

Universidad de las Ciencias Informáticas

Facultad 5



**Interfaz Visual para la configuración de
Entornos Virtuales desarrollados con la
Herramienta Scene Tool Kit.**

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

Autor(es): Leonardo Antonio Nieblas Palau
Yasmany Cubela Medina

Tutor: Ing. Fernando Jiménez López.
Co-Tutor: Ing. Yanoski Camacho Román.

Junio 2008

Haríamos muchas cosas si creyéramos que son muchas menos las imposibles

Mohandas Karamchand Gandhi

DECLARACIÓN DE AUDITORIA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Autores:

Leonardo Antonio Nieblas Palau

Yasmany Cubela Medina

Tutor:

Ing. Fernando Jiménez López

DATOS DE CONTACTO

Ing. Fernando Jiménez López.

Graduado en Ingeniería Informática, en la Ciudad Universitaria José Antonio Echevarría (CUJAE). Actualmente Profesor Instructor de la Universidad de la Ciencias Informáticas impartiendo la asignatura de Gráficos por Computadora además de ser el Jefe del Polo de Realidad Virtual perteneciente a la Facultad 5.

E-Mail: fjimenez@uci.cu

Ing. Yanoski Camacho Román

Graduado en Ingeniería Informática, en la Ciudad Universitaria José Antonio Echevarría (CUJAE). Actualmente Profesor Instructor de la Universidad de la Ciencias Informáticas impartiendo la asignatura de Sistema Operativo además de ser el Jefe del Proyecto “Herramienta de Desarrollo de Sistemas de Realidad Virtual” perteneciente a la Facultad 5.

E-Mail: rcamacho@uci.cu

DEDICATORIA

*A mis padres. A Milorys.
A toda mi familia.*

Yasmany

*A mis padres, a mi abuela.
A toda mi familia.*

Leonardo

AGRADECIMIENTOS

La elaboración de este trabajo ha estado guiada por muchas personas que directa o indirectamente han contribuido a su terminación, a todas ellas queremos hacerles llegar nuestros más sinceros agradecimientos. Agradecerle a la revolución y a nuestro comandante en jefe y siempre victorioso Fidel Castro Ruz, por crear la sociedad más justa del mundo y la oportunidad de escribir estos agradecimientos. De forma especial al Ing. Yanoski R. Camacho Román, dispuesto en todo momento a atender las dudas y propuestas que iban surgiendo y a nuestro tutor el Ing. Fernando Jiménez López por el constante apoyo.

De Yasmany

A todos los miembros de mi familia que siempre han estado a mi lado apoyándome a lo largo de todos estos años. En especial a mi mamá y papá cuyo esfuerzo en mi aprendizaje ha sido más que un deber de padres, también el hombro amigo de un eterno compañero. A Milorys, que con amor, y apoyo incondicional y constante, mantuvo aun en las malas la esperanza en mí.

De Leonardo

A mis padres, quienes siempre me guiaron en mí andar por la vida, por confiar y apoyarme siempre en mis decisiones. A mi abuela quien siempre me ha brindado su sabiduría, y ha sido fuente de inspiración para lograr los resultados que hoy he obtenido. A mi tía, mi tío y mis primos, quienes son parte muy importantes de mí y siempre me han brindado su apoyo y ayuda incondicional. A mi novia Aniuska, por haberme brindado su cariño, comprensión y apoyo en los momentos más difíciles. A todos mis amigos, quienes han estado al tanto de este trabajo en todo momento. A mi familia en general por todo el apoyo y confianza que han depositado en mí.

RESUMEN

Un punto fundamental en el desarrollo de sistemas de realidad virtual, es el hecho de configurar los diferentes estados de render que llevará el entorno virtual en cuestión, para esto la mayoría de las veces se logra de manera directa en la codificación de dichos sistemas. Es por ello que se hace necesario desarrollar una herramienta capaz de disminuir en gran medida este trabajo, de manera que se optimice el tiempo de realización de estas configuraciones.

Para darle cumplimiento a los objetivos de esta investigación, se realizó un análisis de las diferentes características sobre el diseño de Interfaces Gráficas de Usuario, así como las diferentes bibliotecas existentes para el desarrollo de las mismas en sistemas GNU/Linux. También fue necesario investigar acerca de las diferentes bibliotecas de desarrollo de sistemas de realidad virtual, así como las bibliotecas que poseen este mismo objetivo, con la especificidad de presentar Interfaces Gráficas de Usuario. Por último se analizó las diferentes ventajas que brinda el formato de fichero XML para el almacenamiento de información.

Como resultado final se obtuvo una Interfaz Gráfica de Usuario, capaz de optimizar el tiempo de configuración de los estados de render de los diferentes entornos virtuales, que serán utilizados en la realización de los sistemas de realidad virtual. Además de cumplir con las necesidades existentes en el proyecto Herramienta de Desarrollo de Sistemas de Realidad Virtual, proyecto perteneciente a la Facultad #5.

SUMMARY

The fact of set virtual environment render states is a key point in the development of virtual reality systems, which bring the desired effect in the virtual environment. The current configuration is relatively long and tedious, depending directly from the virtual environment and its complexity and demands of the end user. That is why it is necessary to develop a tool capable of greatly reducing the time spent on the process described above, so as to optimize this time.

To give effect to the objectives of this investigation was conducted an analysis of the different characteristics on the graphical user interfaces design, as well as the various existing libraries to develop them in GNU / Linux systems. It was also necessary research on the different libraries in developing virtual reality systems and tools which have the same goals. Finally, we analyze the various advantages offered by the XML file format for storing information.

As a final result was a graphical user interface that can optimize the rendering states configuration time in different virtual environments, which will be used in conducting virtual reality systems. In addition to complying with existing needs in the draft Development Tool of Virtual Reality Systems, a project owned by the School # 5.

TABLA DE CONTENIDOS

DEDICATORIA.....	I
AGRADECIMIENTOS	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1 La interfaz gráfica de usuario	5
1.1.1 ¿Qué es la interfaz grafica de usuario?	7
1.1.2 Clasificación de las interfaces de usuario.....	8
1.1.3 Características humanas del diseño de interfaces graficas de usuario.....	11
1.1.4 Principios para el diseño de interfaces graficas de usuario	12
1.1.5 Pasos para construir una interfaz grafica de usuario	17
1.2 Bibliotecas de desarrollo de sistemas de realidad virtual	20
1.2.1 Características de las bibliotecas de desarrollo de sistemas de realidad virtual más utilizadas	21
1.2.2 Bibliotecas de desarrollo de sistemas de realidad virtual en Cuba.....	25
1.3 Bibliotecas de desarrollo de sistemas de realidad virtual que poseen interfaces graficas de usuario	27
1.3.1 Principales características que presentan estas herramientas.....	27
1.4 Bibliotecas de desarrollo de interfaces graficas de usuario en sistemas GNU/Linux	29
1.4.1 Qt.....	29
1.4.2 Gtk.....	31
1.5 Uso del estándar XML como formato de almacenamiento de datos en los sistemas de realidad virtual	32
CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA.....	33
2.1 Descripción de la solución propuesta	33
2.1.1 XML como formato para el almacenamiento de datos.....	34
2.1.2 Método de selección de objetos en escenas 3d.....	37
2.1.3 Herramientas de desarrollo y lenguaje utilizado	38
2.2 Diseño del sistema.....	40

2.2.1	Reglas del negocio.....	40
2.2.2	Modelo de dominio	41
2.2.3	Captura de requisitos	42
2.2.4	Modelos de casos de uso	43
CAPITULO 3: ANÁLISIS Y DISEÑO DEL SOFTWARE		71
3.1	Modelo de clases de análisis.....	71
3.2	Diagrama de interacción	76
3.3	Diagrama de clases de diseño	87
3.4	Descripción de las clases de diseño en formato expandido	88
CAPITULO 4: IMPLEMENTACION Y PRUEBAS		100
4.1	Estándar de codificación	100
4.2	Diagrama de despliegue.....	100
4.3	Diagrama de componentes	100
4.4	Modelos de pruebas.....	103
CONCLUSIONES.....		110
RECOMENDACIONES.....		111
BIBLIOGRAFÍA CONSULTADA.....		112
REFERENCIAS BIBLIOGRÁFICAS		113
APÉNDICES		115
Glosario de Abreviaturas.....		115
Glosario de Términos.....		116
ÍNDICE DE FIGURAS		119
ÍNDICE DE TABLAS.....		121

INTRODUCCIÓN

El avance de la Informática en los últimos tiempos ha proporcionado la creación y extensión de un nuevo término: “Realidad Virtual”. Desde sus inicios ha tenido una amplia gama de aplicaciones, como, programas de entrenamiento militar, simuladores de diversos tipos, centros de investigación, entre otros, llegando a formar parte de una variedad de programas, tanto en el ámbito profesional como social.

Ésta se caracteriza por su capacidad de permitirle al usuario navegar por un mundo totalmente nacido de su creatividad, una realidad que solo está limitada por las fronteras de su imaginación. Siendo esta su principal característica y atractivo [14].

Podemos definir la realidad virtual como un conjunto de tecnologías que, apoyadas en las computadoras y medios electrónicos, simulan en tiempo real, representaciones de la realidad. Una realidad ilusoria, ya que solo existe dentro de un ordenador. Por lo que podemos decir que es una pseudorrealidad alternativa [15].

La realidad virtual básicamente consiste en simular todas las posibles formas de interacción de una persona con un mundo artificial. Siendo su principal característica frente a otros sistemas tradicionales la posibilidad de interactuar en tiempo real con este entorno creado artificialmente; sensaciones de movimiento, aceleración, cambiar de punto de vista, entre otras, deben ser presentadas al usuario de manera que se sienta inmerso en ese mundo generado por un ordenador, de manera tal que llegue al punto de sentir esta realidad alternativa como una realidad absoluta [17].

