión de Capital Humano, los cuales logran llevar a cabo una planificación tanto laboral como personal, registrando datos históricos de tareas llevadas a cabo, no realizadas, aplazadas, etc., siendo estos datos de gran interés para la evaluación de trabajadores y confección de sus nóminas salariales.

1.3.1 Asistente para calendarios de la herramienta privativa Microsoft Word

El asistente para calendario integrado a la herramienta privativa Microsoft Word es una de las herramientas informáticas que se utilizan para elaborar calendarios en las empresas e instituciones cubanas.

Mediante este asistente los usuarios pueden crear varios estilos de calendario diferente. El mismo proporciona una forma rápida y eficaz para producir un calendario genérico. El programa ofrece, por defecto, algunas plantillas para calendarios, aunque también ofrece la posibilidad de descargar más plantillas, diseños y accesorios en forma gratuita, desde el sitio de internet de esta empresa.

El asistente permite seleccionar el tipo de calendario y el estilo deseado para el mismo. Estas plantillas también incluyen calendarios especiales, como por ejemplo el calendario del año académico, del año fiscal, además de un calendario lunar. Todas estas plantillas son personalizables, se pueden cambiar las imágenes, colores, tipografías y contenidos, al gusto del usuario o cliente. Una vez creado y personalizado el calendario, el programa da la opción de imprimirlo o guardarlo como documento.

Una de las desventajas que posee este asistente es que para obtener acceso al mismo se tiene como requisito indispensable establecer una conexión a internet.

1.3.2 Microsoft Office Outlook 2007

Microsoft Outlook [7] es un programa de organización ofimática y cliente de correo electrónico de Microsoft el cual forma parte de la suite Microsoft Office. Con el paso de los años ha presentado varias versiones dándole mejoramiento a la programación de calendarios del cliente o usuario,

con la novedad en su versión del año 2007 de permitir el intercambio de información de calendarios, utilizando los estándares internacionales para la comunicación entre calendarios de los cuales se abordará en el epígrafe 1.5.

Outlook permite la gestión de varios calendarios. Además permite compartir los calendarios dando la opción de publicarlos en la web, enviarlo por correo electrónico, salvarlo en un fichero externo (extensión .ics) con el objetivo de trasladarlo o ejecutarlo con otros gestores. Además de estas opciones brinda la posibilidad de programar eventos, tareas, citas y lanzar convocatorias de grupo a las cuales permite establecer una prioridad e identificarlas con colores.

Para añadir un evento hay que tener en cuenta que puede hacerse desde la vista de calendario o desde una opción general recogiendo los datos del asunto, descripción, fecha y hora de inicio, fecha y hora de fin del mismo, además de especificar la categoría o ámbito y la prioridad.

También permite la programación de una alarma con tiempo de antelación para el aviso de la ejecución próxima del evento que se añadió. En esta acción se puede establecer una periodicidad; de la alarma hay que especificar la fecha y hora de inicio, fecha y hora de fin, esta última siendo opcional para el usuario; además brinda la posibilidad de seleccionar la frecuencia en la que se repetirá el evento la cual está dividida en una repetición diaria, semanal, mensual, anual y cada una de estas con sus especificaciones respectivamente. Se puede también seleccionar la cantidad de veces que se quiere repetir el evento, característica opcional para el usuario, a su vez se le puede detallar la duración de la misma.

1.3.3 Zimbra Calendar

Zimbra Collaboration Suite (ZCS) nace como un servidor y cliente de mensajería y colaboración, diseñado para proveer herramientas de correo electrónico, calendarios, mensajería instantánea y documentos compartidos; facilitando la interacción de usuarios finales y administradores de sistemas. La empresa creadora de ZCS, Zimbra Inc., desarrolla el software desde 2005, y fue comprada por Yahoo! en 2007. Hoy en día Zimbra es propiedad de VMWare el cual la compró después de la decisión de venta de Yahoo.

El gestor de calendario Zimbra Calendar [8] está estructurado por múltiples vistas, en cada una da la opción de agregar eventos en la hora o día específico y la opción de imprimir. La primera de las vistas es por día, la cual está dividida por horas, con la posibilidad de navegar hacia el día anterior y siguiente. La segunda vista es la de semana, la cual está dividida por días con la posibilidad de navegar hacia la semana anterior y siguiente. La tercera vista es por mes; esta es una vista de calendario clásica que posibilita navegar hacia el mes anterior y siguiente. La cuarta vista es la de semana laboral; esta muestra una vista de los días de la semana laborables. La próxima vista es la vista por agenda, la cual muestra todas las actividades contenidas en la agenda y por último la vista de hoy que muestra las actividades planificadas para el día actual.

Zimbra calendar brinda la posibilidad de añadir cita, para la cual hay que tener en cuenta campos tales como asunto, ubicación, asistentes, descripción y hora que está definido por evento, todo el día, fecha y hora de comienzo, fecha y hora final, repetir y recordatorio. Otra de las opciones de esta agenda es añadir tareas en la cual hay que definir el asunto, ubicación, prioridad, estado, fecha de inicio, fecha de entrega, porcentaje completado y descripción.

1.3.4 iCal

iCal [9] es una aplicación de calendario personal hecha por Apple Inc. que se ejecuta en el sistema operativo Mac OS X. Está integrado con MobileMe (actualmente iCloud) [10], por lo que los calendarios pueden ser compartidos por Internet. También se pueden compartir calendarios a través del protocolo WebDAV.

Esta aplicación permite a los usuarios administrar múltiples calendarios además de crear y administrar actividades grupales o individuales. También posibilita publicar calendarios en Internet. Permite la actualización automática vía Internet de su calendario para mantenerse al día con fechas laborales y eventos. Permite llevar los calendarios a cualquier lugar dentro de los teléfonos celulares que operan con Bluetooth, dispositivos Palm OS e iPod. Con iCal se pueden ver todos los calendarios en una sola ventana o elegir sólo los calendarios que desee el usuario. Además brinda la posibilidad de avisar de un evento de dos maneras: una, mediante una alarma programada y la otra, por correo electrónico.

1.3.5 Google Calendar

Google Calendar [11] es una agenda y calendario electrónico desarrollado por Google. Permite sincronizarlo con los contactos de Gmail de manera que se pueda invitarles y compartir eventos. Está disponible desde el 13 de abril de 2006. Los usuarios no están obligados a tener una cuenta de Gmail, sí deben disponer de una cuenta de usuario de Google (*Google Account*) para poder usar el software. Google Calendar pasó a ser un producto terminado el 7 de julio de 2009. La interfaz de Google Calendar es similar a otras herramientas gestoras de calendario para escritorio tales como Microsoft Outlook o iCal para Mac OS X.

Este gestor de calendario está estructurado por cuatro vistas. La primera vista, por día, muestra el día dividido en horas con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista dando la opción de acceder a los detalles de este y brinda la posibilidad de navegar al próximo o día anterior. La segunda vista, por semana, muestra la semana dividida en días y estos en horas, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este, además posibilita navegar a la próxima o semana anterior. La tercera vista, por mes, muestra la vista clásica de un calendario, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este, además brinda la posibilidad de navegar al próximo o mes anterior. La cuarta y última vista, muestra 4 días divididos en horas donde el primero es el día actual, con la posibilidad de agregar un nuevo evento de forma rápida directamente desde la vista, dando la opción de acceder a los detalles de este y navegar a los 4 próximos o días anteriores. En cada una de las vistas brinda la opción de imprimir la porción de programación mostrada.

Google Calendar ofrece múltiples opciones entre las que se encuentra crear evento. Para poder llevar a cabo esta acción hay que tener en cuenta el nombre, fecha y hora de inicio, duración, recurrencia la cual está limitada porque no permite hacer programaciones de alta complejidad, fecha de fin la cual es opcional para los usuarios, lugar, descripción, recordatorio el cual estará determinado por distintas vías y minutos de antelación y por último la asignación a otras personas.

Otras de las opciones que ofrece es crear tarea, mostrando una ventana con un listado de tareas con la posibilidad de agregar tareas de forma rápida, modificar una seleccionada o especificar informaciones opcionales, así como visualizarla en las vistas del calendario en un área distinta a la de los eventos. Este gestor brinda la posibilidad de exportar e importar calendarios externos a través del estándar iCalendar y permite visualizar varios calendarios a la vez identificándolos por colores. Esta agenda muestra eventos próximos.

1.3.6 Yahoo Calendar

Yahoo Calendar [12] es una de las opciones que brinda la mensajería Yahoo. Este trae asociado varias funcionalidades tales como añadir evento, funcionalidad para la cual el usuario o cliente debe especificar el nombre del evento, tipo de evento, fecha de inicio, hora de inicio, duración, ubicación, notas, compartir el cual puede ser privado o público, establecer una recurrencia limitada, especificar fecha fin y asignar calendario. Estos dos últimos son opcionales para los usuarios. Además permite crear recordatorios, los cuales pueden ser por correo electrónico, mensajería instantánea o SMS.

Otra de las opciones que permite esta aplicación es añadir evento rápido en la cual se debe especificar nombre del evento, fecha y hora de inicio. Este gestor permite añadir tareas en la que debemos tener en cuenta título, fecha tope que puede ser opcional o sin definir, prioridad que está definida de 1 a 5, estado de la tarea en el cual se define si la tarea está en proceso, finalizada o no iniciada, compartir que puede ser privado o público y por último si desea añadir una nota acerca de la misma.

Yahoo Calendar tiene integrado varios tipos de búsquedas tales como búsqueda rápida, búsqueda avanzada, pudiendo hacerse por un filtrado de título, nota y tipo de evento y búsqueda rápida de fecha la cual busca una fecha de forma más rápida con un calendario pequeño separado por meses con sus días. Este calendario da la opción de decidir qué día empieza la semana para cada usuario que haga uso del mismo.

El calendario de Yahoo facilita la forma de visualizar la información ya que está estructurado por vistas las cuales dan la opción de agregar eventos en la hora o día específico y la opción de imprimir. La primera vista, por día, está dividida por horas brindando la posibilidad de navegar hacia el día anterior y siguiente. La segunda vista, por semana, está dividida por días brindando la posibilidad de navegar hacia la semana anterior y siguiente. La tercera vista, por mes, es la vista de calendario clásica que posibilita navegar hacia el mes anterior y siguiente. La cuarta vista, por año, contiene todas las vistas de mes en una sola página dando la posibilidad de acceder a una semana y navegar hacia el año anterior y siguiente. La vista, por evento, muestra un listado con todos los eventos dando la opción de filtrar por eventos pendientes o pasados y por último la vista por tareas, la cual muestra un listado con todas las tareas dando la opción de filtrar por tareas sin acabar o terminadas. Además de todas las opciones que tiene integrado este gestor permite modificar la zona horaria según desee el usuario que haga uso del mismo.

1.3.7 Sunbird

Sunbird [13] es un programa de agenda basado en el “Calendar” de Mozilla. Es una aplicación independiente basada en el módulo de calendario de Mozilla y que cumple con holgura en sus funciones de agenda, lista de tareas y calendario con alarmas.

En su configuración básica el usuario dispone de tres ventanas. La ventana principal muestra una vista general de los eventos por día, semana o mes. Dos ventanas más pequeñas permiten visualizar las tareas definidas y un calendario.

Los eventos se definen mediante una interfaz muy simple. Como en cualquier agenda, es posible asignarle a cada evento un período de tiempo de validez (en días y horas), establecer si se repetirá en el futuro y definirlo dentro de una categoría. Se puede agregar una descripción breve del evento y una lista de direcciones de correo electrónico de las personas interesadas. Cada evento puede definirse como público o privado, con su prioridad y su estado (tentativo, confirmado o cancelado). Es posible configurar una alarma para recordar cada evento, en forma de un sonido o una ventana que aparecerá en la pantalla.

La definición de las tareas a realizar es idéntica a la mencionada para los eventos. La única diferencia es que se puede asignar a cada tarea un porcentaje de progreso, que indica cuanto nos falta para terminarla.

El usuario puede definir diferentes tipos de eventos, cada uno con un color que lo identifica. Se pueden editar los tipos predefinidos, cambiándoles el nombre y el color con el cual aparecerán en la agenda.

Se pueden manejar diferentes agendas dentro del mismo programa. Esto puede ser útil para mantener los eventos y tareas familiares separados de los del trabajo. Dado que es posible compartir la agenda personal con quienes se desee, se puede evitar que los compañeros de trabajo tengan acceso a la agenda personal, mientras que pueden consultar los eventos y tareas laborales que tengan en común.

Los archivos de Sunbird utilizan el estándar abierto iCalendar. Esto permite intercambiar agendas con cualquier programa que utilice el mismo sistema. En el caso de Outlook, no es posible leer las agendas en el formato que utiliza este programa. Sin embargo se puede grabar una agenda en formato iCalendar y luego importarla a Sunbird.

1.5 Estándares para el intercambio de información de calendarios

En la actualidad existe un alto número de herramientas gestoras de calendario que en sus inicios no tenían comunicación entre sí; con el avance de la tecnología y la necesidad de intercambiar información entre calendarios, se han creado diferentes estándares definidos a nivel internacional que permiten la comunicación entre estas herramientas. En este epígrafe se muestra una descripción de cada uno de estos estándares.

1.4.1 vCalendar

vCalendar es un formato estándar para el intercambio de información personal. Es aplicable a una amplia variedad de calendarios y es útil en el intercambio de información a través de una amplia gama de métodos de transporte [14].

vCalendar se puede utilizar para el transporte de información de calendario personal entre las aplicaciones de su elección ya sean administradores de información personal, sistemas de correo o navegadores web. Una vez que la información está en su sistema, puede compartirlo con otra aplicación vCalendar.