La realidad virtual tuvo sus inicios a finales de los años 70 como material para una clase de aviación en el Departamento de Defensa de los Estados Unidos, aunque no fue hasta finales de la década de los 80, cuando en el año 1989 Jaron Lanier se propone el término de realidad virtual [17].

Dentro de sus principales actividades se destacan:

- En 1980 la Compañía StereoGraphics desarrolla las Gafas Stereo.
- En 1982 Thomas Zimmerman patenta un Electroguante que inventó mientras investigaba sobre cómo controlar con la mano un instrumento musical virtual.
- En 1987 la Compañía Inglesa Dimensión Internacional desarrolla un Software de construcción de mundos tridimensionales sobre Computadoras Personales.

Introducción

- En 1988 Scott Foster inventa un dispositivo para la generación de sonido tridimensional.
- En 1989 ATARI saca al mercado la máquina de galería de video juegos con tecnología 3D. En este mismo año Autodesk desarrolla su primer sistema de realidad virtual para Computadoras Personales.
- Desde los años 90 y hasta nuestros días la realidad ha intensificado su uso en una gran cantidad de industrias, como, videos juegos, la industria cinematográfica, entre otras.

Muchos son los usos que se le da hoy en día a la realidad virtual, entre ellos se destacan el entrenamiento de doctores mediante simuladores, que permiten la práctica y el entrenamiento de una forma segura, en la preparación militar, de pilotos y soldados, en la industria química se realizan simulaciones de la acción de pesticidas y la creación de drogas, en la industria automotriz donde se hacen nuevos diseños y pruebas de prototipos [16].

Ayuda en la reconstrucción de escenas criminales, permitiendo esclarecer situaciones y agilizar los procesos, en la industria cinematográfica ha tenido diversos logros como la aplicación de efectos a escenas, como escenas propias de mundos virtuales. Todo esto deja como consecuencia una serie de beneficios [16]. Tales como:

- Entretenimiento en situaciones peligrosas.
- Permite un examen minucioso de hechos y procesos.
- Poner a prueba modelos y principios.
- Situación de una pauta de actuación.
- Visualización en 360 grados.
- Aprende a manipular objetos.

La realidad virtual al igual que muchas cosas no está exenta de algunas desventajas implicadas por su uso. Desventajas tales como:

- Complejidad de los desarrollos.
- Costes elevados en la realización de mundos virtuales.
- Deficiencias en el interfaz entre programas y usuarios.
- Desorientación espacial.
- Dificultad en dominar los mandos y controles.
- Distanciamiento emocional de los objetos y escenas virtuales.
- Servidumbre de los equipos y su mantenimiento.

Introducción

En Cuba, se ha ido desarrollando el uso de la Realidad Virtual con el pasar de los años, varios son los que se están incursionando en la investigación y desarrollo de este campo. La Universidad de las Ciencias Informáticas no ha quedado exenta de esto, por lo que se encuentra desarrollando algunos proyectos que llevan adelante la investigación y desarrollo de este tema. Específicamente en la Facultad #5, que es la facultad encargada de desarrollar esta línea de trabajo, donde actualmente se labora en varios proyectos, entre lo que podemos mencionar: Simuladores de Conducción, Simuladores Quirúrgicos, así como video juegos, entre otros.

Una de las problemáticas existentes a la hora de recrear los entornos virtuales, es la falta de herramientas que permitan hacer un manejo del entorno virtual de una manera visual, de modo que sea más sencilla la creación y modificación de las características de los objetos y del entorno en sí. Actualmente el proyecto Herramienta de Desarrollo para Sistemas de Realidad Virtual, proyecto que desarrolla una biblioteca de clases para el desarrollo de estos sistemas, no cuenta con una Interfaz Gráfica propia que permita la Gestión de Configuraciones en los Entornos Virtuales, y disminuya el tiempo de desarrollo de estos así como su manejo, por lo que esto constituye la situación problemática existente. De ahí que surja la siguiente interrogante, que constituye el **problema científico** de nuestro trabajo: ¿Cómo facilitar la configuración de Entornos Virtuales que serán desarrollados con la biblioteca gráfica SceneToolkit?

Como **objeto de estudio**: El trabajo en los procesos de desarrollo de interfaces visuales, y el **campo de acción**: Se basa en el estudio del desarrollo de interfaces visuales para herramientas de desarrollo de sistemas de realidad virtual.

El **objetivo general** que propone ésta investigación es: desarrollar una herramienta con interfaz visual, para la configuración de los entornos virtuales utilizados en los sistemas de realidad virtual con base en la biblioteca gráfica SceneToolkit y brindar flexibilidad e integración de herramientas empotradas.

En esta investigación nos proponemos como **objetivos específicos** desarrollar una herramienta con una interfaz gráfica de usuario amigable al usuario, la cual muestre entornos de realidad virtual y permita la selección de objetos dentro de dicho entorno; gestionar archivos de configuración para entornos virtuales que serán desarrollados sobre la biblioteca gráfica Scene Tool Kit; y por ultimo crear una arquitectura flexible que soporte la adición de plugins.

Introducción

A continuación se plantean un grupo de tareas que permitirán satisfacer durante el desarrollo de la investigación los objetivos planteados anteriormente, y se muestran de la forma siguiente:

- Indagar acerca de las principales técnicas de desarrollo de interfaces gráficas de usuario.
- Investigar sobre bibliotecas de desarrollo vinculadas a la realidad virtual
- Investigar sobre herramientas con interfaz visual para bibliotecas de desarrollo de entornos virtuales.
- Investigar sobre el formato de almacenamiento de datos XML.
- Investigar sobre bibliotecas de desarrollo de interfaces visuales en GNU/Linux.
- Investigar sobre los distintos algoritmos de selección en escenas 3D.
- Diseño de la solución propuesta.
- Implementación de la solución propuesta.

Se presentará una breve descripción del contenido de los capítulos:

- Capítulo 1 “Fundamentación Teórica”, hace un análisis de las principales funcionalidades para la configuración de los entornos virtuales utilizados en los sistemas de realidad virtual elaborados en la Facultad 5.
- Capítulo 2 “Características del Sistema”, presenta una descripción de las funcionalidades y bondades que brindara la herramienta.
- Capítulo 3 “Análisis y Diseño del Sistema”, se muestran los artefactos propuestos por RUP según la Ingeniería de Software.
- Capítulo 4 “Implementación y Prueba” muestra una descripción de los casos de uso de prueba, así como algunos resultados obtenidos.

Finalmente se muestra un glosario de términos que ayuda al entendimiento del lenguaje técnico que se ha utilizado.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 La interfaz gráfica de usuario

Con el desarrollo y los avances de la Ciencia y la Tecnología, el hombre se ha ubicado en un plano intermedio entre lo tangible y lo intangible computacionalmente hablando, en nuestros días es muy común el convivir con un ordenador, es por ello que cada vez se hace más imperativo una mejor interacción hombre-computadora a través de una adecuada Interfaz Gráfica de Usuario que brinde tanto comodidad como eficiencia.

Las Interfaces Gráficas de Usuarios surgen dada la necesidad de hacer más simple el uso de los ordenadores para todo tipo de usuarios y no solo restringir el uso de estos a usuarios avanzados, llegando así a convertirse en un hábito el usar Interfaces Gráficas de Usuario en inglés Graphics User Interface (GUI), de manera que le sea más fácil la interacción del usuario final con el ordenador.

Como bien afirma Neal Stephenson: “en el principio fue la línea de comandos...”; la mayoría de los ordenadores domésticos requerían de conocimientos de BASIC, ya que el 95% de estos mostraban el intérprete BASIC al encenderse, u ofrecían una interfaz de línea de comandos, lo que provocaba a los usuarios tener conocimientos por encima de la media si deseaban hacer algo más que no fuera utilizar éstos como consolas de videojuegos. En nuestros días prima la cultura de la Interfaz amigable y vistosa, en la que con un simple clic sobre algún componente gráfico que esté presente en pantalla, se sustituye la tediosa tarea de escribir código fuente para que el ordenador interprete que debe realizar alguna acción.

Precursores de las Interfaces Gráficas de Usuario

- **Xerox Alto**

Este fue desarrollado en el Xerox PARC en 1983, este fue el primer ordenador personal, así como el primero que utilizó la metáfora de escritorio y una Interfaz Gráfica de Usuario [\[1\]](#).

- **Xerox Star 8010**

Esta estación de trabajo, conocida oficialmente como el “Sistema de Información Estrella 8010” en inglés “8010 Star Information System” fue introducida por Xerox Corporation en 1981.

Este constituyó el primer sistema comercial que incorpora varias tecnologías que hoy en día son comunes en los ordenadores personales, incluyendo la pantalla con Bitmaps en lugar de solo texto, una Interfaz Gráfica de Usuario basada en Ventanas, Iconos, Carpetas, Mouse, Red Ethernet, Servidores de Archivos, Impresoras y E-Mail [2].

- **Apple Lisa, Macintosh, Apple II GS**

Tras una visita realizada al Xerox PARC en 1979, el equipo de Apple, encabezado por Jef Raskin, se centra en diseñar un entorno gráfico para su nueva generación de PC de 16bit, el cual estaría presente en el Apple Lisa cuyo lanzamiento sería en 1983. Este sistema gráfico fue portado al sucesor de Apple II, el Apple II GS. Paralelo a esto un equipo de trabajo que se encontraba desarrollando el Apple Macintosh que no vería la luz hasta 1984, este incluiría una versión mejorada del entorno gráfico del Apple Lisa. Desde ese entonces el Mac reina como paradigma de usabilidad de un Entorno Gráfico; pese a que los cambios que se le realizarían al Sistema Operativo serían transparentes al usuario [2].