La especificación vCalendar se adecua como formato de intercambio entre las aplicaciones o sistemas. El formato se define independiente del método particular utilizado para su transporte. El transporte de este intercambio podría ser un sistema de archivos, de punto a punto de redes telefónicas públicas, el transporte por cable de red, o algún tipo de transporte sin cables.

Antes de la introducción de la especificación de vCalendar, los usuarios de estas aplicaciones solían tener una re-clave de la información original, transcribiendo a menudo desde el primer día los planificadores de papel, pedazos de papel o mensajes de correo electrónico. Con la llegada de la especificación de vCalendar, esta información se puede intercambiar de forma automatizada y consistente.

La especificación vCalendar captura la fecha y hora. La especificación de todos esos valores en un vCalendar se definen en términos de la norma internacional para la representación de fechas y horas, ISO8 8601 [15].

El uso de este estándar garantiza la comunicación en todas partes del mundo, incluso los que pueden utilizar otra fecha, la hora y los formatos de zona horaria. El formato utilizado en la especificación de vCalendar para la identificación de idiomas y juegos de caracteres se basa en las actuales normas ISO y IETF (*Internet Engineering Task Force*).

El vCalendar tiene aplicación directa a los usuarios de forma que pueda utilizar la información de calendario en la red de trabajo. Organizaciones en Internet también pueden utilizar el formato vCalendar para automatizar la publicación de la información del calendario de sus páginas principales.

1.4.2 iCalendar

iCalendar es un estándar (RFC 5545) [16], creado por el Comité de Calendario y Agendas del IETF (en inglés, *Internet Engineering Task Force*) encabezado por Anik Ganguly de Open Text Inc., y redactado por Frank Dawson de Lotus y Derik Stenerson de Microsoft Corporation.

iCalendar se basa en el trabajo previo de vCalendar 1.0. Es un estándar para el intercambio de información de calendarios. También se conoce como "iCal", debido al nombre del programa de Apple, que fue la primera aplicación en implementarlo [17]. iCalendar permite a los usuarios enviar solicitudes de reuniones y tareas a otros usuarios mediante el correo electrónico o compartir archivos con la extensión.ics.

Está diseñado para ser independiente del protocolo de transporte. La reaparición de la nueva propiedad de identificación permite a los casos individuales de calendario estar unidas entre sí y la recurrencia con gran poder de manejo de excepciones. iCalendar cuenta con el apoyo adecuado para las zonas de tiempo y puede representar exactamente las entradas en la hora local, y la hora local con una zona de tiempo especificado. Esto es particularmente importante para repetir entradas que pueden abarcar un día de ahorro. iCalendar apoya normas como repetir con una frecuencia de segundos, minutos y horas. Este tipo de normas de repetición no se puede representar en vCalendar.

La especificación iCalendar, mejora el nivel de interoperabilidad entre los distintos calendarios y la programación de aplicaciones y sistemas.

1.4.3 hCalendar

hCalendar (abreviatura de **HTML iCalendar**) es un estándar de microformato pensado para calendarios y eventos. Constituye una representación exacta de iCalendar, expresada en estilo XHTML [17]. Es válido para HTML, XHTML, XML y RSS. Es un estándar para mostrar una representación semántica en XHTML de la información de un calendario en formato iCalendar acerca de un evento, en páginas web, usando las clases de HTML.

Los microformatos son un conjunto de convenciones abiertas basadas en los datos existentes y las normas ampliamente adoptadas para incrustar la semántica para los dominios de problemas específicos legibles de documentos XHTML. Estos constituyen a lo que se ha denominado la web semántica en minúsculas. [18]

Los microformatos se han diseñado con el objetivo de evitar dependencias innecesarias de los recursos externos (como la URL), que pueden ser fácilmente incorporadas en cualquier contenido XHTML. También es fácil de componer microformatos usando otros microformatos.

Debido a que los microformatos aprovechan XHTML, pueden ser dirigidos y manipulados desde JavaScript y DOM. Una aplicación de esto puede verse en la extensión de Firefox de Mozilla, en las colas de exportación que apoya la detección y la exportación de los convenios de microformatos de una página web.

1.6 Metodología de desarrollo de software SXP

A partir de la necesidad de crear un asistente para LibreOffice Writer que garantice el proceso de elaboración de calendarios, se empleará la metodología ágil de desarrollo SXP para guiar la realización del mismo, pues es la que se utiliza en el proyecto SIMAYS al cual pertenecen los autores de la presente investigación.

SXP es una metodología desarrollada en la UCI que reúne las buenas prácticas de XP y SCRUM; ambas metodologías ágiles. SXP mejora la actividad productiva aumentando el nivel de responsabilidad de los miembros del equipo y ayudando al líder del proyecto a tener un mejor control del mismo. Su particularidad es tener como parte del equipo al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Esta metodología consta de 4 fases:

Fase de Planificación-Definición: En esta primera fase es donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto. Cuenta con tres tareas principales: ingeniería de sistemas o de información, planificación del proyecto y análisis de los requisitos.

Fase de Desarrollo: En esta fase se implementa el sistema en una serie de iteraciones listo para entregar al usuario final. Cuenta con tres tareas fundamentales: diseño del software, generación de código y prueba del software.

Fase de Entrega: En esta fase es donde se despliega y se pone en marcha el software.

Fase de Mantenimiento: En esta última fase se realiza el soporte para el usuario final.

SXP es ideal para proyectos de corta duración con requisitos cambiantes o no bien definidos, donde prevalezca la retroalimentación entre el cliente y el equipo de trabajo. El desarrollo con SXP se realiza en iteraciones cortas (*sprints*) a lo largo de 3 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 [20].

1.7 Herramientas y tecnologías a utilizar en el desarrollo del asistente

1.7.1 API UNO

La API UNO [19] (del inglés *Application Programming Interface of Universal Network Objects*), es el modelo de componentes utilizado en las suites ofimáticas OpenOffice y LibreOffice. Está basado en componentes de interfaz y diseñado para ofrecer interoperabilidad entre aplicaciones escritas en diferentes lenguajes de programación, arquitecturas de hardware y plataformas informáticas. UNO está publicado bajo los términos de la licencia GNU Lesser General Public License (LGPL) como software libre y de código abierto, y disponible para Linux, Solaris, Windows, Mac OS X, entre otros.

Se decide utilizar UNO porque es el modelo de componentes que utiliza LibreOffice para lograr interoperabilidad entre sus aplicaciones escritas en diferentes lenguajes de programación. Permitirá integrar el asistente para la elaboración de calendarios a la suite ofimática LibreOffice, específicamente a la aplicación LibreOffice Writer.

1.7.2 Lenguaje de programación Java

Java es un lenguaje de programación orientado a objetos, que permite a los programadores realizar aplicaciones de escritorio o web. Se caracteriza por ser un lenguaje sencillo y robusto. Destaca por ser un lenguaje de código abierto y multiplataforma, características por las cuales ha logrado una gran expansión por todo el mundo. En la actualidad incluye un gran número de librerías para múltiples trabajos como: el trabajo con la red, tratamiento de excepciones, hilos para el procesamiento concurrente, entre otras. Java y Python constituyen los únicos lenguajes en los que hasta el momento se han desarrollado asistentes (*wizards*) para LibreOffice.

El elemento principal por el que se decide utilizar Java como lenguaje de programación es que el mismo constituye el lenguaje para el cual UNO (modelo de componentes seleccionado anteriormente) presenta un soporte total (**Ver Tabla 1**). Incidentalmente es el lenguaje que mejor dominan los autores del presente trabajo de diploma.

Lenguaje Programación	UNO Scripting	UNO Componentes	Implementación con documentos de OpenOffice.org
C	Tal vez en el futuro	Tal vez en el futuro	No
C++	Sí	Sí	No
Java	Sí	Sí	Sí
Python	Tal vez en el futuro (en desarrollo)	Tal vez en el futuro (en desarrollo)	Tal vez en el futuro
StarBasic	Sí	No	Sí
Automatización OLE (Win32 únicamente)	Sí	Tal vez en el futuro	Tal vez en el futuro

Tabla 1: Soporte de UNO para lenguajes de programación.

1.7.3 Entorno de Desarrollo Integrado (IDE) Netbeans

Netbeans es un IDE que permite desarrollar aplicaciones de escritorio, web y móviles de una manera rápida y sencilla con lenguajes como Java, C/C++, PHP, entre otros. Es una herramienta libre y de código abierto. Posee una gran comunidad de usuarios y desarrolladores en todo el mundo [21]. Netbeans IDE es fácil de instalar y de uso instantáneo; además se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X.

Se decidió utilizar Netbeans pues este IDE posee compatibilidad total con el lenguaje de programación seleccionado (Java).

1.7.4 Herramienta de modelado Visual Paradigm UML

Visual Paradigm permite representar todo tipo de diagramas UML para las distintas fases por la que transita un software en desarrollo como la captura de requisitos, análisis, diseño e implementación. Esta poderosa herramienta facilita la generación de código, lo que facilita el trabajo de los desarrolladores y además está disponible en múltiples plataformas. Permite configurar las líneas de redacción, el modelado de base de datos, el modelado de requisitos, el modelado del proceso de negocio, la interoperabilidad, la generación de documentación y la generación de código base para diferentes lenguajes de programación como Java, C# y PHP además de permitir la integración con entornos de desarrollo integrado (IDE). [22]

Se decide utilizar Visual Paradigm como herramienta de diseño UML pues permitirá modelar los artefactos que se generen durante todo el desarrollo de la solución propuesta en el presente trabajo de diploma. Esta herramienta, aunque no es libre constituye la mejor opción para realizar las tareas antes descritas. Se integrará con el IDE seleccionado (Netbeans) para facilitar el modelado de las clases y será utilizada en su versión para GNU/Linux.

1.7.5 Herramientas de desarrollo colaborativo

Se hará uso de las herramientas, control de versiones Subversion y su plataforma visual RapidSVN.

1.7.5.1 Subversion

Subversion es un sistema de control de versiones creado en el año 2000 por la comunidad y desarrolladores de CollabNet, Elego, VisualSVN, WANdisco. Fue diseñado para sustituir a CVS (*Concurrent Version System*). Se le conoce también como svn por ser el nombre de la herramienta utilizada en la línea de comando [23]. Dentro de sus principales características se encuentran las siguientes:

- Mantiene versiones no sólo de archivos, sino también de directorios.

- Es un sistema de control de versiones libre bajo una licencia de tipo Apache/BSD.
- Es un sistema general que se puede utilizar para administrar cualquier conjunto de ficheros.
- Los archivos versionados no tienen un número de revisión independiente.
- Hace un mejor uso del ancho de banda, ya que en las transacciones se transmiten sólo las diferencias y no los archivos completos.
- Realiza el seguimiento de las versiones de proyectos completos.

1.7.5.2 RapidSVN

Es una plataforma visual para el sistema Subversion escrito en 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 éste 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. [24]

Los dos software analizados se seleccionaron debido a las características que presentan, permitiendo tener siempre asegurada las últimas copias y cambios que se realicen en los documentos generados de la investigación así como también del código que se desarrolla. El Subversion o SVN es uno de los sistemas de control de versiones más utilizados actualmente (**Ver Anexo 2**) y el RapidSVN es uno de los clientes más completos con el que se cuenta para su uso.

1.12 Conclusiones del capítulo

A partir del estudio realizado sobre los sistemas y herramientas que se utilizan en Cuba y el mundo para elaborar calendarios se pudo inferir en la forma en que estas herramientas realizan el proceso de elaboración de calendarios, lo que permitió obtener varias características que se tendrán en cuenta para el desarrollo de la solución propuesta las cuales se muestran a continuación:

- Elaborar calendarios mediante la utilización de plantillas.
- Visualizar varios tipos de calendarios.
- Gestionar tareas.

Al concluir el análisis sobre los estándares definidos a nivel internacional para lograr la comunicación entre calendarios, se propone el uso del estándar iCalendar, para una futura integración de la herramienta a desarrollar con otros sistemas y herramientas gestoras de calendario. Esta propuesta se fundamenta con el hecho de que este estándar ha sido implementado en una variedad de herramientas gestoras de calendario incluyendo iCal de Apple, Mozilla Calendar (incluyendo Mozilla Sunbird), Google Calendar, Chandler, Lotus Notes, ScheduleWorld, KOrganizer, Mulberry, Evolution de Novell, Kronolith, Simple Groupware, Hotmail Calendar, Nuvvo, Microsoft Outlook y Trac [17]; elemento que facilitará el intercambio de información entre la aplicación a desarrollar y demás herramientas gestoras de calendario que utilicen este estándar.

Se empleará la metodología ágil de desarrollo SXP para guiar el proceso de desarrollo del asistente. Se seleccionaron las siguientes herramientas y tecnologías: API UNO para lograr la interoperabilidad entre el asistente a desarrollar y la suite ofimática LibreOffice, lenguaje de programación Java, IDE Netbeans, Visual Paradigm UML como herramienta de modelado, Subversion para el control de versiones y RapidSVN como herramienta para administrar el Subversion.

Capítulo #2: Análisis y diseño de la solución.

2.1 Introducción

Una vez definida la metodología, herramientas, lenguaje y tecnologías a utilizar para el desarrollo de la solución, se debe proceder a analizar la misma. Para esto, en el presente capítulo se realizará la propuesta de solución, se definirán los requisitos funcionales y no funcionales en la Lista de Reserva de Producto (LRP), se reflejarán en las Historias de Usuarios las funcionalidades del asistente con sus prototipos de interfaces gráficas de usuario. Se describirá la arquitectura y se definirán los patrones de diseño a utilizar. Además se realizará el modelo de diseño de la aplicación donde se elaborará el diagrama de clases del diseño y el diagrama de paquetes.

2.2 Propuesta de solución

El presente trabajo de diploma propone desarrollar un asistente para la elaboración de calendarios en la herramienta LibreOffice Writer. Este asistente les permitirá a los usuarios elaborar calendarios semanales, mensuales y/o anuales mediante la utilización de plantillas, lo que permitirá ahorrar el tiempo necesario para la preparación inicial y el formato de los calendarios de uso frecuente (semanal, mensual y anual) en muchas empresas cubanas.

Además se le mostrará al usuario una vista previa de los calendarios a su disposición permitiéndole obtener una idea del aspecto que tendrán una vez finalice el proceso de generación de los mismos.

El asistente le permitirá al usuario adicionar tareas desde la vista de tareas disponible en la herramienta. Para esto se le brinda la posibilidad de seleccionar la fecha y hora de inicio de la misma, además de especificar un título, el lugar, categoría, descripción y por último seleccionar la frecuencia en la que se repetirá la tarea la cual está dividida en una repetición diaria, semanal, mensual y anual; además le dará la opción de personalizar dicha frecuencia. Esta opción le permitirá al usuario especificar otras maneras de repetir la tarea creada. El asistente también

brindará la posibilidad de modificar, eliminar y mostrar las tareas desde la vista Tareas de la herramienta. Antes de que el usuario cierre el asistente, se le brindará la opción de crear otro(s) calendario(s) de cualquier tipo (semanal, mensual o anual) con las mismas tareas que el anterior.

2.3 Lista de Reserva de Producto (LRP)

La LRP es una colección organizada y priorizada de los requisitos del producto. El objetivo de la misma se centra en especificar las cualidades requeridas por el software. Además de determinar el orden en que se le irá dando cumplimiento a cada requerimiento recogido según la prioridad establecida en cada uno. Los requisitos que en ella se recogen son clasificados en funcionales y no funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física. Expresan qué debe hacer el sistema. En cambio, los requisitos no funcionales son propiedades o cualidades que el sistema debe tener. Estas propiedades se pueden ver como las características que hacen al producto atractivo, usable, rápido o confiable. Recogen restricciones del sistema y de su desarrollo (robustez, entorno en el que funcionará la aplicación, volumen de información que gestionará, etc.).

En la siguiente tabla se muestra la LRP con cada requisito funcional y no funcional identificado. En la misma se recoge la prioridad con la que se le dará solución al requisito, el nombre del desarrollador que le dará cumplimiento, su número identificativo (Ítem*), la descripción del mismo, el tiempo estimado para su desarrollo dado en semanas y los roles que realizaron la estimación. Por último se muestran los requisitos no funcionales definidos para la realización de la aplicación a desarrollar.

RF (Requisitos Funcionales)				
Prioridad	Ítem *	Descripción	Estimación (Semanas)	Estimado por
Alta				
Adriam Delgado y	1	Crear calendario semanal.	2	Analista-

Maria del Carmen Rios				Programador
	2	Crear calendario mensual.	2	
	3	Crear calendario anual.	2	
	4	Gestionar tarea.	2	
Media				
	5	Añadir tareas al calendario semanal.	2	
	6	Añadir tareas al calendario mensual.	2	
	7	Añadir tareas al calendario anual.	2	
Baja				
	8	Mostrar vista previa del calendario.	1	
RNF (Requisitos No Funcionales)				
Software				
1	Tener instalada la Máquina Virtual de Java con una versión mayor o igual a 7.			
2	Tener instalada la suite ofimática LibreOffice en una versión mayor o igual a 3.			
Restricciones del diseño				
3	Lenguaje de programación Java.			
4	IDE de programación Netbeans.			
5	La aplicación deberá ser multiplataforma.			
Usabilidad				
6	El sistema debe poseer una interfaz cómoda, que posibilite a los usuarios sin experiencia una rápida adaptación.			
Interfaz				
7	Interfaz sencilla e intuitiva.			

Tabla 2: "Lista de Reserva de Producto (LRP)".

2.4 Historias de Usuario (HU)

Las historias de usuario se utilizan para especificar los requisitos del software. Las mismas son escritas por los clientes y su construcción depende principalmente de la habilidad que tengan estos para definir las. Son escritas en lenguaje natural, sin un formato predeterminado; estas guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo. Se elaboraron 8 HU, en correspondencia con los requisitos funcionales expuestos en la LRP. Estas permitieron especificar el modo en que se desarrollarán las funcionalidades del asistente propuesto.

2.4.1 HU Crear calendario semanal.

Historia de Usuario	
Número: HU-01	Nombre Historia de Usuario: Crear calendario semanal.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá la creación de un calendario semanal.	
Observaciones: Para crear un calendario semanal el usuario selecciona la opción Calendario semanal, además del rango de fechas (fecha inicial y final) para el cual desea elaborar el mismo. La herramienta tendrá lista la plantilla correspondiente al calendario elegido por el usuario con los datos seleccionados. Cuando el usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario correspondiente.	
Prototipo de interfaz: (Ver Anexo 3)	

2.4.2 HU Crear calendario mensual.

Historia de Usuario	
Número: HU-02	Nombre Historia de Usuario: Crear calendario mensual.
Modificación de Historia de Usuario Número: Ninguna.	

Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá la elaboración de un calendario mensual.	
Observaciones: Para crear un calendario mensual el usuario selecciona la opción Calendario mensual; selecciona además, el mes y el año para el cual desea elaborar el mismo. La herramienta tendrá lista la plantilla correspondiente al calendario elegido por el usuario con los datos seleccionados. Cuando el usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario correspondiente.	
Prototipo de interfaz: (Ver Anexo 4)	

2.4.3 HU Crear calendario anual.

Historia de Usuario	
Número: HU-03	Nombre Historia de Usuario: Crear calendario anual.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá la creación de un calendario anual.	
Observaciones: Para crear un calendario anual el usuario selecciona la opción Calendario anual; selecciona además, el año para el cual desea elaborar el mismo. La herramienta tendrá lista la plantilla correspondiente al calendario elegido por el usuario con los datos seleccionados. Cuando el usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario correspondiente.	
Prototipo de interfaz: (Ver Anexo 5)	

2.4.4 HU Gestionar tarea.

Historia de Usuario	
Número: HU-04	Nombre Historia de Usuario: Gestionar tarea.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Alta.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá adicionar, eliminar, modificar y mostrar las tareas que luego se le añadirán a los calendarios.	
Observaciones: Para adicionar una tarea se deben completar los campos siguientes: fecha y hora de inicio, título, lugar, categoría, repetir y descripción. El sistema mostrará las tareas que se vayan adicionando en la vista de tareas de la herramienta. Se podrá eliminar la(s) tarea(s) que seleccione el usuario además de modificar cualquier campo haciendo doble clic sobre ellos.	
Prototipo de interfaz: (Ver Anexo 6)	

2.4.5 HU Añadir tareas al calendario semanal.

Historia de Usuario	
Número: HU-05	Nombre Historia de Usuario: Añadir tareas al calendario semanal.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Media.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá añadir tareas al calendario semanal.	
Observaciones: El usuario debe seleccionar la opción correspondiente al calendario semanal. Podrá añadir tareas al mismo siempre y cuando haya una tarea adicionada a la vista de tareas disponible en la herramienta. El asistente tendrá lista la plantilla correspondiente al calendario semanal. Cuando el	

usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario y las tareas correspondientes.

Prototipo de interfaz: (Ver Anexo 7)

2.4.6 Añadir tareas al calendario mensual.

Historia de Usuario	
Número: HU-06	Nombre Historia de Usuario: Añadir tareas al calendario mensual.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Media.	Puntos Estimados: 2 semanas.
Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá añadir tareas al calendario mensual.	
Observaciones: El usuario debe seleccionar la opción correspondiente al calendario mensual. Podrá añadir tareas al mismo siempre y cuando haya una tarea adicionada a la vista de tareas disponible en la herramienta. El asistente tendrá lista la plantilla correspondiente al calendario mensual. Cuando el usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario y las tareas correspondientes.	
Prototipo de interfaz: (Ver Anexo 8)	

2.4.7 HU Añadir tareas al calendario anual.

Historia de Usuario	
Número: HU-07	Nombre Historia de Usuario: Añadir tareas al calendario anual.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Media.	Puntos Estimados: 2 semanas.

Riesgo en Desarrollo: Medio.	Puntos Reales: 2 semanas.
Descripción: Permitirá añadir tareas al calendario anual.	
Observaciones: El usuario debe seleccionar la opción correspondiente al calendario anual. Podrá añadir tareas al mismo siempre y cuando haya una tarea adicionada a la vista de tareas disponible en la herramienta. El asistente tendrá lista la plantilla correspondiente al calendario anual. Cuando el usuario cargue la plantilla, se abrirá automáticamente un documento Writer con el calendario y las tareas correspondientes.	
Prototipo de interfaz: (Ver Anexo 9)	

2.4.8 HU Mostrar vista previa del calendario.

Historia de Usuario	
Número: HU-08	Nombre Historia de Usuario: Mostrar vista previa del calendario.
Modificación de Historia de Usuario Número: Ninguna.	
Usuario: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	Iteración Asignada: 1
Prioridad en Negocio: Baja.	Puntos Estimados: 1 semana.
Riesgo en Desarrollo: Medio.	Puntos Reales: 1 semana.
Descripción: Permitirá mostrar una vista previa del calendario seleccionado.	
Observaciones: El asistente muestra al usuario una vista previa del tipo de calendario seleccionado (semanal, mensual o anual).	
Prototipo de interfaz: (Ver Anexo 10)	

2.6 Descripción de la arquitectura

Generalmente, en el proceso de desarrollo de una aplicación, suelen emplearse varios estilos arquitectónicos. Esto permite dar una mejor solución mediante el empleo de las ventajas que proporcionan cada uno de ellos. En el desarrollo del asistente se utiliza el estilo arquitectónico Arquitectura en Capas (*N-Capas* o *N-Layer*).

El estilo arquitectónico seleccionado se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de interactuar con otras capas y las responsabilidades la funcionalidad que implementan. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que se deba realizar algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado. Dentro de las características que posee este estilo se destacan las siguientes: la mayoría de las interacciones ocurren solo entre capas vecinas, los componentes de cada capa se comunican con los componentes de otras capas a través de interfaces bien conocidas y cada nivel agrega las responsabilidades y abstracciones del nivel inferior.

La selección de dicho estilo se fundamenta en la necesidad de dividir las capas de acuerdo con su responsabilidad. Como se muestra en la **Ilustración 1**, se definieron 3 capas: la capa LibreOffice, encargada de presentarle el asistente al usuario; la capa Presentación, encargada de capturar la información del usuario; y la capa Lógica de Negocio, encargada de recibir las peticiones del usuario y enviar las respuestas tras el proceso. En ella es donde se establecen todas las reglas del negocio que deben cumplirse.

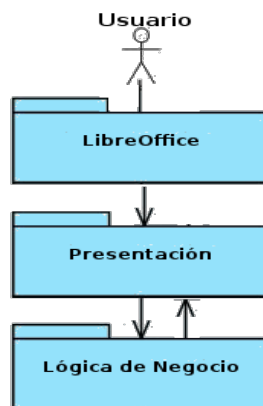


Ilustración 1: Arquitectura N-Capas (3-Capas).

2.7 Patrones de diseño empleados

Los patrones de diseño se han convertido en una metodología por excelencia del diseño a la hora de desarrollar una solución; describen clases y objetos, así como la comunicación entre ellos en vista de resolver un problema general del diseño en un contexto específico.

2.7.1 Patrones GRASP

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) describen los principios fundamentales de la asignación de responsabilidades a objetos, expresado en forma de patrones. Para el desarrollo del asistente se utilizan los siguientes patrones GRASP:

Experto (*Expert*): Se basa en asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. De esta manera se garantiza que los sistemas sean más fáciles de entender, mantener y ampliar, presentándose la oportunidad de reutilizar los componentes en futuras aplicaciones. En el asistente se hace uso de este patrón en el caso de implementar las funcionalidades de interacción con la API UNO, donde esta responsabilidad es asignada a la clase Documento.java.

Creador (*Creator*): Define la clase que debe tener la responsabilidad de crear una nueva instancia de otra clase. La clase B tiene la responsabilidad de crear una nueva instancia de la clase A si agrega, contiene, registra, utiliza o tiene los datos de inicialización de los objetos de A. Un caso en los que se pone de manifiesto este patrón en el asistente, es en la clase Manager.java cuando para la creación de tareas, se crea una instancia de la clase entidad Tarea.java para utilizar sus datos de inicialización.

2.7.2 Patrones GOF

Los patrones GoF (*Gang of Four*) [26] se clasifican en tres categorías basadas en su propósito: creacionales, estructurales y de comportamiento. En el desarrollo del asistente se hace uso del patrón Fachada, el cual se enmarca dentro de los estructurales y el patrón Estrategia, dentro de los de comportamiento.

Fachada (*Facade*): Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema, además define una interfaz de alto nivel que hace que el subsistema sea más fácil de entender. La clase CallWizards.java que se encuentra dentro del código de LibreOffice, actúa como fachada proporcionando una interfaz única y simplificada a los servicios más generales de LibreOffice.

Estrategia (Strategy): Define una familia de algoritmos, encapsula cada uno de ellos y los hace intercambiables. Además permite que un algoritmo varíe independientemente de quien los use. Este patrón se pone de manifiesto en la clase `RepitPersonalizado.java`. Esta clase contiene el método abstracto `RepetirFechasPersonalizado()` (método **estrategia**) el cual se implementa de diferentes formas en las clases `RepitPersonalizadoAnualPorDia.java`, `RepitPersonalizadoDiario.java`, `RepitPersonalizadoMensual.java`, `RepitPersonalizadoSemanal.java`, `RepitPersonalizadoAnualPorNumDia.java`.

2.8.2 Diagrama de paquetes

Para lograr un mayor entendimiento del funcionamiento del sistema, se hace necesario describir su estructura física. Esto se debe a que las mejoras que se le realicen posteriormente, deberán estar acopladas a la estructura que se define, proporcionando uniformidad al sistema en general. La **Ilustración 3** muestra cómo quedó organizado el asistente en términos de paquetes.