- **Apple y Windows**

La Compañía Apple, quien a principios de los años 80 (en 1982), había comenzado como una microempresa integrada por dos empleados (Steve Jobs y Steve Wozniak) había crecido hasta llegar a convertirse en una empresa de 300 millones de dólares.

Un año después, en 1983, Apple ya se convertiría en una empresa de 1000 millones de dólares, llegando a alcanzar el mismo valor que IBM.

En 1987 IBM se ve obligada a entrar en el mercado de los Ordenadores Personales con entornos Gráficos con su modelo PS/2, a su vez, aliándose con Bill Gates (Microsoft), quien había desarrollado el OS/2. Este Sistema Operativo presentaba una Interfaz Gráfica muy similar a la de Apple, pero el OS/2 no se convirtió en el nuevo estándar del sector, debido a los conflictos de intereses existentes entre IBM y Microsoft.

En el año 1985, Microsoft lanza al mercado el Windows 1.0, entorno gráfico para ordenadores PC IBM compatibles, este presentaba un parecido asombroso al Mac OS. Lo cual provocó que Apple en contrapartida realizara una demanda contra Microsoft, ascendiendo a varios miles de millones de dólares, ya que este había incurrido en violaciones de Copyright.

La aparición de Ordenadores IBM clónicos provocó que el Sistema Windows se popularizara, lo que provocó que Apple perdiera mercado, no siendo hasta finales de 1990 que esta se recuperó con el lanzamiento de nuevos productos.

En el año 1991, John Sculley, director de Apple, reveló que la compañía estaba considerando competir contra Microsoft en el campo de Software vendiendo su Sistema Operativo a terceros. Este reveló mantener conversaciones con su antiguo rival IBM, conversaciones encaminadas a compartir tecnologías. Ambos en conjunto decidieron crear una joint venture para el desarrollo de un sistema Operativo avanzado que ambas empresas utilizarían en sus nuevos Ordenadores y licenciarían a terceros. Este plan constituía un desafío directo a Microsoft.

Microsoft sobreponiéndose a todo esto consiguió convertir su sistema Operativo Windows en el Sistema operativo más utilizado a nivel mundial, dejando así a Apple en segundo lugar[2].

1.1.1 ¿Qué es la interfaz grafica de usuario?

El concepto de Interfaz se ha definido ampliamente desde diferentes puntos de vista, según el ámbito de conocimientos en que se encuentre aplicándose este concepto. Por ejemplo en la biología ha sido definido como la capa de un organismo que separa su interior del exterior, en la electrónica y las telecomunicaciones, se ha definido como puerto a través del que se envían o reciben señales desde un sistema de subsistemas hacia otros, en química es la superficie existente entre dos fases distintas en una mezcla heterogénea [18].

Recurriendo a la etimología de la palabra interfaz, encontramos que esta es una palabra compuesta por dos vocablos. Inter y Faz, Inter proviene del latín inter, y significa, "entre" o "en medio", y faz proviene del latín facies, y significa "superficie", "vista o lado de una cosa". Por lo que una traducción literal del concepto de Interfaz atendiendo a su etimología sería, "superficie vista, o lado mediador" [2]. Visto desde la interacción hombre-ordenador, hablamos de interfaz de usuario, haciendo referencia a la forma genérica del espacio que media relación de una persona y un ordenador o sistema interactivo. La interfaz de usuario es la llamada "ventana mágica" de un sistema informático, y que le brinda la posibilidad a una persona de interactuar con él [18].

Cuando hablamos de Interfaz Gráfica de Usuario, estamos en presencia de un concepto más específico, al contrario del concepto de Interfaz, tiene una localización determinada y definida. Si el concepto de interfaz se define etimológicamente como la cara o superficie mediadora, el concepto de interfaz gráfica de usuario, supone un tipo específico de interfaz, que utiliza metáforas visuales y gráficos como paradigma interactivo entre el usuario y el ordenador [18].

Una definición formal de interfaz gráfica de usuario es la descrita por la Wikipedia americana, la cual la define como: “un método de interacción con un ordenador a través del paradigma de manipulación directa de imágenes gráficas, controles y texto” [18].

1.1.2 Clasificación de las interfaces de usuario

Dentro de las clasificaciones de Interfaz de Usuario se pueden distinguir básicamente dos tipos fundamentales [1]:

- Interfaces de Hardware, la cual es utilizada a nivel de dispositivos, utilizados para ingresar, procesar y entregar los datos. Ejemplo: Teclado, Ratón y Pantalla.
- Interfaces de Software, está destinada a entregar información acerca de los procesos y herramientas de control a través de lo que el usuario observa habitualmente en la pantalla.

A partir de estas clasificaciones generales se pueden ir desprendiendo algunas más, así por ejemplo a partir de su evolución tenemos:

La evolución de las Interfaces de Usuario marchan paralelamente al desarrollo de los Sistemas Operativos, actualmente la interfaz de usuario constituye uno de los principales componentes de los Sistemas Operativos.

Constituyen algunas de las interfaces de usuario que a lo largo de la historia han ido apareciendo, y a la vez ejemplificándolas con las sucesivas versiones de los Sistemas Operativos más populares [1].

Interfaces de líneas de mandatos (command-line user interfaces, CUIs)

Comúnmente conocido como DOS, fue el sistema operativo de los primeros PCs, y forma a su vez el estilo de interacción hombre-ordenador más antiguo. En este el usuario le introducía órdenes y mandatos al ordenador haciendo uso de un lenguaje formal con un vocabulario y una sintaxis propia.

Estas órdenes eran introducidas típicamente por teclado, y estaban encaminadas a realizar una acción.

Estos tipos de interfaces de usuario son utilizadas comúnmente, ya que no suelen brindar al usuario mucha información por parte del sistema, y debe conocer a profundidad cual es el funcionamiento del ordenador, así como la ubicación de los programas. El modelo de la interfaz es el del programador y no el del usuario [1].

Inconvenientes

Representa una carga de memoria para el usuario, ya que debe memorizar los mandatos, siendo inclusive la ayuda de difícil lectura y entendimiento, no siempre presenta nombres adecuados a las funciones que realiza, el significado de los mandatos presenta un difícil entendimiento, ya que varios de estos presentan los mismos o parecidos significados (DEL - ERASE), y es inflexible a los nombres (DEL - DELETE).

Ventajas

Es una interfaz potente, flexible y es controlada por el usuario. Estas solo representan ventajas para usuarios con un alto nivel de experiencia en su uso, ya que la sintaxis es estricta y los errores pueden resultar graves.

Interfaces de Menú

Básicamente un menú es una lista de opciones que se muestran en la pantalla o en una ventana de la pantalla para que los usuarios elijan la opción que deseen. Estos permiten dos cosas [3]:

- Navegar dentro de un sistema, mostrando los caminos que guían de un sitio a otro.
- Seleccionar elementos de una lista, los cuales representan propiedades o acciones que los usuarios desean realizar sobre algún objeto.

Se puede decir que estamos frente a este tipo de interfaz de usuario cuando el ordenador se torna una herramienta de usuario y no solo de programadores. Actualmente las interfaces gráficas u orientadas a objetos continúan haciendo uso de este tipo de interfaces.

Este tipo de Interfaces de Usuario hace referencia a distintos tipos de menú, entre los que podemos encontrar [1]:

- **Menú de Pantalla completa**

Este tipo de menú se distingue por su ubicación en la pantalla, están situados en la parte superior de la pantalla, y son muy utilizados en las aplicaciones actuales. Cuentan con una lista de acciones genéricas que dan paso a menús desplegables donde se concretan.

- **Menú de barra y menú desplegable**

Pueden llevar a su vez a otros menús, este tipo es mayormente conocido como menú en cascada. Estos pueden realizar cambios en ellos mismo de manera dinámica, así como deshabilitar opciones que no estén disponibles para un contexto dado. Ejemplo: Menús en cascada de la barra de inicio de Windows 95.

- **Paletas o barras de herramientas**

Son menuces gráficos que contienen acciones, herramientas y opciones que se pueden colocar en la pantalla. Son muy utilizados en programas gráficos, como ejemplo se pueden apreciar las paletas de herramienta del Microsoft PowerPoint.

- **Menús Contextuales (Pop-Up)**

Estos tipos de menús son los más recientes, adoptan este nombre ya que el contenido del menú depende del contexto de trabajo del usuario. Solo contienen las opciones que se le pueden aplicar al objeto seleccionado, además de otras opciones de uso frecuente, las cuales también pueden ser accesibles desde el menú de barra. Ejemplo: El menú contextual del icono en el escritorio de Windows.

En resumen, este tipo de interfaz de usuario, siempre y cuando se encuentren bien estructuradas, son bien recomendadas para usuarios noveles o principiantes. Son fáciles de aprender y de recordar. Estas pueden variar desde menús muy simples hasta menús muy avanzados.

Precauciones: No ocupar demasiado espacio de la pantalla, recordar la información acumulada de menús precedentes, no colocar demasiados elementos en el menú, así como agrupar estos de manera lógica, permitir la personalización por parte del usuario, hacer uso de una terminología adecuada y

consistente dentro de la aplicación, así como con otros programas (Exit, Quit, Escape, Clore, Return, Back). Estas interfaces son comúnmente utilizadas en conjunción con otros tipos de interfaces.

Interfaces Gráficas de Usuario (graphical user interfaces, GUIs)

Como ya se ha comentado anteriormente estas fueron desarrolladas originalmente por XEROX, aunque alcanzaron su popularidad gracias a la compañía Apple. Existen tres estilos principales de interfaces gráficas hombre-ordenador, los cuales son [1]:

1. Lo que tú vez es lo que puedes conseguir (WYSIWYG, What you see is what you get).
2. Manipulación directa.
3. Interfaces de usuario basadas en íconos.