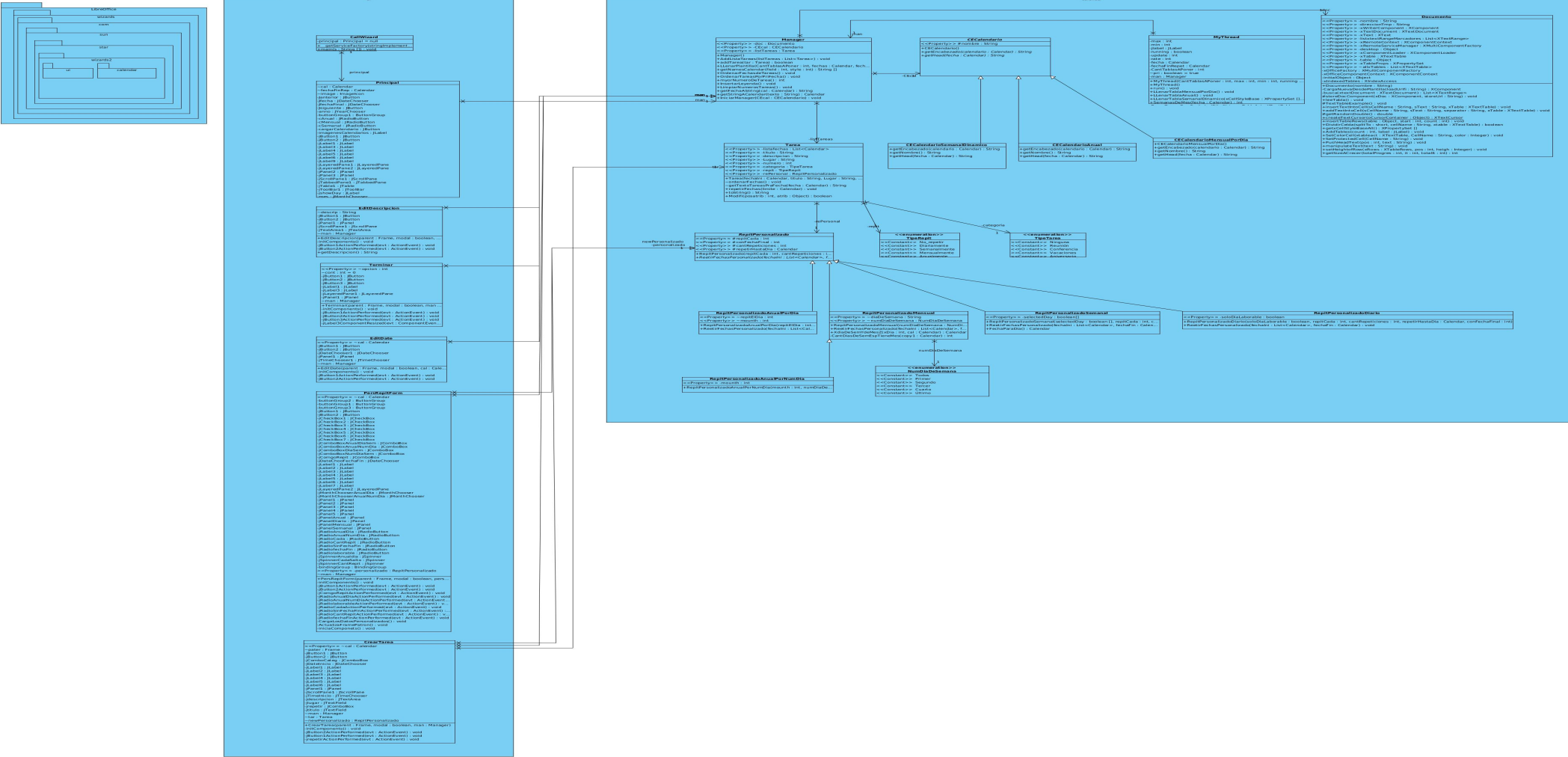


Ilustración 3: “Diagrama de paquetes del diseño”.

2.9 Conclusiones del capítulo

El desarrollo de este capítulo permitió obtener una comprensión cabal de los detalles de la aplicación mediante la descripción de la solución propuesta. Se generaron artefactos como son la Lista de Reserva del Producto que contiene los requisitos funcionales y no funcionales con los que debe cumplir el asistente y las HU que especifican el modo en que se efectuarán las funcionalidades del mismo. Además se definió la arquitectura N-Capas (3-Capas) y los patrones de diseño Experto, Creador, Fachada y Estrategia para brindar una mejor solución mediante el empleo de las ventajas que estos proporcionan. Por último se mostró el diagrama de clases del diseño basado en la arquitectura definida y el diagrama de paquetes correspondiente a la solución propuesta en el que se evidencia la estructura física del asistente. Al término de este capítulo se dejan sentadas las bases para el desarrollo de la aplicación de una forma clara y eficiente, en el menor tiempo posible.

Capítulo #3: Implementación y pruebas de la solución.

3.1 Introducción

Para lograr el desarrollo satisfactorio de la solución propuesta en el capítulo anterior, en el presente capítulo se describirá el plan de liberaciones, las tareas de ingeniería y el estándar de código a utilizar en la implementación de la misma. De forma adicional, se abordarán aspectos fundamentales para la integración del asistente con LibreOffice Writer. Por último, se presentarán los casos de prueba de aceptación recogidos como parte de la metodología de desarrollo de software seleccionada, las cuales se enfocan en la verificación del cumplimiento de las funcionalidades declaradas en el proceso de concepción de la aplicación.

3.2 Plan de liberaciones

Una vez realizadas las Historias de Usuario, y estimado el esfuerzo por los desarrolladores para la realización de las mismas, se hace necesario realizar la planificación de las etapas de implementación del asistente. Este plan contiene las HU por iteraciones, definiendo cuáles de ellas serán desarrolladas en cada iteración del proceso de implementación. La implementación del asistente se decide realizar en tres iteraciones, las cuales se describen a continuación:

➤ Iteración 1.

Esta primera iteración tiene como objetivo dar cumplimiento a las HU arquitectónicamente relevantes (01, 02, 03, 04); aquellas que constituyen el eje central de la estructura básica del asistente.

➤ Iteración 2.

Esta segunda iteración tiene como objetivo dar cumplimiento a las HU de prioridad media (05, 06, 07). Estas le brindarán al sistema un alto nivel de aceptación por parte del usuario ya que le permite a este añadir las tareas que tengan programadas a cualquiera de los calendarios disponibles en la herramienta.

➤ Iteración 3.

La tercera y última iteración tiene como objetivo dar cumplimiento a la HU número 8. Con la implementación de esta se culmina con las funcionalidades del asistente propuesto. Una vez terminada esta iteración se tendrá la aplicación lista para ser utilizada.

Iteración	Descripción de la iteración.	Orden de la HU a implementar.	Duración total de la iteración.
1	Desarrollo de las Historias de Usuario de alta prioridad.	01, 02, 03, 04	8 semanas
2	Desarrollo de las Historias de Usuario de prioridad media.	05, 06, 07	6 semanas
3	Desarrollo de la Historia de Usuario que tiene prioridad baja.	08	1 semana

3.3 Tareas de Ingeniería (TI)

3.3.1 TI asociadas a la HU Crear calendario semanal.

Tarea de Ingeniería	
Número Tarea: 1.1	Número Historia de Usuario: HU-04
Nombre Tarea: Diseñar el prototipo de interfaz de la aplicación.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se diseña una interfaz que permita a los usuarios interactuar con la aplicación.	

Tarea de Ingeniería	
Número Tarea: 1.2	Número Historia de Usuario: HU-01

Nombre Tarea: Elaborar la plantilla para el calendario semanal.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se elabora la plantilla a ser utilizada para el tipo de calendario (semanal).	

Tarea de Ingeniería	
Número Tarea: 1.3	Número Historia de Usuario: HU-01
Nombre Tarea: Estudiar la librería jcalendar-1.3.3.jar.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se estudia la librería jcalendar-1.3.3.jar para utilizar sus componentes visuales.	

3.3.2 TI asociada a la HU Crear calendario mensual.

Tarea de Ingeniería	
Número Tarea: 2.1	Número Historia de Usuario: HU-02
Nombre Tarea: Elaborar la plantilla para el calendario mensual.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se elabora la plantilla a ser utilizada para el tipo de calendario (mensual).	

3.3.3 TI asociada a la HU Crear calendario anual.

Tarea de Ingeniería	
Número Tarea: 3.1	Número Historia de Usuario: HU-03

Nombre Tarea: Elaborar la plantilla para el calendario anual.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se elabora la plantilla a ser utilizada para el tipo de calendario (anual).	

3.3.4 TI asociadas a la HU Gestionar tarea.

Tarea de Ingeniería	
Número Tarea: 4.1	Número Historia de Usuario: HU-04
Nombre Tarea: Estudiar la librería JTimeChooser_0.1.0.jar.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se estudia la librería JTimeChooser_0.1.0.jar para utilizar sus componentes visuales.	

Tarea de Ingeniería	
Número Tarea: 4.2	Número Historia de Usuario: HU-04
Nombre Tarea: Diseñar el prototipo de interfaz para la vista Tareas disponible en la herramienta.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se diseña una interfaz que permita visualizar los componentes para interactuar con el sistema.	

Tarea de Ingeniería	
Número Tarea: 4.3	Número Historia de Usuario: HU-04

Nombre Tarea: Implementar la funcionalidad adicionar tarea.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se implementa la funcionalidad que permite adicionar una tarea genérica.	

Tarea de Ingeniería	
Número Tarea: 4.4	Número Historia de Usuario: HU-04
Nombre Tarea: Seleccionar la(s) tarea(s) a eliminar.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se implementa la funcionalidad que permite eliminar una o varias tareas.	

Tarea de Ingeniería	
Número Tarea: 4.5	Número Historia de Usuario: HU-04
Nombre Tarea: Mostrar tarea.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se implementa la funcionalidad que permite mostrar la(s) tarea(s) en el componente visual jTable que se muestra en la vista Tareas disponible en la herramienta.	

Tarea de Ingeniería	
Número Tarea: 4.6	Número Historia de Usuario: HU-04
Nombre Tarea: Editar tareas.	

Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se implementan las funcionalidades que permiten editar la(s) tarea(s).	

3.3.5 TI asociada a la HU Añadir tareas al calendario semanal.

Tarea de Ingeniería	
Número Tarea: 5.1	Número Historia de Usuario: HU-05
Nombre Tarea: Estudiar la clase XCellText.	
Tipo de Tarea : Desarrollo	Puntos Estimados: 1 semana
Fecha Inicio:	Fecha Fin:
Programador Responsable: Adriam Delgado Rivero y Maria del Carmen Rios Jiménez.	
Descripción: Se estudia la clase XCellText la cual permitirá introducir los datos de las tareas dentro de las celdas de una tabla.	

3.4 Estándar de codificación

Una de las buenas prácticas propuestas por las metodologías ágiles es la adopción de un estándar de codificación por parte del equipo de desarrollo del software. Este estándar está compuesto por varias normas las cuales son muy útiles por muchas razones, entre ellas se destacan las siguientes:

- Facilitan el mantenimiento de una aplicación.
- Permiten que cualquier programador entienda y pueda mantener la aplicación.
- Mejoran la legibilidad del código, al mismo tiempo que permiten su rápida comprensión.

Para dar cumplimiento a la solución propuesta se escogió el estándar CamelCase recomendado por la empresa Sun Microsystems. Este estándar es empleado en el desarrollo de software sobre la plataforma Java. Ha sido difundido y aceptado ampliamente por toda la comunidad Java. (**Ver Anexo 11**).

3.5 Integración del asistente con LibreOffice

Para lograr la integración del asistente desarrollado con LibreOffice a continuación se explica cómo funciona dicho proceso.

LibreOffice llama a la clase *CallWizard*. Esta a su vez llama a la vista inicial del asistente. La clase *CallWizard* se utiliza como fachada para comunicar LibreOffice con el asistente, manteniéndolos desacoplados. LibreOffice brinda el manejador de servicios (*ServiceManager*), usualmente utilizando el servidor: localhost, y el puerto: 8100. El manejador de servicios facilita el uso de API UNO y por tanto todas las interfaces que esta provee. Las interfaces de API UNO fueron utilizadas para las operaciones siguientes: abrir plantillas, modificar y adicionar contenido (texto y tablas).

3.5.1 Uso de la clase CallWizard

Antes de describir el uso de la clase *CallWizards* y de API UNO es necesario conocer la estructura que exige LibreOffice a la hora de elaborar un asistente (*wizards*).

Cuando se elabora un asistente para LibreOffice (en este caso denominado Calendar), todo el contenido del mismo debe almacenarse en la carpeta *wizards* que se encuentra dentro del código fuente de LibreOffice (en la dirección /core/wizards/). Dentro de este directorio, cada asistente tiene un archivo *Jar_NombreDelAsistente.mk* (en este caso *Jar_Calendar.mk*), donde se especifican las direcciones de cada una de las librerías y clases que contiene dicho asistente. Además se tiene un archivo *Jar_commonwizards.mk* en el cual se especifican las direcciones de cada una de las librerías y clases comunes para todos los asistentes de LibreOffice. En *Module_wizards.mk* se declaran cada uno de los archivos *Jar_NombreDelAsistente.mk*, pertenecientes a cada *wizards* de LibreOffice.

Las clases correspondientes a la capa Presentación del asistente se encuentran en la dirección: `/core/wizards/com/sun/star/wizards/ui/`.

Las clases correspondientes a la capa Lógica de Negocio del asistente se encuentran en la dirección: `/core/wizards/com/sun/star/wizards/Calendar/`. Dentro de este directorio se encuentra el archivo `Calendar.component` mediante el cual se llama a la clase `CallWizards` cuyo método `Main()` se encargada de presentarle la interfaz principal del asistente al usuario.