Básicamente un Interfaz Gráfica de Usuario es una representación gráfica en una pantalla de ordenador de los programas, datos y objetos, así como la interacción con ellos. Esta proporciona al usuario las herramientas necesarias para realizar sus operaciones, más que una lista de operaciones que el ordenador es capaz de hacer.

1.1.3 Características humanas del diseño de interfaces gráficas de usuario

Factores Humanos:

Al diseñar Interfaces de Usuarios se deben tener presente las habilidades cognitivas y de percepción de las personas, y adaptar el programa de ellas. Así una de las cosas más importantes que estas Interfaces pueden hacer es reducir la dependencia de las personas de su propia memoria, no forzándoles a recordar cosas innecesariamente (Ejemplo: información contenida en una pantalla anterior) o a repetir operaciones ya realizadas (Ejemplo: Introducir un dato repetidas veces).

Las personas tienen habilidades muy distintas a las de un ordenador, y ésta debe utilizar las suyas para soslayar las de aquellas, como por ejemplo la escasa capacidad de la memoria de corto alcance [1].

- **Velocidad de Aprendizaje:** La cual está referida a la velocidad con que el usuario aprende a utilizar el sistema. Es decir, pretende que el usuario minimice este tiempo de aprendizaje.
- **Velocidad de Respuesta:** Es el tiempo necesario en que se realiza una operación en el sistema.
- **Tasa de Errores:** Esta define el porcentaje de errores que comete el usuario.
- **Retención:** Es la capacidad de retención de información que presenta un usuario sobre un sistema en un periodo de tiempo.

- Satisfacción: Está referida a conocer en qué medida un usuario está satisfecho con un sistema dado.

Adecuación [1]:

1. Características Físicas: Cada persona difiere en cuando a características físicas respecto a otras, existen personas que no les gustan los teclados mientras que a otras sí. Es por ello que existen teclados ergonómicos. Esto mismo ocurre con el mouse.
2. Ambiente: Este se refiere al lugar donde va a ser utilizado el sistema, es por ello que cada Interfaz debe de estar adecuada al lugar de su utilización.
3. Visibilidad: Toma en cuenta el grado de iluminación del lugar donde se pondrá en práctica el sistema, dando así una medida del grado de reflejo del brillo en la pantalla.
4. Personalidad: De acuerdo a la edad, nivel socio-económico, etc.
5. Cultura: Se debe de tener en cuenta el factor de mercado para el sistema a nivel internacional, pues depende del nivel cultural del adquiridor, el ambiente que debe mostrar la Interfaz, por ejemplo, los japoneses no utilizan las mismas pantallas, ventanas, etc. que otros países u regiones.

Motivación [1]:

- Sistemas Vitales: Son sistemas referidos a aquellos sistemas que son de vida o muerte, es decir muchas personas dependen de ellos. Un ejemplo práctico es la utilización de un sistema para reactores nucleares, estos sistemas trabajan en tiempo real, y es de suma importancia la seguridad y efectividad del mismo.
- Sistemas Comerciales e Industriales: Estos se utilizan para elevar la productividad y realizar mayores cantidades de ventas.
- Sistemas de Oficina, Hogar y Juegos: En este influye un factor importante, que es el mercado a quien está dirigido, ya que estos tienen que ser muy amigables y satisfacer al cliente.
- Sistemas de Investigación: Realizan tareas muy específicas y tratan de imitar el medio en el que se desenvuelve el usuario.

1.1.4 Principios para el diseño de interfaces graficas de usuario

Para el diseño de las Interfaces Gráficas de Usuarios existen un conjunto de principios, que, a la hora de desarrollar este tipo de Interfaces se deben de tomar en cuenta para lograr la obtención de una Interfaz Gráfica adecuada y de buena aceptación por parte de los usuarios. Entre los más importantes podemos destacar:

La Anticipación:

Plantea la anticipación de las aplicaciones a las necesidades de los usuarios finales, haciéndole el trabajo más simplificado al usuario, ya que este no tendría que centrarse en la búsqueda y recopilación de información, o invocar las herramientas que va a utilizar [19].

La Autonomía:

Está referida a la disponibilidad que debe brindar la Interfaz Gráfica y el Entorno de Trabajo al usuario. Debe sentirse inmerso en un ambiente flexible, donde sea capaz de aprender rápidamente el uso de la aplicación. Sin embargo está comprobado que el entorno de trabajo debe contar con algunas características específicas, es decir, debe ser explorable pero no azaroso [19].

La Percepción del Color:

Básicamente trata la idea de que se deben utilizar además de los colores convencionales en las Interfaces de Usuario, vías secundarias para hacerle llegar la información pertinente a aquellos usuarios que presentan deficiencias en la visualización de colores. Ejemplo de esto sería la introducción de comandos abreviados (Shortcut-keys), lo cual constituye la visualización de combinaciones de teclas, que el usuario puede utilizar para acceder más rápidamente a las herramientas requeridas [19].

Valores por Defecto:

El término "Defecto", no debe ser utilizado en aplicaciones o servicios. Este puede ser reemplazado por alguna otra palabra o término, de manera que le dé al usuario una visión de lo que desea hacer, o está sucediendo. Los valores por defecto deberían ser opciones inteligentes y sensatas, además de que estos deben ser fáciles de modificar [19].

Consistencia:

Para que las Interfaces de Usuario obtengan una elevada consistencia se necesita profundizar en varios aspectos, los cuales están catalogados por niveles. Para ello se realiza un ordenamiento de mayor a menor consistencia [19]:

- Interpretación del comportamiento del usuario: Este establece que la Interfaz de Usuario debe llegar a comprender que significan los comandos abreviados (Shortcut-Keys) que han sido definidos por el usuario.
- Estructuras Invisibles: Estas requieren la realización de una definición clara de ellas, ya que de no hacerse, los usuarios podrían no llegar a comprender nunca su uso. Ejemplo de esto es la ampliación de las ventanas mediante la extensión de sus bordes.
- Pequeñas estructuras visibles: Estas estructuras definen el establecimiento de un conjunto de controles controlados por los usuarios, de manera que le permita ahorrar tiempo en la ejecución de tareas específicas. Ejemplos: Icono y/o botón para impresión.
- Una sola aplicación o servicio: La Interfaz de Usuario debe permitir la visualización de la aplicación o servicio utilizado como un componente único. Ejemplo de esto es, cuando la

Interfaz de Usuario despliega un único menú, pudiendo además acceder al mismo mediante comandos abreviados.

- Un conjunto de aplicaciones o servicios: Ésta constituye una abstracción superior de la anterior, es decir, establece que la Interfaz de Usuario permite la visualización de la aplicación o servicio como un conjunto de componentes. Ejemplo de esto es, cuando la Interfaz de Usuario presenta un conjunto de barras de comandos desplegadas en diferentes localizaciones dentro de la pantalla, siendo activadas de manera diferente.
- Consistencia del ambiente: Establece una concordancia entre la Interfaz de Usuario y el ambiente de trabajo. Lo cual se pone de manifiesto cuando la Interfaz de Usuario utiliza componentes de control como menús, botones de comandos; de manera análoga a otras Interfaces de Usuario que se estén utilizando en el ambiente de trabajo.
- Consistencia de la Plataforma: Es básicamente el establecimiento de una concordancia entre la Plataforma utilizada y la Interfaz de Usuario. Ejemplo de este es cuando la Interfaz de Usuario mantiene un esquema basado en ventanas, el cual se mantiene acorde al manejo del Sistema Operativo Windows.

La aparición de inconsistencias en el comportamiento de los componentes que integran la Interfaz de Usuario, debe de ser fácil de visualizar. La uniformidad entre los componentes que integran la Interfaz de Usuario debe ser evitada, además de que los objetos deben ser consistentes con su comportamiento. Por lo que si dos objetos actúan de diferentes maneras, estos deben lucir diferentes, siendo el testeo la única manera de verificar si la Interfaz de Usuario cumple con las expectativas del usuario.

Eficiencia del Usuario:

Plantea que debe de ser considerada el nivel de productividad del usuario antes que la productividad del ordenador. Si el usuario queda en espera a alguna respuesta del sistema por largos períodos de tiempo, éste tiempo perdido puede llegar a convertirse en pérdidas económicas para la organización, es por ello que los mensajes de ayuda deben ser claros y sencillos, y proveer respuesta al problema en cuestión, los menús y etiquetas de botones deben tener las palabras claves del proceso. Un ejemplo claro es cuando tenemos una Interfaz de usuario, en la cual están contenidos los botones Aceptar y Aplicar, los cuales aparentan realizar la misma función, una solución a esta incorrecta definición podría ser la supresión de uno de estos si realizaran la misma función, u etiquetándolos de manera que reflejen claramente los procesos específicos que ejecutan [19].

Ley de Fitt:

Constituye la ley más básica y más conocida de entre todas las leyes o principios del diseño de Interfaces de Usuarios. Plantea que, el tiempo para alcanzar un objetivo es una función de la distancia y tamaño del objeto, lo cual no es más que cuanto más grande y más cercano al puntero del ratón esté un objeto, más fácil será realizar clic sobre él. Es por ello que se recomienda la utilización de objetos grandes para la ejecución de las funciones más importante.

Tomando como ejemplo la barra de botones de Firefox, cuando los usuarios navegan por internet, el botón que más se utiliza es el botón Anterior, es por ello que este debe ser el botón más fácil de pulsar, de esta forma se minimiza el esfuerzo requerido por parte del usuario a la hora de utilizar la aplicación, siendo así que este se concentra profundamente en la navegación por la web.