3.5.2 Uso de API UNO

Como se ha mencionado en capítulos anteriores la API UNO se utilizó para lograr interoperabilidad entre el asistente y la suite ofimática LibreOffice. No es objetivo del presente trabajo de diploma explicar cómo funciona API UNO, solo se brindarán los aspectos fundamentales que se tuvieron en cuenta para su utilización en el asistente desarrollado. Para un estudio completo de la API UNO se recomienda consultar la guía para desarrolladores de OpenOffice.org (*OpenOffice.org Developer's Guide*).

- El asistente desarrollado inicializa UNO y obtiene la conexión remota (*XRemoteContext*) con OpenOffice.org mediante la llamada al método `com.sun.star.comp.helper.Bootstrap.bootstrap()`.
- A partir de *XRemoteContext* se obtiene el administrador de servicio remoto (*XRemoteServiceManager*) mediante el método `getServiceManager()`.
- Luego se crea el objeto escritorio remoto (*desktop*) mediante el método `createInstanceWithContext()` de la interfaz *XRemoteServiceManager*.
- A partir de una referencia de *desktop* se obtiene *XComponentLoader* mediante el método `queryInterface()`.

- De la interfaz *XComponentLoader*, se utiliza el método *loadComponentFromURL()*, encargado de abrir los archivos de OpenOffice.org y retornar la interfaz *XComponent*. En este método se utiliza la propiedad booleana *AsTemplate* perteneciente al servicio *com.star.document.MediaDescriptor*. Mediante la misma se le ordena a la oficina crear un nuevo documento desde un archivo dado.
- A partir de *XComponent* se obtiene la interfaz *XTextDocument*, así como la interfaz *XTextTablesSupplier* mediante el método *queryInterface()*.

Una vez obtenidas las interfaces *XTextDocument* y *XTextTablesSupplier* se facilita el acceso a las interfaces que contienen las funcionalidades que fueron utilizadas para el desarrollo del asistente.

La siguiente tabla muestra el conjunto de interfaces de API UNO que fueron utilizadas a partir de *XTextDocument* y *XTextTablesSupplier*. Además se muestran los métodos que se utilizaron de cada una de las interfaces, así como una breve descripción de los mismos.

Interfaz	Método	Descripción
XTextTablesSupplier	getTextTables()	Obtiene la colección de tablas.
XIndexAccess	getByIndex()	Devuelve un objeto Object del cual se obtiene XTextTable.
XTextTable	getCellByName()	Devuelve la celda de nombre <i>name</i> .
	createCursorByCellName(<i>name</i>)	Crea un cursor en celda de nombre <i>name</i> .
	getRows()	Devuelve las filas de una tabla.
XTextTableCursor	splitRange()	
XTableRows	insertByIndex(<i>start</i> , <i>count</i>)	Inserta <i>count</i> filas a partir de <i>start</i> índice de filas.
	getCount()	Devuelve la cantidad de filas de una tabla.

XText	setString(<i>str</i>)	Crea texto <i>str</i> .
	insertString(<i>str</i>)	Inserta texto <i>str</i> .
	createTextCursor()	Devuelve la interfaz XTextCursor.
XPropertySet	setProperty(<i>property</i> , <i>value</i>)	Establece el valor <i>value</i> en la propiedad de nombre <i>property</i> .
	getProperty(<i>property</i>)	Devuelve el valor de la propiedad con nombre <i>property</i> .
XTextDocument	getText()	Devuelve la interfaz XText.
XSearchable	findAll(<i>str</i>)	Busca <i>str</i> en todo su contexto.
XTextCursor	gotoEnd()	Coloca el cursor al final del texto.
Todas	queryInterface(<i>child</i> , <i>father</i>)	Devuelve la subinterfaz hija <i>child</i> dada la interfaz padre <i>father</i> .

Tabla 3: "Interfaces y métodos que se utilizaron de API UNO."

En la imagen siguiente se puede observar la estructura jerárquica de cada una de las interfaces de API UNO que se utilizaron:

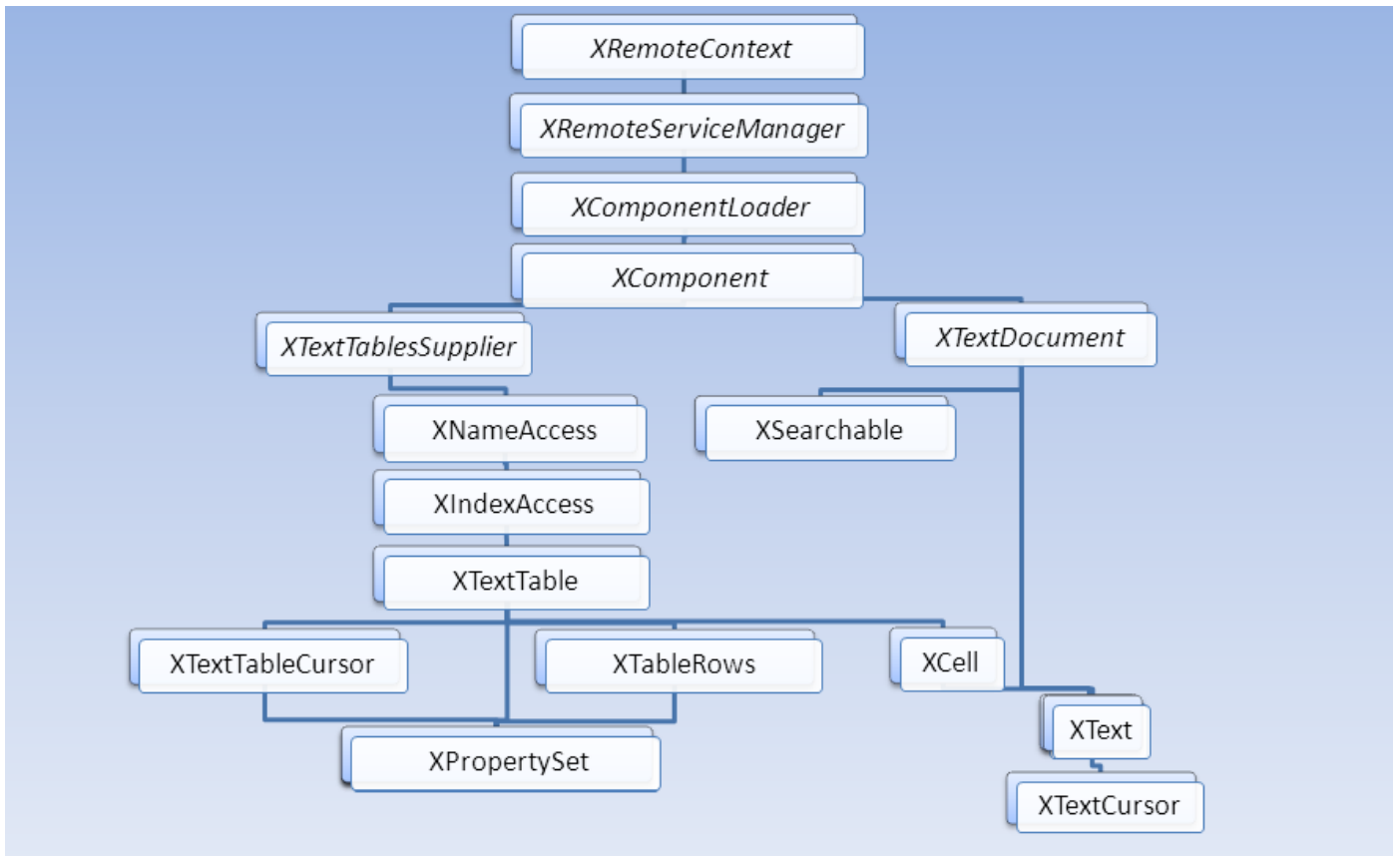


Ilustración 4: "Jerarquía de interfaces utilizadas de API UNO."

Luego de implementada la solución, se procede a describir un conjunto de pruebas de aceptación asociadas a las Historias de Usuario, con el objetivo de verificar que la aplicación desarrollada cumpla con el funcionamiento esperado.

3.6 Pruebas

En el proceso de desarrollo de un software, la realización de pruebas constituye una actividad importante para llevar a cabo el control de la calidad del mismo.

A la aplicación desarrollada se le realizaron disímiles pruebas para comprobar su factibilidad. Las pruebas verifican el software en busca de errores en el manejo de las restricciones de integridad de las clases del sistema. Tienen como objetivo la verificación de discrepancias entre los

requisitos y los resultados de la ejecución del software. Las pruebas determinan además el nivel de calidad y permiten comprobar el grado de cumplimiento con respecto a las especificaciones iniciales del sistema.

Se utilizó el método de caja negra [27] para realizar las pruebas sobre la interfaz del asistente. Se realizaron pruebas de aceptación a nivel de cliente con las que se pudo constatar que el sistema cumple puntualmente con todos los requisitos planteados. Al término de las pruebas se obtuvo un producto con calidad que cumple con todas las especificaciones inicialmente planteadas.

En la fase de desarrollo de la metodología SXP, específicamente en el flujo de trabajo prueba se genera un artefacto denominado Caso de Prueba de Aceptación. A continuación se muestran los Casos de Prueba de Aceptación correspondientes a las HU implementadas.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 01	Nombre Historia de Usuario: Crear calendario semanal.
Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad crear calendario semanal, seleccionando los datos correctamente.	
Condiciones de Ejecución: Acceder al asistente.	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. Seleccionar la opción Calendario semanal. 2. Seleccionar fecha inicial para elaborar el calendario. 3. Seleccionar fecha final para elaborar el calendario. 4. Se presiona el botón Generar Calendario. 	
Resultado Esperado: Debe aparecer un documento Writer con el calendario semanal para el rango de fecha seleccionado.	

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba: 02

Nombre Historia de Usuario: Crear calendario mensual.

Nombre de la persona que realiza la prueba: Jailen García González

Descripción de la Prueba: Prueba a la funcionalidad crear calendario mensual, seleccionando los datos correctamente.

Condiciones de Ejecución: Acceder al asistente.

Entrada / Pasos de ejecución:

1. Seleccionar la opción Calendario mensual.
2. Seleccionar el mes para el cual se desea generar el calendario.
3. Seleccionar el año para el cual se desea generar el calendario.
4. Se presiona el botón Generar Calendario.

Resultado Esperado: Debe aparecer un documento Writer con el calendario del mes y año seleccionado.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba: 03

Nombre Historia de Usuario: Crear calendario anual.

Nombre de la persona que realiza la prueba: Jailen García González

Descripción de la Prueba: Prueba a la funcionalidad crear calendario anual, seleccionando los datos correctamente.
Condiciones de Ejecución: Acceder al asistente.
Entrada / Pasos de ejecución: 1. Seleccionar la opción Calendario anual. 2. Seleccionar el año para el cual se desea generar el calendario. 3. Se presiona el botón Generar Calendario.
Resultado Esperado: Debe aparecer un documento Writer con el calendario del año seleccionado.
Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación	
Código Caso de Prueba: 04	Nombre Historia de Usuario: Gestionar tarea.
Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad adicionar tarea.	
Condiciones de Ejecución: Acceder a la vista de tareas disponible en la herramienta.	
Entrada / Pasos de ejecución: 1. Seleccionar la opción adicionar tarea. 2. Completar los campos con los datos siguientes: <div style="margin-left: 40px;">Fecha y hora de inicio: 13:30, 16 de mayo del 2013</div> <div style="margin-left: 40px;">Título: Predefensa</div> <div style="margin-left: 40px;">Lugar: Aula 303</div>	

Categoría: Ninguna

Repetir: No_repetir

Descripción: Ninguna

3. Presionar el botón Aceptar.

Resultado Esperado: Debe aparecer la tarea con los datos especificados en la vista de tareas de la herramienta.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba: 05

Nombre Historia de Usuario: Gestionar tarea.

Nombre de la persona que realiza la prueba: Jailen García González

Descripción de la Prueba: Prueba a la funcionalidad eliminar tarea.

Condiciones de Ejecución: Acceder a la vista de tareas disponible en la herramienta.

Entrada / Pasos de ejecución:

1. Seleccionar la(s) tarea(s) a eliminar.

2. Presionar el botón eliminar.

Resultado Esperado: Deben desaparecer de la lista de tareas las que fueron eliminadas.

Evaluación de la Prueba: Satisfactoria.

Caso de Prueba de Aceptación

Código Caso de Prueba: 06

Nombre Historia de Usuario: Gestionar tarea.

Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad modificar tarea.	
Condiciones de Ejecución: Acceder a la vista de tareas disponible en la herramienta. Para modificar los campos de texto se hará doble clic sobre los mismos, uno para los demás.	
Entrada / Pasos de ejecución:	
Para el caso que se quieran modificar todos los campos de la tarea:	
1. Seleccionar la fecha y hora de inicio: 13:30, 12 de junio del 2013	
2. Título: Exposición de la tesis.	
3. Lugar: Aula 303	
4. Categoría: Ninguna	
5. Se repite: No repetir	
6. Descripción: Ninguna	
Resultado Esperado: Debe aparecer la tarea con los datos correspondientes.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 07	Nombre Historia de Usuario: Añadir tareas al calendario semanal.
Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad añadir tareas al calendario semanal.	
Condiciones de Ejecución: Acceder al asistente.	

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Seleccionar el tipo de calendario (semanal) con sus datos correspondientes (fecha inicial y final). 2. Crear la(s) tarea(s) que se deseen adicionar al calendario. 3. Presionar el botón Generar Calendario.
<p>Resultado Esperado: Debe aparecer un documento con el calendario semanal para el rango de fecha seleccionado. Además deben aparecer las tareas que se le adicionaron.</p>
<p>Evaluación de la Prueba: Satisfactoria.</p>

<p>Caso de Prueba de Aceptación</p>	
<p>Código Caso de Prueba: 08</p>	<p>Nombre Historia de Usuario: Añadir tareas al calendario mensual.</p>
<p>Nombre de la persona que realiza la prueba: Jailen García González</p>	
<p>Descripción de la Prueba: Prueba a la funcionalidad añadir tareas al calendario mensual, seleccionando y completando los datos correctamente.</p>	
<p>Condiciones de Ejecución: Acceder al asistente.</p>	
<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. Seleccionar el tipo de calendario (mensual) con sus datos correspondientes (mes y año). 2. Crear la(s) tarea(s) que se deseen adicionar al calendario. 3. Presionar el botón Generar Calendario. 	
<p>Resultado Esperado: Debe aparecer un documento con el calendario mensual para el mes y año seleccionado. Además deben aparecer las tareas que se le adicionaron.</p>	
<p>Evaluación de la Prueba: Satisfactoria.</p>	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 09	Nombre Historia de Usuario: Añadir tareas al calendario anual.
Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad añadir tareas al calendario anual, seleccionando y completando los datos correctamente.	
Condiciones de Ejecución: Acceder al asistente.	
Entrada / Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar el tipo de calendario (anual) con su dato correspondiente (año). 2. Crear la(s) tarea(s) que se deseen adicionar al calendario. 3. Presionar el botón Generar Calendario. 	
Resultado Esperado: Debe aparecer un documento con el calendario para el año seleccionado. Además deben aparecer las tareas que se le adicionaron.	
Evaluación de la Prueba: Satisfactoria.	

Caso de Prueba de Aceptación	
Código Caso de Prueba: 10	Nombre Historia de Usuario: Mostrar vista previa de calendario.
Nombre de la persona que realiza la prueba: Jailen García González	
Descripción de la Prueba: Prueba a la funcionalidad mostrar vista previa de calendario.	
Condiciones de Ejecución: Acceder al asistente.	
Entrada / Pasos de ejecución:	

1. Seleccionar el calendario deseado.
Resultado Esperado: Debe mostrar una vista previa del calendario seleccionado.
Evaluación de la Prueba: Satisfactoria.

3.7 Conclusiones del capítulo

En este capítulo se definió el plan de liberaciones mediante el cual se llevó a cabo la planificación de las etapas de implementación del asistente, así como las tareas de ingeniería que se desarrollaron. Se escogió el estándar de código CamelCase recomendado por la empresa Sun Microsystems, el cual permitió mejorar la legibilidad del código con el objetivo de facilitar el entendimiento del mismo por parte de cualquier programador. Se abordaron aspectos fundamentales a tener en cuenta para la integración del asistente con la herramienta LibreOffice Writer como fueron la utilización de la clase CallWizard que contiene los elementos para la conexión con LibreOffice y la utilización de la API UNO para lograr la interoperabilidad entre el asistente desarrollado y LibreOffice Writer mediante la utilización de los métodos de sus interfaces. Por último se diseñaron y ejecutaron pruebas de aceptación a nivel de cliente al asistente desarrollado, utilizando el método de caja negra. Después de realizar las mismas se comprobó que la aplicación funciona correctamente cumpliendo con todos los requisitos funcionales expuestos en el capítulo anterior.

Conclusiones generales

Al término de la presente investigación se evidencia el cumplimiento de los objetivos planteados al inicio de la misma, lo que permitió arribar a las siguientes conclusiones:

- El estudio de sistemas y herramientas gestoras de calendario, permitió identificar características y funcionalidades que resultaron útiles para desarrollar el asistente, las cuales se muestran a continuación:
 - ✓ Elaborar calendarios mediante la utilización de plantillas.
 - ✓ Visualizar varios calendarios.
 - ✓ Adicionar tareas a los calendarios.
- Con el análisis y diseño del asistente se generaron artefactos como son la Lista de Reserva del Producto que contiene los requisitos funcionales y no funcionales con los que debe cumplir el asistente, el Plan de liberaciones en el que se planifican las etapas de desarrollo del asistente y las Historias de Usuario que especifican cada uno de los requisitos de la aplicación. Además se obtuvo el diagrama de clases del diseño basándose en una arquitectura N-Capas (3-Capas) y el diagrama de paquetes en el que se evidencia la estructura física del asistente.
- El desarrollo del asistente, guiado por la metodología ágil SXP, permitió obtener una aplicación *desktop* completamente libre y multiplataforma, que automatiza el proceso de elaboración de calendarios en cualquier empresa o entidad.
- El diseño y ejecución de pruebas de aceptación mediante el método de Caja Negra, permitió obtener un asistente con calidad, que cumple puntualmente con todos los requisitos planteados.

Recomendaciones

Existen varias funcionalidades que quedan fuera del alcance definido para esta investigación y que se considera aumentarían el valor agregado de la misma. Algunas de ellas se recomiendan a continuación:

- Incluir el asistente en la personalización cubana de la suite ofimática LibreOffice.
- Lograr el intercambio de información entre el asistente desarrollado y demás herramientas gestoras de calendario existentes utilizando el estándar recomendado iCalendar.
- Lograr la aceptación del asistente en la comunidad de LibreOffice. Un requisito indispensable para lograr la aceptación del asistente en la comunidad de LibreOffice es internacionalizarlo. La internacionalización es el proceso de diseñar software de manera tal que pueda adaptarse a diferentes idiomas y regiones sin la necesidad de realizar cambios de ingeniería ni en el código.

Referencias bibliográficas

[1]. “Ger. Software libre y gobierno Los países más Open Source | WebAyunate.2010.” [Online]. Available:

<http://www.webayunate.com/software-libre-gobierno-paises-que-usan-open-source-linux/>.

[Accessed: 06-Oct-2012].

[2]. “Yoandy Pérez Villazón. Guía cubana para la migración a Software Libre. 2008.” [Online]. Available:

<ftp://ftp.cult.cu/softwarelibre/documentacion/Guia%20cubana%200.32.pdf>. [Accessed: 07-Oct-2012].

[3]. “Migrando al software libre en Cuba - Software Libre.” [Online]. Available:

<http://www.somoslibres.org/modules.php?name=News&file=article&sid=4231>. [Accessed: 08-Oct-2012].

[4]. “LibreOffice - Wikipedia, la enciclopedia libre.” [Online]. Available:

<http://es.wikipedia.org/wiki/LibreOffice>. [Accessed: 23-Oct-2012].

[5]. “Características LibreOffice.” [Online]. Available:

<http://es.libreoffice.org/caracteristicas/>. [Accessed: 29-Oct-2012].

[6]. “ISO/IEC 26300:2006 - Information technology -- Open Document Format for Office Applications (OpenDocument) v1.0.” [Online]. Available:

www.iso.org/iso/catalogue_detail?csnumber=43485. [Accessed: 11-Nov-2012].

[7]. “Ayuda Microsoft Outlook 2007. 2007.”

[8]. “Zimbra Calendar is an open source group calendar with Linux and Mac OS X server.” [Online]. Available:

<http://www.zimbra.com/products/calendar-collaboration.html> . [Accessed: 08-Feb-2013].

[9]. "ChannelPlanet." [Online] 2006. [Online]. Available:

<http://www.channelplanet.com/index.php?idcategoria=10302>. [Accessed: 10-Feb-2013]

[10]. "Apple - iCloud - Tu contenido en todos tus dispositivos." [Online]. Available:

www.apple.com/es/icloud/ . [Accessed: 12-Feb-2013].

[11]. "Google. Visita guiada de Google Calendar." [Online]. Available:

<http://www.google.com/intl/es/googlecalendar/tour2.html>. [Accessed: 07-Mar-2013]

[12]. "Yahoo Calendar." [Online]. Available:

<http://ycorpblog.com/2008/10/07/time-for-a-calendar-revival/>. [Accessed: 10-Mar-2013]

[13]. "Mozilla Sunbird." [Online]. Available:

<http://www.difundefirefox.com/mozilla-sunbird>. [Accessed: 12-Mar-2013]

[14]. "vCalendar Overview". [Online]. Available:

<http://www.imc.org/pdi/vcaloverview.html>. [Accessed: 15-Mar-2013].

[15]. "ISO 8601:2004 -Data elements and interchange formats -- Information interchange – Representation of dates and times. " [Online]. Available:

http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40874.

[Accessed: 21-Mar-2013].

[16]. "RFC 5545 - Internet Calendaring and Scheduling Core Object Specification (iCalendar)." [Online]. Available:

<http://tools.ietf.org/html/rfc5545>. [Accessed: 23-Mar-2013].

- [17]. "iCalendar - Wikipedia, la enciclopedia libre." [Online]. Available:
<http://es.wikipedia.org/wiki/Icalendar>. [Accessed: 30-Mar-2013].
- [18]. "Microformats Community. Microformats." [Online]. Available:
<http://microformats.org/wiki/hcalendar>. [Accessed: 04-Abr-2013].
- [19]. "Uno - Apache OpenOffice Wiki." [Online]. Available:
<http://wiki.openoffice.org/wiki/Uno>. [Accessed: 08-Abr-2013].
- [20]. Gladys Marsi Peñalver Romero, y Abel Meneses Abad. SXP, metodología ágil para proyectos de software libre. 2009.
- [21]. "Netbeans IDE - Overview." [Online]. Available:
<http://netbeans.org/features/index.html>. [Accessed: 11-Abr-2013].
- [22]. "Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) por Visual Paradigm International Ltd. - reporte y descarga." [Online]. Available:
www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/. [Accessed: 29-Apr-2013].
- [23]. "Subversion (software) - Wikipedia, la enciclopedia libre." [Online]. Available:
[http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software)). [Accessed: 30-Abr-2013].
- [24]. "RapidSVN-doc.ubuntu-es." [Online]. Available:
http://doc.ubuntu-es.org/Rapid_SVN. [Accessed: 30-Abr-2013].
- [25]. S. Pressman, Roger. Ingeniería de software. Un enfoque práctico. Quinta edición.

[26]. "Design Patterns: Elements of Reusable Object-Oriented Software. ERICH GAMMA, RICHARD HELM, RALPH JOHNSON, JOHN VLISSIDES. Pearson Education, 1994. 1994." [Online]. Available:

<http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf> [Accessed: 03-May-2013].

[27]. "Caja negra (sistemas) - Wikipedia, la enciclopedia libre." [Online]. Available: [http://es.wikipedia.org/wiki/Caja_negra_\(sistemas\)](http://es.wikipedia.org/wiki/Caja_negra_(sistemas)). [Accessed: 03-May-2013].

Bibliografía

1. “Ger. Software libre y gobierno Los países más Open Source | WebAyunate.2010.” [Online]. Available:

<http://www.webayunate.com/software-libre-gobierno-paises-que-usan-open-source-linux/>.

[Accessed: 06-Oct-2012].

2. “Yoandy Pérez Villazón. Guía cubana para la migración a Software Libre. 2008.” [Online]. Available:

<ftp://ftp.cult.cu/softwarelibre/documentacion/Guia%20cubana%200.32.pdf>. [Accessed: 07-Oct-2012].

3. “Migrando al software libre en Cuba - Software Libre.” [Online]. Available:

<http://www.somoslibres.org/modules.php?name=News&file=article&sid=4231>. [Accessed: 08-Oct-2012].

4. “LibreOffice - Wikipedia, la enciclopedia libre.” [Online]. Available:

<http://es.wikipedia.org/wiki/LibreOffice>. [Accessed: 23-Oct-2012].

5. “Características LibreOffice.” [Online]. Available:

<http://es.libreoffice.org/caracteristicas/>. [Accessed: 29-Oct-2012].

6. “ISO/IEC 26300:2006 - Information technology -- Open Document Format for Office Applications (OpenDocument) v1.0.” [Online]. Available:

www.iso.org/iso/catalogue_detail?csnumber=43485. [Accessed: 11-Nov-2012].

7. “Ayuda Microsoft Outlook 2007. 2007.”

8. “Zimbra Calendar is an open source group calendar with Linux and Mac OS X server.” [Online]. Available:

<http://www.zimbra.com/products/calendar-collaboration.html> . [Accessed: 08-Feb-2013].

9. "ChannelPlanet." [Online] 2006. [Online]. Available:

<http://www.channelplanet.com/index.php?idcategoria=10302>. [Accessed: 10-Feb-2013]

10. "Apple - iCloud - Tu contenido en todos tus dispositivos." [Online]. Available:

www.apple.com/es/icloud/ . [Accessed: 12-Feb-2013].

11. "Google. Visita guiada de Google Calendar." [Online]. Available:

<http://www.google.com/intl/es/googlecalendar/tour2.html>. [Accessed: 07-Mar-2013]

12. "Yahoo Calendar." [Online]. Available:

<http://ycorpblog.com/2008/10/07/time-for-a-calendar-revival/>. [Accessed: 10-Mar-2013]

13. "Mozilla Sunbird." [Online]. Available:

<http://www.difundefirefox.com/mozilla-sunbird>. [Accessed: 12-Mar-2013]

14. "vCalendar Overview". [Online]. Available:

<http://www.imc.org/pdi/vcaloverview.html>. [Accessed: 15-Mar-2013].

15. "ISO 8601:2004 -Data elements and interchange formats -- Information interchange – Representation of dates and times. " [Online]. Available:

http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=40874.

[Accessed: 21-Mar-2013].

16. "RFC 5545 - Internet Calendaring and Scheduling Core Object Specification (iCalendar)." [Online]. Available:

<http://tools.ietf.org/html/rfc5545>. [Accessed: 23-Mar-2013].