El tamaño de un elemento dentro de las Interfaces Gráficas de Usuarios, puede parecer más grande si lo colocamos en los bordes de la pantalla, de esta manera cuando el cursor del ratón llega al borde de la pantalla, se sitúa exactamente en el borde, independientemente de la velocidad con que se estuviese moviendo el ratón. Para el usuario esto no significa más que los objetos que están en la frontera de la pantalla que tienen tamaño infinito. Es por ello tan fácil el hacer clic sobre algún control que se encuentre en la esquina superior derecha de la pantalla, simplemente sería desplazar el puntero hacia arriba y a la derecha sin limitaciones, no siendo así si fuésemos a pulsar sobre algún control ubicado en el centro de la pantalla [19].

Para resumir, podemos concluir que:

- Los controles más utilizados en la aplicación, deben de ser los más grandes y de mayor distinción.
- Se deben utilizar los bordes y esquinas de la pantalla para lograr que los controles sean virtualmente infinitos.
- Nunca colocar los controles a un pixel de distancia del borde de la pantalla o de una esquina.

Interfaces Explorables:

Plantea la facilidad que debe brindársele al usuario de salir ágilmente de la Interfaz de Usuario, dejando almacenado de alguna manera el estado de avance del trabajo que está realizando, dando así la posibilidad de continuidad posteriormente.

La aplicación debe permitir a aquellos usuarios inexpertos o principiantes en el uso de ésta, salir ágilmente, por lo que debe proveerse a estos usuarios de guías que permitan realizar estas tareas no usuales.

Es recomendable que el usuario pueda incorporar elementos visuales que le permitan, tanto un desplazamiento inmediato a ciertos puntos del trabajo en cuestión, como un punto de partida, que le permita tener un sentido de dirección a seguir en su trabajo. La Interfaz de Usuario debe permitir realizar la inversa de cualquier acción que se pueda realizar y que pueda implicar riesgo en el trabajo que se esté realizando, dándole así la posibilidad al usuario de explorar el sistema sin temor a cometer errores.

Es recomendable la incorporación de un comando “Deshacer”, ya que con este se suprimiría la necesidad de contar con diálogos de confirmación para cada acción que se realice en el sistema.

El usuario debe sentir seguridad a la hora de salir de la aplicación o del sistema cuando así lo desee, es por ello que se requiere un objeto fácil de accionar y que brinde la posibilidad de finalizar ésta [19].

Objetos de Interfaz Humana:

Estos objetos no necesariamente están inmersos en sistemas orientados a objetos, pueden ser vistos, escuchados, tocados o percibidos de alguna manera. Además de que deben ser entendibles, consistentes y estables.

Entre las representaciones más utilizadas de estos objetos encontramos las barras de controles, las cuales simplifican la operación con el sistema. Estas barras dadas las características ilustrativas que presentan los botones que la componen son de fácil aprendizaje y comprensión [19].

Uso de Metáforas:

Las buenas metáforas crean figuras mentales que resultan fáciles de recordar. De esta manera la Interfaz de Usuario puede contener objetos asociados al modelo conceptual de manera visual, con sonido u otras características que sean perceptibles por el usuario, ayudando así en la simplificación del uso del sistema.

Un ejemplo de una incorrecta utilización de el uso de metáforas, es el encontrar la metáfora de una cámara de video para representar el procesamiento de un documento por una impresora, en este caso, el botón [<<] carece de sentido, ya que una vez impreso no puede volver a un estado anterior [19].

Curvas de Aprendizaje:

El aprendizaje por parte de los usuarios de un producto y su usabilidad no son mutuamente excluyentes. La idea principal es que la Curva de Aprendizaje sea nula, y que el usuario principalmente pueda alcanzar el dominio total de la aplicación sin realizar esfuerzo [19].

Reducción de Latencia:

Recomienda el uso de tramas (multi-threading), ya que permite colocar la latencia en segundo plano (background). Las técnicas de trabajo multitareas posibilitan realizar trabajo ininterrumpido al usuario, realizando así las tareas de transmisión y computación de datos en segundo plano [19].

Protección del Trabajo:

Principalmente está enfocada a la seguridad que debe brindar el sistema o aplicación para lograr que el usuario nunca pierda su trabajo, ya sea por la ocurrencia de errores propios, problemas de transmisión de datos, pérdida de energía o cualquier otra razón inevitable [19].

Auditoría del Sistema:

La mayoría de los navegadores de internet (browsers), no mantienen información acerca de la situación del usuario en el entorno, pero para cualquier aplicación es conveniente conocer un conjunto de características tales como: hora de acceso al sistema, ubicación del usuario en el sistema y lugares que ha accedido entre otros. Además el sistema debe brindar la flexibilidad de poder dejar salir al usuario del sistema y este al volver a ingresar continuar trabajando desde el estado en que había dejado el trabajo [19].

Legibilidad:

La manera en que debe mostrarse la información en las Interfaces de Usuario debe ser fácil de ubicar y leer, permitiendo así que la Interfaz de Usuario favorezca la usabilidad del sistema de software. Para lograr este resultado, deben tenerse en cuenta algunos aspectos como son: el texto que aparezca en la Interfaz de Usuario debe tener un alto contraste, la utilización de colores como el texto en negro sobre fondo blanco o amarillo claro, el tamaño de las fuentes debe ser lo suficientemente grande como para poder ser leído en monitores estándares, es importante hacer clara la presentación visual (coloración/agrupación) de objetos, evitar la presentación de excesiva información [19].

Interfaces Visibles:

El creciente uso de internet, ha favorecido la implementación de Interfaces Visuales. Lo cual significa que el usuario siempre va a visualizar una página en específico, pero nunca va a tener conocimiento de la totalidad del espacio de páginas de internet. Es por ello que la navegación en las aplicaciones debe ser reducida a la mínima expresión. El usuario debe sentir la sensación de que está en un solo lugar y que el que varía es su trabajo. Esto no solamente elimina la necesidad de mantener mapas u otros sistemas de ayuda para la navegación, sino que brindan además una sensación de autonomía al usuario [19].

1.1.5 Pasos para construir una interfaz grafica de usuario

Pasos Clásicos:

Durante el proceso de diseño de una Interfaz de Usuario se pueden distinguir cuatro fases o pasos fundamentales [1].

- Reunir y analizar la información del usuario.
- Diseñar la Interfaz de Usuario.
- Construir la Interfaz de Usuario.
- Validar la Interfaz de Usuario.

Reunir y analizar la información del usuario:

Está referida a analizar y concretar qué tipos de usuarios va a hacer uso de la aplicación, qué tipos de tareas van a realizar, la manera en que se van a realizar estas tareas, qué tipos de exigencias presentan los usuarios del programa o aplicación, así como el entorno en que comúnmente se desenvuelven los usuarios, que a su vez estos pueden ser físico, social o cultural [1].

Diseñar la Interfaz de Usuario:

En esta fase de diseño de una Interfaz de Usuario, es donde se definen los objetivos de usabilidad de la aplicación, las tareas del usuario, los objetos y acciones de la interfaz, los iconos, vistas y representaciones visuales de los objetos, los menús de los objetos y ventanas. Es por ello que se requiere dedicar tiempo y recursos a esta fase antes de entrar en la codificación. Todos estos elementos visuales pueden ser hechos primero a mano y luego refinados con las herramientas pertinentes [1].

Construir la Interfaz de Usuario:

En esta fase es interesante la realización de un prototipo previo, es decir una primera versión de la aplicación que se realice rápidamente, y que de la posibilidad de visualizar el producto, dando así la posibilidad de realizarle las pruebas pertinentes, para así no arrastrar errores a la hora de codificarlo definitivamente [1].

Validar la Interfaz de Usuario:

Durante esta fase se recomienda la realización de pruebas de usabilidad del producto, con mayor énfasis si fuese con los usuarios finales del mismo. Es de suma importancia que la realización del diseño sea a partir de las exigencias del usuario y no del sistema [3].

Existen 11 pasos en el proceso de diseño, los cuales están centrados en las tareas, estos son [1]:

- Entender quién usará el sistema para hacer qué.
- Elegir tareas representativas para el diseño.
- Plagiar o copiar.
- Bosquejar un diseño.
- Pensar acerca del diseño.
- Crear un prototipo.
- Evaluar con los usuarios.
- Repetir.
- Construirla.
- Rastrearla.
- Cambiarla.

Técnicas y pasos avanzados para el diseño de interfaces de usuario

Presentación de la Información:

No debe haber exceso de objetos en la pantalla, y los que existan deben encontrarse bien distribuidos. Cada uno de los elementos visuales que se encuentran en la pantalla tiene un factor de influencia sobre los usuarios, así como la combinación de ellos con el resto de los elementos presentes en la pantalla [1].

Elementos de diseño de pantalla y su percepción visual:

Análisis de Color: Este constituye probablemente el elemento de una interfaz de usuario que más erróneamente se utiliza. El color es comunicador de información, no es solamente un elemento decorativo (Ejemplo: reforzar los mensajes de error). Deben utilizarse combinaciones de colores adecuadas (Ejemplo las paletas proporcionadas por los Sistemas Operativos), el color debe atraer la atención del usuario, pero no tornarse agotador después de un período de tiempo. Es de suma importancia seguir las líneas de diseño existentes. Principio básico: diseñar primero en blanco y negro, y luego añadir el color [1].

Análisis de Audio: Primeramente es necesario definir cuando es más apropiado la utilización de audio que la información visual. Seguido a esto se debe determinar el sonido adecuado, y por último se debe permitir la personalización del sonido, variación de volumen así como su desactivación.

El uso de sonidos en las aplicaciones puede causar molestias dependiendo del lugar en que se utilicen, en lugares abiertos puede resultar poco efectivo además de que puede resultar embarazoso para algunas personas. El sonido debe tratarse para brindar información, por ejemplo, un mensaje de aviso de correo electrónico o de bienvenida, o al iniciar una sesión de trabajo [1].