17. "iCalendar - Wikipedia, la enciclopedia libre." [Online]. Available:
<http://es.wikipedia.org/wiki/Icalendar>. [Accessed: 30-Mar-2013].
18. "Microformats Community. Microformats." [Online]. Available:
<http://microformats.org/wiki/hcalendar>. [Accessed: 04-Abr-2013].
19. "Uno - Apache OpenOffice Wiki." [Online]. Available:
<http://wiki.openoffice.org/wiki/Uno>. [Accessed: 08-Abr-2013].
20. Gladys Marsi Peñalver Romero, y Abel Meneses Abad. SXP, metodología ágil para proyectos de software libre. 2009.
21. "Netbeans IDE - Overview." [Online]. Available:
<http://netbeans.org/features/index.html>. [Accessed: 11-Abr-2013].
22. "Visual Paradigm for UML (ME) - (Paradigma Visual para UML (ME)) (Visual Paradigm for UML (ME)) por Visual Paradigm International Ltd. - reporte y descarga." [Online]. Available:
www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%28M%C3%8D%29_14720_p/. [Accessed: 29-Apr-2013].
23. "Subversion (software) - Wikipedia, la enciclopedia libre." [Online]. Available:
[http://es.wikipedia.org/wiki/Subversion_\(software\)](http://es.wikipedia.org/wiki/Subversion_(software)). [Accessed: 30-Abr-2013].
24. "RapidSVN-doc.ubuntu-es." [Online]. Available:
http://doc.ubuntu-es.org/Rapid_SVN. [Accessed: 30-Abr-2013].
25. S. Pressman, Roger. Ingeniería de software. Un enfoque práctico. Quinta edición.

26. "Design Patterns: Elements of Reusable Object-Oriented Software. ERICH GAMMA, RICHARD HELM, RALPH JOHNSON, JOHN VLISSIDES. Pearson Education, 1994. 1994." [Online]. Available:

<http://www.uml.org.cn/c++/pdf/DesignPatterns.pdf> [Accessed: 03-May-2013].

27. "Caja negra (sistemas) - Wikipedia, la enciclopedia libre." [Online]. Available:

[http://es.wikipedia.org/wiki/Caja_negra_\(sistemas\)](http://es.wikipedia.org/wiki/Caja_negra_(sistemas)). [Accessed: 03-May-2013].

28. OpenOffice.org 3.1. Developer's Guide.

29. Hernández Sampieri, Roberto. Metodología De La Investigacion. Cuarta edición. 2006.

Glosario de términos

Ofimática: Es un acrónimo compuesto por los términos oficina e informática. El concepto hace referencia a la automatización de las actividades que se realizan en una oficina.

Suite Ofimática: Es un 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.

Microsoft Office: Suite ofimática de la empresa Microsoft. Algunos de los programas que puede incluir (de acuerdo con la versión) son los siguientes: Word, Excel, PowerPoint, Visio, Access, FrontPage.

LibreOffice: es una Suite Ofimática libre, la cual fue creada como una bifurcación de la también libre Openoffice.org tras la adquisición de Sun Microsystems por parte de la multinacional Oracle Corporation y la posterior cancelación del proyecto OpenSolaris por parte de la misma corporación. Actualmente se encuentra en la versión 3.6.1 y es la aplicación ofimática por defecto en muchas distribuciones Linux como OpenSUSE, Ubuntu, Fedora, entre otras. . Esta suite incluye aplicaciones tales como: Writer, Calc, Impress, Base, Math y Draw las cuales son de uso habitual en una oficina.

Asistente: Aplicación informática que guía al usuario inexperto en el manejo de un programa.

Calendario: Sistema de representación del paso de los días, agrupados en unidades superiores, como semanas, meses y años.

PDF: (*Portable Document Format*) Formato de Documento Portátil es un formato de archivo sencillo. Tiene como objetivo representar al documento de forma independiente al periférico de reproducción, sea este una pantalla de ordenador, una impresora, etc. Los archivos PDF pueden ser creados a través de aplicaciones como el *Acrobat* de la empresa *Adobe*.

Google Trends: es una herramienta de Google que muestra los términos de búsqueda más populares del pasado reciente.

MobileMe: fue una suite en Internet diseñado por Apple con el objetivo de proveer de servicios de Internet al iPhone, Mac OS X, Windows, iPod Touch e iPad. Fue sustituido por iCloud.

WebDAV: (*Web-based Distributed Authoring and Versioning*) o Autoría y Versionado Distribuidos basados en Web es un conjunto de extensiones del protocolo HTTP que permiten a los usuarios editar y manejar ficheros en colaboración utilizando un servidor web remoto.

SMS: (*Short Message Service*) o Servicio de Mensajes Cortos es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

Mozilla: Es un proyecto de software libre que desarrolla soluciones de código abierto para la World Wide Web.

Mozilla Firefox: es un navegador web libre, con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios externos.

ISO: (*International Organization for Standardization*) Organización Internacional para la Estandarización creada en 1947, es una organización internacional no gubernamental, que redacta y aprueba normas técnicas internacionales. Dichas normas se conocen como normas ISO.

IETF: (*Internet Engineering Tasking Force*) Estándar internacional para certificar la calidad de un software.

HTML: (*Hyper Text Markup Language*) Lenguaje de Marcado de Hipertexto diseñado para estructurar textos y relacionarlos en forma de hipertexto.

XHTML: (*Extensible Hyper Text Markup Language*) Lenguaje de Marcado de Hipertexto Extensible es una versión más estricta y limpia de HTML, que nace precisamente con el objetivo de reemplazar a HTML ante su limitación de uso con las cada vez más abundantes herramientas basadas en XML.

XML: (*Extensible Markup Language*) Lenguaje de Marcado Extensible es un lenguaje desarrollado por el World Wide Web Consortium (W3C).

RSS: (*Really Simple Syndication*) es una familia de formatos para syndicar contenidos, especificado en XML. El RSS es usado por sitios de noticias, blogs, etc.

URL: (*Universal Resource Locator*) Localizador Universal de Recursos es el mecanismo con el cual el World Wide Web (www) asigna una dirección única a cada uno de los recursos de información de cualquier lugar de internet.

JavaScript: Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

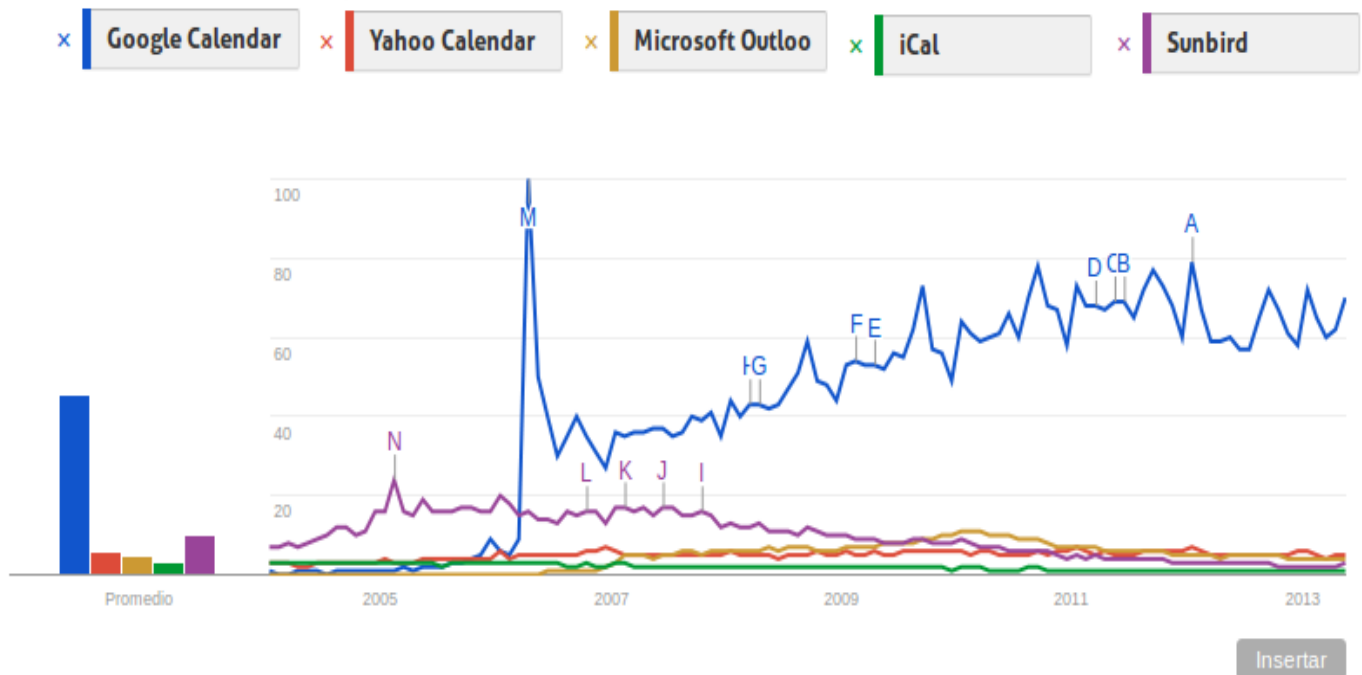
DOM: (*Document Object Model*) Modelo en Objetos para la representación de Documentos. Es un modelo computacional a través de la cual los programas y scripts pueden acceder y modificar dinámicamente el contenido, estructura y estilo de los documentos HTML y XML.

API: (*Application Programming Interface*) Interfaz de Programación de Aplicaciones es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

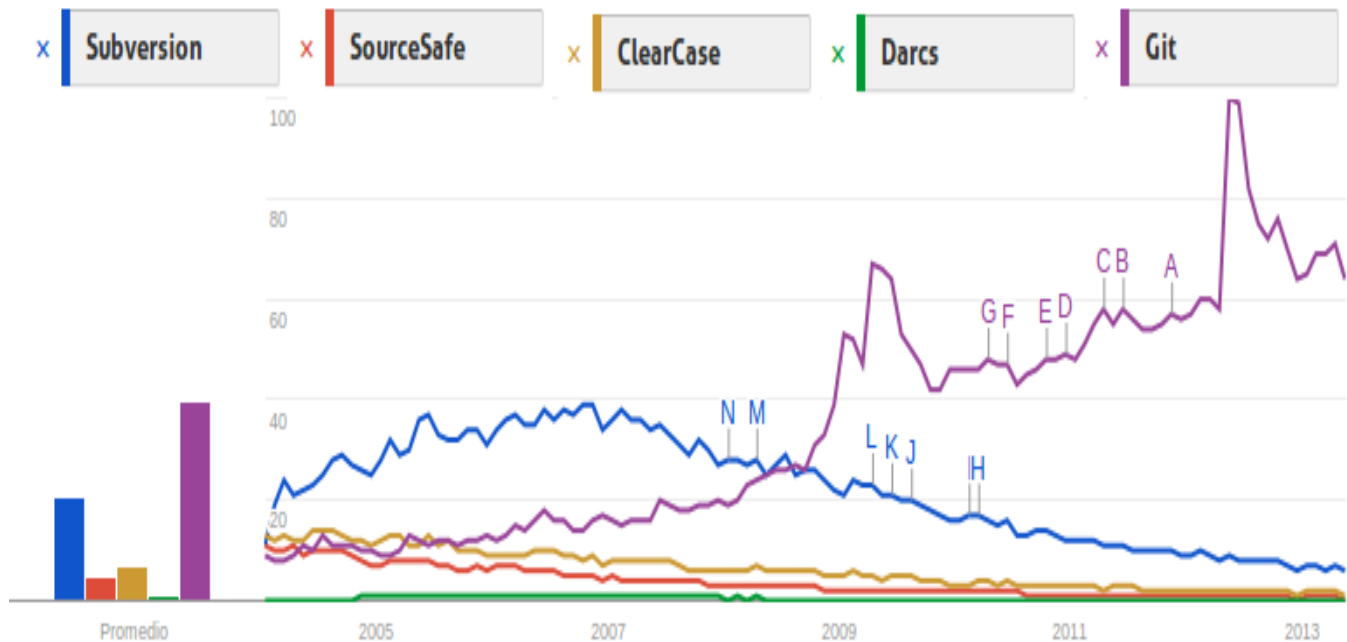
UNO: (*Universal Network Objects*) Objetos de Red Universales es el modelo de componentes utilizado por las suites ofimáticas LibreOffice y OpenOffice.org.

Anexos

Anexo 1: Tendencia al uso de herramientas gestoras de calendario.



Anexo 2: Tendencia al uso de herramientas de control de versiones.



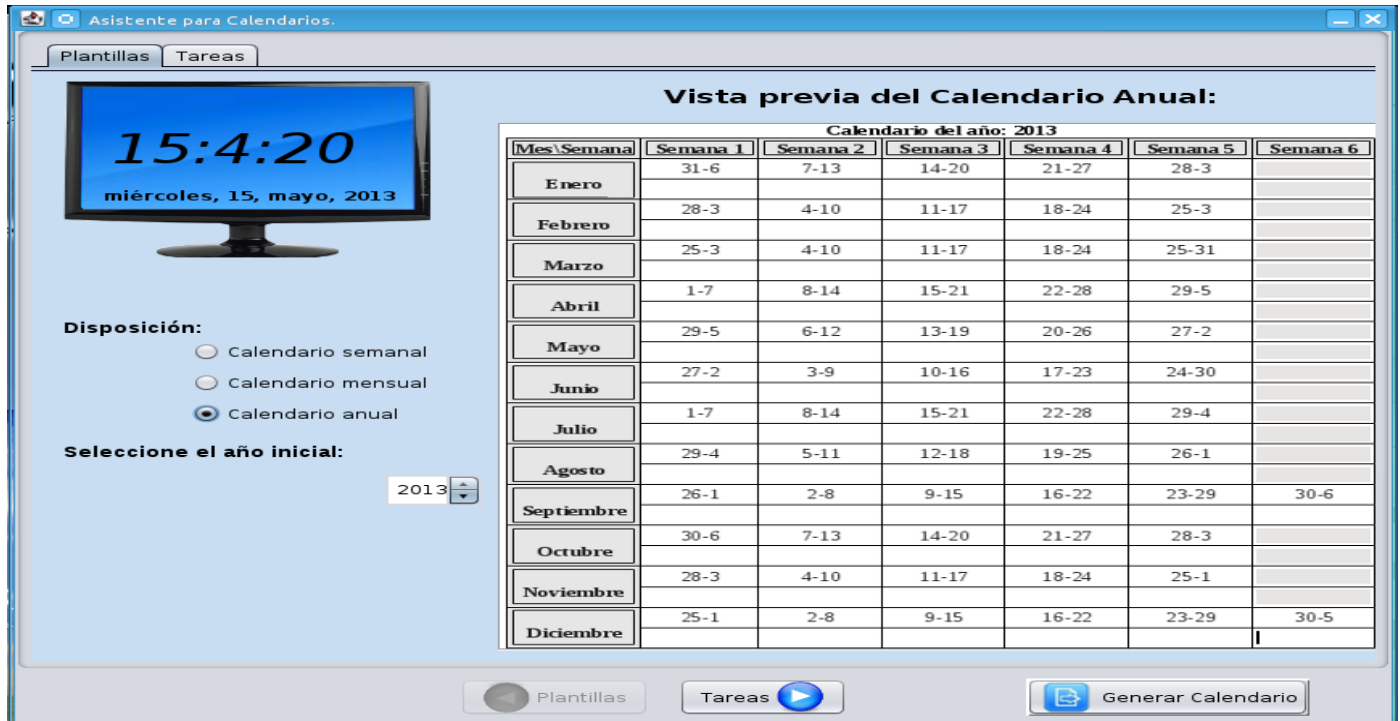
Anexo 3: Interfaz correspondiente a la HU-01.



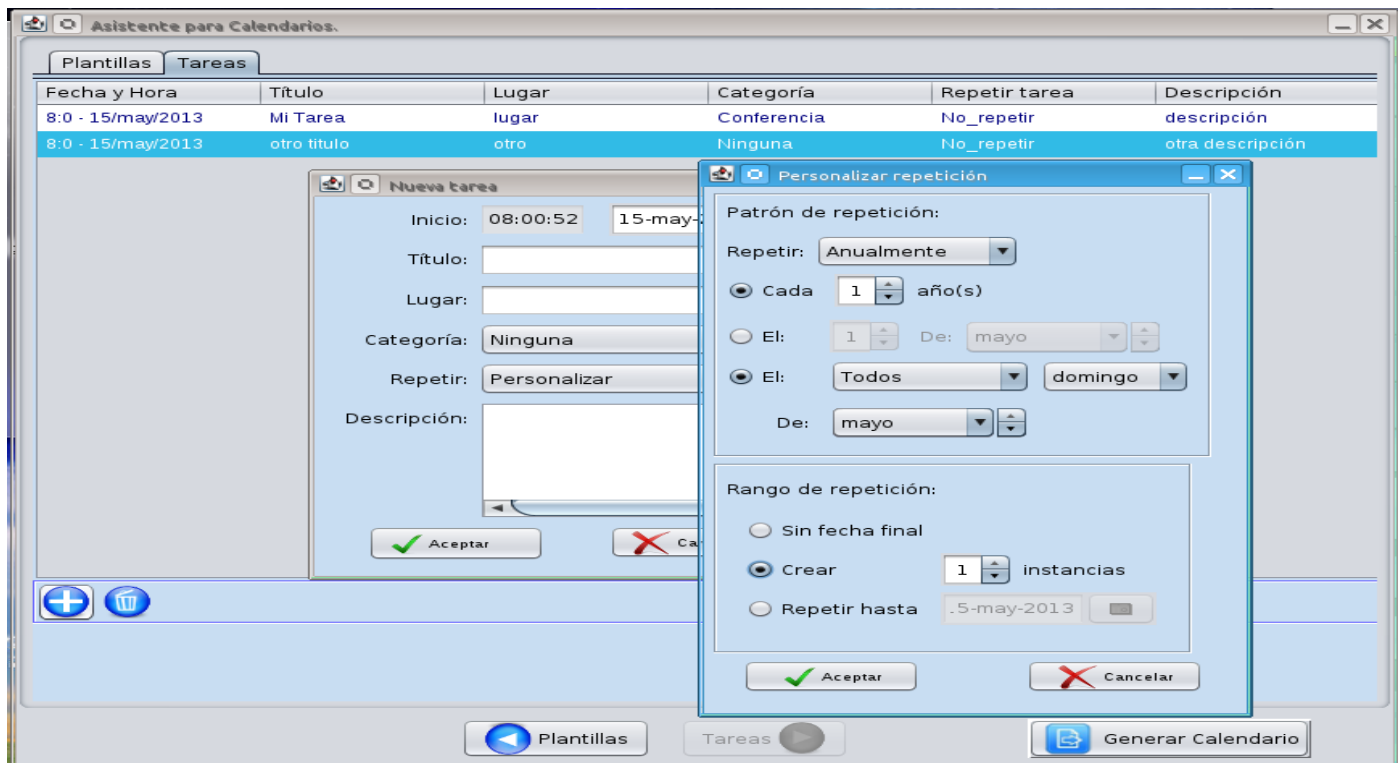
Anexo 4: Interfaz correspondiente a la HU-02.



Anexo 5: Interfaz correspondiente a la HU-03.



Anexo 6: Interfaz correspondiente a la HU-04.



Anexo 7: Interfaz correspondiente a la HU-05.

2013 Mayo: Semana 3

MIÉRCOLES 13/5/2013	JUEVES 16/5/2013
13:30 En: Aula 303. Predefensa	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.
MARTES 14/5/2013	VIERNES 17/5/2013
8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen. 9:0 Entregar documento tesis : Entregar el documento a los miembros del tribunal.
MIÉRCOLES 15/5/2013	SÁBADO 18/5/2013
8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.	8:0 Vacaciones En: Varadero. Viaje a la Playa
	DOMINGO 19/5/2013
	8:0 Vacaciones En: Varadero. Viaje a la Playa

Anexo 8: Interfaz correspondiente a la HU-06.

Mayo de 2013

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
29	30 Abril	1 Mayo	2	3	4	5
		4:0 En: Plaza de la Revolución. Marcha por el 1ro Mayo	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis.	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis.		
6	7	8	9	10	11	12
8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis.	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis. 8:0 Proyecto En: Laboratorio 103. Realizar pruebas a la aplicación	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis. 8:0 Proyecto En: Laboratorio 103. Realizar pruebas a la aplicación	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis. 8:0 Proyecto En: Laboratorio 103. Realizar pruebas a la aplicación	8:0 Proyecto En: Laboratorio 103. Concluir con el desarrollo de la tesis.		
13	14	15	16	17	18	19
13:30 En: Aula 303. Predefensa	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen.	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen. 9:0 Entregar documento tesis : Entregar el documento a los miembros del tribunal.	8:0 Vacaciones En: Varadero. Viaje a la Playa	8:0 Vacaciones En: Varadero. Viaje a la Playa
20	21	22	23	24	25	26
8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen. 10:0 Viaje En: La Habana. Salir de compras : Comprar el vestuario para la tesis	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen. 8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 En: Laboratorio 103. Arreglar documento : Corregir resumen. 8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia
27	28	29				
8:0 Viaje En: <u>Sancti-Spiritus</u> . Ver a la familia	8:0 Viaje En: <u>Cienfuegos</u> . Ver a la familia					

Anexo 9: Interfaz correspondiente a la HU-07.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

P

Calendario del año: 2013

Mes/Semana	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Enero	31-6	7-13	14-20	21-27	28-3	
Febrero	28-3	4-10	11-17	18-24	25-3	
Marzo	25-3	4-10	11-17	18-24	25-31	
Abril	1-7	8-14	15-21	22-28	29-5	
Mayo	29-5	6-12	13-19	20-26	27-2	
			1, 2, 2, 2	2, 2, 2, 3, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	
Junio	27-2	3-9	10-16	17-23	24-30	
	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	
Julio	1-7	8-14	15-21	22-28	29-4	
	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	2, 2, 2, 2, 2, 2, 2	

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1

Buscar

Anexo 10: Interfaces correspondientes a la HU-08.

Vista previa del Calendario Semanal:

2013 Abril: Semana 1

LUNES 14/2013	JUEVES 4/4/2013
MARTES 24/2013	VIERNES 5/4/2013
MIÉRCOLES 3/4/2013	SÁBADO 6/4/2013
	DOMINGO 7/4/2013

Vista previa del Calendario Mensual:

Mayo de 2013

Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
29	30 Abril	1 Mayo	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31 Mayo	1 Junio	2

Vista previa del Calendario Anual:

Calendario del año: 2013

Mes\Semana	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6
Enero	31-6	7-13	14-20	21-27	28-3	
Febrero	28-3	4-10	11-17	18-24	25-3	
Marzo	25-3	4-10	11-17	18-24	25-31	
Abril	1-7	8-14	15-21	22-28	29-5	
Mayo	29-5	6-12	13-19	20-26	27-2	
Junio	27-2	3-9	10-16	17-23	24-30	
Julio	1-7	8-14	15-21	22-28	29-4	
Agosto	29-4	5-11	12-18	19-25	26-1	
Septiembre	26-1	2-8	9-15	16-22	23-29	30-6
Octubre	30-6	7-13	14-20	21-27	28-3	
Noviembre	28-3	4-10	11-17	18-24	25-1	
Diciembre	25-1	2-8	9-15	16-22	23-29	30-5

Anexo 11: Estándar de codificación utilizado.

1. Deben evitarse los ficheros de gran tamaño que contengan más de 1000 líneas.
2. Los métodos deben agruparse por funcionalidad en lugar de agruparse por ámbito o accesibilidad. Por ejemplo, un método privado puede estar situado entre dos métodos públicos. El objetivo es desarrollar código fácil de leer y comprender.
3. Como norma general se establecen 4 caracteres como unidad de sangría. Los entornos de desarrollo integrado (IDE) más populares, tales como Eclipse o Netbeans que es el que se utiliza para el desarrollo del asistente, incluyen facilidades para formatear código Java.
4. La longitud de línea no debe superar los 80 caracteres por motivos de visualización e impresión.
5. Cuando una expresión ocupe más de una línea, esta se podrá romper o dividir en función de los siguientes criterios:
 - Tras una coma.
 - Antes de un operador.
 - Alinear la nueva línea con el inicio de la expresión al mismo nivel que la línea anterior.

Ejemplos:

```
unMetodo(expresionLarga1, expresionLarga 2, expresionLarga 3,
          expresionLarga 4, expresionLarga 5);
if ((condicion1 && condicion2)
    || (condicion3 && condicion4)
    ||!(condicion5 && condicion6)) {
unMetodo();
}
```

6. Los comentarios deben contener el siguiente formato:

- Comentarios de bloque. Permiten la descripción de ficheros, clases, bloques, estructuras de datos y algoritmos.

```
/*  
 * Esto es un comentario  
 * de bloque  
 */
```

- Comentarios de línea. Son comentarios cortos localizados en una sola línea. Si ocupa más de una línea se utilizará un comentario de bloque.

```
/* Esto es un comentario de línea */  
// Esto es otro comentario de línea
```

- Comentario al final de la línea. Comentario situado al final de una sentencia de código y en la misma línea.

```
int contador = 0; // Inicialización del contador  
contador++; /* Incrementamos el contador */
```

7. Toda variable local tendrá que ser inicializada en el momento de su declaración, salvo que su valor inicial dependa de algún valor que tenga que ser calculado previamente.

Ejemplo:

```
int idUnidad = 1;  
String[] funciones = { "Administración", "Intervención", "Gestión" };
```

8. Las declaraciones deben situarse al principio de cada bloque principal en el que se utilicen, y nunca en el momento de su uso. La única excepción a esta regla son los índices de los bucles "for", ya que, en Java, pueden incluirse dentro de la propia sentencia "for".

Ejemplo:


```

public void unMetodo() {
    int contador = 0; // inicio del método
    ...
}
for (int i=0; contador<10; i++) {
int contador1 = 0;
...
}

```

9. Se debe evitar el uso de declaraciones que oculten a otras declaraciones de ámbito superior.

```

int contador = 0; // Inicio del método
public void unMetodo() {
    if (condicion) {
        int contador = 2; // ii EVITAR !!
        ...
    }
    ...
}

```

10. Cada línea debe contener como máximo una sentencia.

Ejemplo:

```

int contador++;
int variable--;

```

11. Todas las sentencias de un bloque deben encerrarse entre llaves "{ ... }", aunque el bloque conste de una única sentencia. Esta práctica permite añadir código sin cometer errores accidentalmente al olvidar añadir las llaves.

Ejemplo:

```
if (condicion) {  
    variable++;  
}
```

12. La sentencia "try/catch" siempre debe tener el formato siguiente:

```
try {  
    sentencias;  
} catch (ClaseException e) {  
    sentencias;  
}
```

13. Los nombres de clases deben ser sustantivos y deben tener la primera letra en mayúsculas. Si el nombre es compuesto, cada palabra componente deberá comenzar con mayúscula.

Ejemplo:

```
class Calendario  
class CalendarioAnual
```

14. Las variables se escribirán siempre en minúsculas. Las variables compuestas tendrán la primera letra de cada palabra componente en mayúsculas.

15. Las variables nunca podrán comenzar con el carácter "_" o "\$". Los nombres de variables deben ser cortos y sus significados tienen que expresar con suficiente claridad la función que desempeñan en el código.

Ejemplo:

```
Calendario calendario;  
String nombre;
```

16. Use minúsculas para constantes nativas de Java. Siempre use cadenas en mayúsculas para la definición de sus propias constantes.

Ejemplo:

```
boolean variable=true;  
int CONSTANTE;
```