Análisis y Animación: Está definida como el cambio en el tiempo de la apariencia visual de un elemento gráfico. Por ejemplo, el progreso de acciones requeridas por el usuario (copia de ficheros, instalación de aplicaciones), estado de progresos (iconos de impresoras), acciones posibles a realizar por el usuario (cambio de cursor al desplazar el ratón). La animación puede ser utilizada al igual que ocurre con el sonido para brindar información al usuario, ejemplo al subrayar iconos importantes, mostrar el estado de un objeto en específico o al explicar su comportamiento [1].

Diseño Internacional: Este es un término con el cual se debe tener mucho cuidado a la hora de utilizarlo, ya que es abarcador de mucho trabajo en este campo. Debe tenerse mucho cuidado con las diferencias culturales (gestos, terminología, tratamiento de imágenes, formatos de teléfonos o calendarios, entre otros) [1].

Análisis y Elección de Controles: Es de suma importancia definir bien los controles a utilizar, ya que muchas veces existe la duda de cuáles utilizar y cuáles no [1].

Diferentes controles para los mismos datos:

Guías de Expertos

Existen varias guías de diseño, las cuales han sido reveladas de expertos y comités de desarrollo de este tipo de trabajo, dichas guías complementan las reglas de oro anteriormente estudiadas, por citar algunas de ellas [1]:

- Demasiada simetría puede hacer las pantallas difíciles de leer.
- Si se ponen objetos sin alinear, hacerlo drásticamente.

- Asimetría = activo, simetría = sereno.
- Elementos de tamaño y color similares se perciben como pertenecientes a un grupo.
- Asumir errores en la entrada del usuario.
- Diseñar para el usuario, no para demostrar los propios conocimientos tecnológicos.
- Unos gráficos espectaculares no salvaran a una mala interfaz.

1.2 Bibliotecas de desarrollo de sistemas de realidad virtual

Desde el punto de vista de la Informática (computación), una biblioteca es un conjunto de procedimientos y funciones (subprogramas) agrupadas en un archivo con el fin de que puedan ser aprovechadas por otros programas. Existen dos tipos de bibliotecas, las bibliotecas estáticas o de enlace y las bibliotecas compartidas o de enlace dinámico. De estas, las primeras hacen enlaces entre ellas, o sea, arreglan las referencias a rutinas en el programa para que apunten a su localización en la biblioteca al momento de compilación, mientras que el segundo tipo de bibliotecas se enlazan en tiempo de ejecución.

La denominación de biblioteca compartida hace énfasis en que, comúnmente, los procesos que la enlazan comparten una única parte de la memoria donde se encuentran las instrucciones de los subprogramas.

Algunos ejemplos de bibliotecas son:

- En Windows las llamadas dll (Dynamic Link Library), las cuales entran dentro de la clasificación de las bibliotecas dinámicas, mientras que las estáticas son las bien llamadas lib.
- En los sistemas Unix y Linux, las bibliotecas dinámicas tienen extensión so, mientras que las estáticas toman extensión a.

En el mundo de la realidad virtual podemos encontrar muchas de estas bibliotecas vinculadas a bibliotecas de desarrollo de sistemas de realidad virtual, algunas más avanzadas que otras, algunas de las cuales son más fáciles de usar que otras. Dentro de estas bibliotecas vinculadas a la realidad virtual, se puede distinguir que la mayoría brindan al usuario ficheros tanto de enlace dinámico, como de enlace estático, ambas con el objetivo de hacer más eficiente el proceso de compilación o ejecución según sea el caso.

Algunas de estas bibliotecas de desarrollo de sistemas de realidad virtual son:

- DirectX.
- Ogre.
- G3D.
- A7

1.2.1 Características de las bibliotecas de desarrollo de sistemas de realidad virtual más utilizadas

Cada una de estas bibliotecas destinadas al desarrollo de sistemas de realidad virtual, se diferencian una de otras, ya sea por el tipo de licencia o la plataforma donde es utilizada. Estas a su vez presentan características especiales como son las diferentes áreas dentro de la realidad virtual sobre la que actúan, los diferentes formatos de fichero con información grafica que son capaces de leer, entre otras. A continuación expondremos un grupo de características esenciales de cada una de estas bibliotecas anteriormente mencionadas, con el propósito de establecer características importantes a configurar en un entorno virtual.

1.2.1.1 DirectX

Esta biblioteca está considerada como una colección de APIs creadas para facilitar tareas relacionadas con la programación de juegos en la plataforma Microsoft Windows. El kit de desarrollo de DirectX es distribuido gratuitamente por Microsoft. Originalmente estas eran distribuidas por los desarrolladores de juegos con sus paquetes, pero más tarde fueron incluidas en Windows.

Como se mencionó anteriormente está constituida por un conjunto de APIs, entre los que podemos encontrar [\[4\]](#):

- Direct3D: Utilizado en el procesado y programación de gráficos en tres dimensiones, la cual constituye una de sus características más utilizadas.
- Direct Graphics: Centrada en la representación de imágenes en dos dimensiones, así como para la representación de imágenes en tres dimensiones.

- Direct Input: Empleado en el procesamiento de datos obtenidos del teclado, el ratón, mando de juegos (joystick), y otros controles para juegos.
- Direct Play: Utilizado en la comunicación por la red.
- Direct Sound: Utilizado en la reproducción y grabación de sonidos de ondas.
- Direct Music: Destinado a la reproducción de pistas musicales compuestas con Direct Music Producer.
- Direct Show: Encargado de la reproducción de audio y video con transparencia de red.
- Direct Setup: Este constituye el API fundamental ya que es el que contiene la instalación de componentes DirectX.

A pesar de ser un producto desarrollado exclusivamente para la plataforma Windows, existe actualmente una implementación de su API para sistemas Unix, en particular para Linux y X Window System, más conocido como Cedega, desarrollada por la empresa de software Transgaming y orientada a la ejecución de juegos desarrollados para Windows bajo sistemas Unix.

1.2.1.2 Ogre

Acrónimo del inglés Object-Oriented-Graphics-Rendering-Engine, es un motor de renderizado en tres dimensiones orientado a escenas y escrito en el lenguaje C++ y licenciado bajo la licencia LGPL (Lesser General Public License), además de contar con una comunidad muy activa. Sus librerías evitan la dificultad de la utilización de capas inferiores de librerías Gráficas como OpenGL y DirectX, además provee una interfaz basada en objetos del mundo y otras clases de alto nivel [3].

Ogre no solo fue desarrollado con el fin de la realización de juegos, sino también para simulaciones, aplicaciones de negocios o cualquier otro uso, además, para el caso de los juegos fue diseñado para explotar al máximo las posibilidades materiales de las tarjetas 3D, todo esto a través de una interfaz orientada a objetos; ofrece un sistema de partículas de gestión de recursos muy potente, además de una serie de funcionalidades muy interesantes.

Funcionalidades:

- Productividad [\[5\]](#):
 - Fácil de usar con interfaz orientada a objetos diseñada para minimizar el esfuerzo requerido al renderizar escenas 3D y separar lo de la implementación 3D como OpenGL o DirectX.
- Plataformas y soporte de APIs 3D
 - Posee soporte para DirectX y OpenGL.
 - Soporte para Windows en todas sus versiones, GNU/Linux y Mac.
 - Construcción en Visual C++ y Code::Blocks en Windows y g++ en GNU/Linux.
- Materiales y soporte de Shaders
- Mayas
- Animaciones
- Funcionalidades de la escena
- Efectos especiales

1.2.1.3 G3d

G3D es un motor gráfico 3D de grado comercial y disponible como código abierto bajo licencia BSD. Es utilizado en el desarrollo de juegos comerciales, roles investigativos, en realización de simuladores militares, y cursos de universidades. Proporciona un juego de rutinas y estructuras comunes, las cuales son necesarias en la mayoría de los programas gráficos. Hace uso de interfaces orientada a objetos, diseñada con el fin de minimizar el esfuerzo requerido al renderizar escenas 3D, y separarlo de la implementación como OpenGL. Constituye una base sólida muy optimizada para la construcción de aplicaciones 3D, provee un conjunto de aplicaciones de ejemplo, realizadas con el objetivo de mostrarle a los usuarios las diferentes formas en que pueden estructurar sus proyectos 3D, así como mostrar los usos más comunes de las diferentes bondades que brinda la biblioteca [\[6\]](#).

Esta biblioteca presenta soporte para la carga y texturizado de objetos con imágenes png, tga, bmp, jpg, pcx, ppm, pgm, pbm, dds, así como formatos ico (ícono). Presenta interfaces para la carga de

ficheros de mayas como el ply2, ifs, md2 y md3, además de otros ficheros como el 3ds y obj; cuenta con un formato de fuentes propio, para así evitar problemas de ilegalidades, actualmente cuenta con más de 20 tipos diferentes de fuentes y presenta interfaces que le ayudan al desarrollo de nuevas fuentes, de acuerdo a las necesidades del desarrollador. No presenta soporte oficial para el manejo de audio, aunque si tiene contribuciones que proveen de código que ayudan en el uso de las librerías más populares de audio [6].

1.2.1.4 A7

Este constituye el núcleo del sistema GameStudio, herramienta destinada a la creación de Sistemas 3D en tiempo real. Genera las imágenes 3D y los comportamientos del mundo visualizado. Fue desarrollada como el sucesor de A6 en el 2007. Consta de un nuevo árbol del manejador de escena, el cual incluye características dinámicas, ya que cambia dependiendo de la distribución del juego, ya sea de interiores o exteriores. Permite además, la adición de ocho luces, la mayor cantidad permitida por los diferentes hardwares, también puede añadir sombras, tanto estáticas como dinámicas [13].

El motor de render gráfico incluye características tales como [13]:

- Múltiples cámaras, vistas de render y monitores.
- Soporte para DirectX 9, DirectPlay, DirectShow, DirectSound.
- Árbol binario adaptativo (ABT por sus siglas en inglés) para el control de la escena.
- Particionamiento Binario del Espacio (BSP por sus siglas en inglés) y Puntos Potencialmente Visuales (PVS por sus siglas en inglés).
- Soporte para interiores y exteriores basados en LOD, sistema de terrenos.
- Manejador para soporte de luces e ilimitados puntos de luces.
- Sombras dinámicas y estáticas.
- Áreas de niebla, portales de cámaras, reflexiones y espejos.
- Animaciones basadas en huesos.
- Materiales para objetos estáticos y dinámicos.

También incluye funcionalidades adicionales para el trabajo con redes y sonidos, así como objetos 2D. Brinda una interfaz para el desarrollo de plugins en C generando DLL, con el objetivo de añadir funcionalidades nuevas.

1.2.2 Bibliotecas de desarrollo de sistemas de realidad virtual en Cuba

Cuba no ha quedado exenta del desarrollo de bibliotecas con el fin de desarrollar sistemas de realidad virtual, aunque aún se encuentra en un nivel muy bajo en el desarrollo de estas, hasta que en el año 1994, surge el Centro de Investigación y Desarrollo #2 (CID 2), más conocido en el mercado como SIMPRO (Simuladores Profesionales). Esta línea de investigación se fortalece a partir de la creación de la Universidad de la Ciencias Informáticas (UCI) en el año 2002. La facultad #5 de la UCI desarrolla como línea fundamental de investigación la Realidad Virtual, específicamente el proyecto, "Herramienta para el Desarrollo de Realidad Virtual". Este proyecto cuenta con una biblioteca que ha sido desarrollada durante el transcurso de los años y como base de las investigaciones que allí se desarrollan, llevando como nombre SceneToolKit (STK), y se han desarrollado con ella productos tales como:

- Simulador de conducción 2.0, expuesto en los laboratorios de Realidad Virtual de la facultad.
- Simulador de conducción de camión, el cual se encuentra en uso en la División de Tanques de Managua.
- Simulador de tiro antiaéreo para la pieza C10.
- Simulador de conducción de camión, expuesto en la Feria Informática 2007.

Consta con un grupo de características principales tales como:

- Está desarrollada en C++;
- Utiliza las bibliotecas DirectX y OpenGL.
- Está desarrollada en base al Sistema Operativo GNU/Linux, aunque mantiene su carácter multiplataforma.
- Presenta una arquitectura en tres capas.
 - Capa Interfaz o Aplicación: Presenta las funcionalidades de interfaces visuales de usuario.

- Capa Media o Engine: Es la encargada de controlar la funcionalidad central de la biblioteca; matemática, geometrías, grafo de escena, luces, cámaras, control de animaciones, etc.
- Capa Inferior, Base o Render: Encargada del acceso a las bibliotecas base, ya sea DirectX u OpenGL.
- Consta con el uso de ficheros propios para el almacenamiento de información tridimensional.
- La organización de los elementos en el grafo de escena permite que las actualizaciones en orden descendente en el árbol, actualicen y transmitan datos geométricos, así como la determinación de los elementos visibles en función del boundary, el recorrido ascendente del árbol proporciona un mecanismo de actualización de las fronteras o boundary.
- Su estructura de nodos está conformada por una interfaz principal, la cual controla la posición, orientación y controladores de tiempo (datos generales para cada nodo). Las clases que la implementan y la hacen un tipo de nodo más específico:
- Contiene animaciones que permiten añadir realismo a los Entornos Virtuales, logrando un efecto visual agradable. Algunas de estas animaciones son:
 - Animaciones por huesos.
 - Definición de caminos.
 - Árboles y ruedas inteligentes.
 - Seguimiento de terreno con tres puntos.
 - Detección de colisiones.

Al analizar las características de esta biblioteca podemos definir las principales particularidades que presenta, así como los parámetros modificables que brindan para lograr la configuración deseada por el usuario programador.

- Organización del grafo de escena.
- Relaciones entre las entidades del grafo de escena.
- Efectos especiales, luces, sombras, niebla, entre otros.

- Relación entre las animaciones y las entidades del grafo de escena.
- Colisiones entre las entidades del grafo de escena.

Las características anteriormente expuestas, permiten añadir y configurar efectos que actúan sobre los objetos tridimensionales, que a su vez conforman el Entorno Virtual y perfeccionan el realismo que estos imprimen en la crítica del espectador.

1.3 Bibliotecas de desarrollo de sistemas de realidad virtual que poseen interfaces graficas de usuario

Existen varias bibliotecas de desarrollo de sistemas de realidad virtual, las cuales presentan interfaces de usuarios (GUIs), lo cual le atribuye a los desarrolladores de entornos virtuales ciertas bondades a la hora de configurar estos entornos virtuales, ya que esta tarea incluye un alto grado de complejidad en cuanto a posicionar los elementos en el entorno, ajustar los parámetros de luces y sombras, la asignación de determinados comportamientos a dichos elementos, así como agruparlos según sean las necesidades de los desarrolladores, la modificación de alguna propiedad determinada a todo los elementos, así como algunas características específicas del entorno como es la niebla, zbuffer y el modo de representación, ya sea malla o solido. Estas Interfaces Gráficas de Usuario le permiten a los desarrolladores realizar su trabajo de un modo más eficiente, disminuyendo el tiempo empleado en la creación y configuración de los entornos virtuales.

Ejemplos de algunas de estas herramientas:

- Ogre Studio.
- Game Studio.

1.3.1 Principales características que presentan estas herramientas

Ogre Studio:

Es una herramienta para el uso con la biblioteca gráfica Ogre. Una herramienta modular basada en plugins que permite la edición de escenas gráficas y código, con comprobación de sintaxis y soporte para debugger, entre otras funcionalidades. Hace uso de otras bibliotecas tales como [8]:

- C# .Net 2.0.
- OgreDoNet.

- Ogre (1.2).
- MagicLibrary

Soporta funcionalidades como [8]:

- Interfaces para la construcción de nuevos plugins, sólo la interfaz visual principal para el usuario pertenece al diseño, las demás funcionalidades son añadidas como plugins.
- Una ventana para la visualización de la escena, muestra el grafo de escena (Nodos, Entidades y Luces).
- En la pantalla principal se muestra una ventana de Ogre, donde se actualiza en tiempo real la escena creada.
- Parser de sintaxis para formatos como .cpp, .xml, entre otros. Adjunta funcionalidades de punto de ruptura en depuración mediante DotNetFireball.

GameStudio:

No está definido como un motor de juegos o un editor de juegos, ni como un lenguaje de programación de juegos, sino como una mezcla de todos estos. El sistema de desarrollo contiene todas las herramientas necesarias para la creación de juegos comerciales ya sean en dos o tres dimensiones, así como la realización de cualquier tipo de aplicaciones gráficas de tiempo real. Este incluye un previo conocimiento del estado del arte de los motores gráficos para juegos en segunda o tercera dimensión, incluye también un conjunto de editores para agilizar el desarrollo de estos sistemas de realidad virtual; editor de niveles del juego, editor de modelos y terrenos, incluye el lite-C como lenguaje de scripting, así como un compilador, debugger y editor. Incluye además muchas otras facilidades de uso, como son librerías de modelos y texturas [9].

Game Studio es una herramienta orientada al usuario que brinda características profesionales, pensada y diseñada para proporcionar a cualquier usuario, ya este tenga conocimientos avanzados de programación o no, la posibilidad de crear juegos totalmente profesionales. Incluye un motor gráfico tridimensional (3D), con un editor de actores y mapas 3D, donde el juego tendrá lugar. También incluye librerías con objetos prefabricados, así como con juegos de ejemplo. Permite la creación de juegos en primera persona, en tercera persona, aventuras, juegos de rol, juegos de plataforma, simuladores y, en resumen, todo lo que pase en un momento por nuestro pensamiento.

Esta herramienta permite las facilidades de importar modelos de objetos ya definidos anteriormente en editores profesionales de objetos tridimensionales como, Worldcraft, Milkshape, Maya, 3D Studio Max. Así como también permite el soporte de los formatos de archivos más comunes, tales como, FBX, 3DS, ASE, OBJ, MDL, MD2, MAP. Así como permite la importación de archivos provenientes de estos editores de objetos anteriormente mencionados, admite importar terrenos de herramientas con estos fines, como el Terragen.

Tiene un generador de plantillas, que en dependencia del tipo de juego que se desee desarrollar, si es un juego de carreras, un juego de acción, entre otros. Esto nos da la posibilidad de crear un juego sin ningún tipo de programación, dándonos la posibilidad de con un simple clic del mouse, poder posicionar objetos en su juego, así como asignarle comportamientos, siendo todo esto configurable, sin programar una sola línea de código.

Existen tres versiones de Game Studio [9]:

- La Edición Extra, principalmente desarrollada para el desarrollo de juegos, contiene todas las funcionalidades para crear juegos 2d y 3D.
- La Edición Comercial, que como su nombre lo indica, está desarrollada para proyectos comerciales, brinda soporte de efectos y shaders para las nuevas versiones de hardware, así como facilidades para las redes de internet.
- La Edición Profesional, es para el desarrollo de grandes proyectos comerciales, tiene básicamente las mismas características que la Edición Comercial, pero incluye un año de soporte, así como un render de mayor nivel.

1.4 Bibliotecas de desarrollo de interfaces graficas de usuario en sistemas GNU/Linux

1.4.1 Qt

Qt es una biblioteca multiplataforma destinada al desarrollo de interfaces gráficas de usuario. Fue desarrollada por la compañía noruega Trolltech. Esta biblioteca es mayormente utilizada en entornos de escritorio KDE, aunque también es perfectamente funcional en otros escritorios como Gnome. Hace uso del lenguaje C++ de forma nativa, además existen interfaces para C, Python (PyQt), Java (QtJambi), Perl (PerlQt), Gambas (Gb.Qt), Ruby (QtRuby), PHP (PHP-Qt), Mono (Qtoto), entre otros [10].

Historia e impacto en el software libre:

Inicialmente Qt aparece como una biblioteca desarrollada por Trolltech en el año 1992, siguiendo un desarrollo basado en el código abierto pero no código libre. Esta se utilizó activamente en el desarrollo del escritorio KDE, el cual se desarrolló entre 1996 y 1998, llegando a alcanzar un notable éxito y rápida expansión. Esto fomentó la utilización de Qt para el desarrollo de aplicaciones de escritorio, situación catalogada por el proyecto GNU como peligrosa para el desarrollo del software libre. Para contrarrestar esta situación, se plantean dos iniciativas ambiciosas [10]:

1. El equipo de GNU, en el año 1997, le dio inicio al desarrollo del entorno de escritorio Gnome, haciendo uso de otra de las librerías para el desarrollo de interfaces visuales más utilizados en Linux, siendo esta Gtk.
2. Se le da comienzo a la realización de una biblioteca compatible con Qt, pero totalmente libre llamada Hamnoy.

En noviembre del año 1998 se anuncia el cambio de licencia de Qt, pues esta, a pesar de todo no contaba con el beneplácito de la Fundación de Software Libre (Free Software Foundation). Siendo así el 4 de septiembre del año 2000, TrollTech comenzó a ofrecer esta biblioteca bajo la licencia GPL en su versión 2.1. El 18 de enero de 2008, TrollTech anunció que Qt ofrecería Qt bajo la licencia GPL v3. Actualmente cuenta con un sistema de triple licencia, GPL v2 y GPL v3 para el desarrollo de software de código abierto (open source) y software libre, y otra licencia de pago para el desarrollo de aplicaciones comerciales.

Actualmente Qt se encuentra en su versión 4, y además de las múltiples mejoras que se le han realizado, ahora las bibliotecas Qt son también liberadas bajo la licencia GPL para Sistemas Windows y Mac.

Plataformas para las que se ha desarrollado esta biblioteca [10]:

- X11: Para X Windows System bajo la licencia GPL.
- Mac: Para Mac OS X bajo la licencia GPL.
- Windows: Para sistemas Windows bajo la licencia GPL (las antiguas versiones, anteriores a la 4.X eran no libres para este sistema operativo).

- **PDA:** Para dispositivos empotrados, también con licencia GPL y generalmente distribuido junto con Qtopia, permite introducir y crear scripts en las aplicaciones creadas con Qt.

Actualmente también está disponible QSA (Qt Scripts for Applications), que, basándose en ECMAScript/JavaScript, permite introducir y crear scripts en las aplicaciones creadas con Qt.

Además existen cuatro ediciones de Qt, las cuales están disponibles dentro de cada una de las plataformas anteriores, llamadas:

- **Qt Console:** Edición para desarrolladores non-GUI, o que no poseen Interfaz Gráfica de Usuario.
- **Qt Desktop Light:** Edición que posee un nivel reducido de GUI, está orientado a redes y bases de datos.
- **Qt Desktop:** Edición completa.
- **Qt Open Source Edition:** Edición completa, aunque presenta algunas excepciones como es el caso del control ActiveQt (ActiveX) para Windows, destinada solo a desarrolladores de software libre.

Módulos de Clases de Qt.

- **Núcleo.**
- **GUI.**
- **Redes.**
- **OpenGL.**
- **Base de Datos.**
- **XML.**

1.4.2 Gtk

Gtk+ o The Gimp Toolkit (conjunto de rutinas para GIMP), es un conjunto importante de bibliotecas o rutinas para desarrollar Interfaces Gráficas de Usuario (GUI) principalmente para entornos gráficos GNOME, XFCE y ROX de sistemas Linux. GTK es software libre (licenciado bajo la licencia LJPL), multiplataforma y parte importante del proyecto GNU [11].

En sus inicios GTK fue concebido para desarrollar el programa de manejo de imágenes GIMP, sin embargo actualmente es muy usada por muchos otros programas en los sistemas GNU/Linux. Cabe hacer mención a la biblioteca Qt, la cual constituye una opción alternativa a GTK y es muy utilizada además en entornos KDE mayormente.

GTK fue diseñada para permitir la programación con lenguajes como C, C++, Java, Perl o Python. Actualmente, su última versión (GTK 2), la cual incluye una gran cantidad de mejoras importantes, lo que lo hace superior a sus versiones anteriores. Sin embargo no son compatibles su primera versión y la segunda.

GTK está basado en tres bibliotecas fundamentales del equipo de desarrollo de esta [11]:

- **Glib**
- **GTK**
- **GDK**
- **ATK**
- **Pango**

Existe GTKGLExt, la cual es una extensión de GTK 2.0 o superior para OpenGL, esta proporciona una serie de objetos GDK con soporte para renderizado OpenGL en GTK+ y en aplicaciones que permitan GTKWidget API add-ons. También desarrolla GtkGLArea, la cual provee la GtkWidget API, quien es la encargada de habilitar el render de OpenGL en Gtk+. PyGtkGLExt, constituye una alternativa para los desarrolladores de aplicaciones en Python. GtkGLExt esta licenciado bajo la licencia LGPL al igual que Gtk. Esta extensión de Gtk para el trabajo con OpenGL sobre aplicaciones desarrolladas con la biblioteca Gtk aun se encuentra bajo desarrollo por lo que no se encuentra integrada completamente a Gtk.

1.5 Uso del estándar XML como formato de almacenamiento de datos en los sistemas de realidad virtual

XML o Extensible Markup Language (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y una

adaptación del SGML (Standard Generalized Markup Language) y permite definir la gramática de lenguajes específicos (de esta misma manera HTML es a su vez un lenguaje definido por SGML) [12].

XML no fue concebido únicamente para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Es una tecnología sencilla que posee a su alrededor otras que hacen su complemento y a su vez la hacen mucho más grande, así como con unas posibilidades mucho mayores. En la actualidad tiene gran importancia, ya que permite la compatibilidad entre sistemas para compartir información de una manera segura, fiable y fácil.

Algunas de las ventajas que XML reporta son [12]:

- Es extensible, lo cual significa que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de la aplicación.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

Actualmente existe un grupo de tecnologías que hacen uso de este tipo de formato para el almacenamiento de información, específicamente en el ámbito de la Realidad Virtual. Actualmente podemos encontrar diferentes tipos de ficheros 3D, que hacen uso del formato XML para almacenar información del estado de la geometría, textura, etc. Esto a su vez ha permitido grandes avances en la tecnología web, ya que actualmente podemos encontrar disponibilidades de navegación por entornos en tres dimensiones. Es decir que actualmente el formato XML ha posibilitado la extensión de la realidad virtual hasta esta esfera.

CAPITULO 2: CARACTERÍSTICAS DEL SISTEMA

2.1 Descripción de la solución propuesta

Como propuesta de soluciones técnicas de este trabajo se propone, el desarrollo de una aplicación de interfaz gráfica de usuario para la configuración de entornos virtuales que serán utilizados en el desarrollo de sistemas de realidad virtual con base en la biblioteca gráfica SceneToolKit (STK).

El desarrollo de esta aplicación enmarca un conjunto de tareas que facilitan en gran medida la configuración de los entornos virtuales que serán utilizados. Es por ello que se proponen las siguientes:

- Brindar la posibilidad de visualizar y seleccionar objetos dentro de escenas 3D.
- Realizar grupos de objetos dentro de escenas 3D.
- Permite la configuración de los estados de render para la escena 3D.
- Genera tres tipos de ficheros:
 - Fichero de Configuración Global.
 - Fichero de Configuración de los Entornos Virtuales.
 - Fichero de Configuración de Prefijos.

2.1.1 XML como formato para el almacenamiento de datos

El almacenamiento de la información que genera aplicación se estructuró en tres ficheros como se mencionó anteriormente, uno para la configuración global, otro para la configuración de los entornos virtuales y por último un fichero para la configuración de prefijos.

Como formato para el almacenamiento de dicha información se escogió el estándar XML. Actualmente este formato es uno de los más utilizados en el mundo dada las grandes ventajas que presenta, es un lenguaje extensible y de fácil entendimiento por parte de terceros.

En el caso de esta aplicación, se necesita que la información este estructurada por categorías, siendo XML el lenguaje más adaptable a esta manera de organizar la información, ya que este estructura la información en contenedores o categorías, brindando así una amplia compatibilidad entre la manera de organizar la información de esta aplicación con la manera de guardar la información de XML.

2.1.1.1 Forma de salvado

El fichero será exportado en formato plano, puesto que anteriormente se analizaron los principales ficheros que hacen uso de XML como formato para almacenar la información, siendo estos salvados en esta forma, ejemplo de estos son el X3D y el 3DXML, que son algunos de los más utilizados en la actualidad en el mundo de la Realidad Virtual.

2.1.1.2 Estructura y nombre de los ficheros

La manera de salvar la información estará distribuida en tres ficheros diferentes, cada uno con funcionalidades específicas en la aplicación. Estos llevarán como formato en su extensión <stk> [abreviaturas del nombre del fichero]. Siendo stk el nombre de la herramienta para la cual se está elaborando, la herramienta lleva por nombre Scene Tool Kit, donde sus abreviaturas son stk.

Los tres ficheros llevarán como nombre, Configuración Global (Global Configuration, gc como siglas en inglés), ConfiguraciónEntornoVirtual (vec como siglas en inglés), ConfiguraciónPrefijos (Prefix Configuration, pc como siglas en inglés).

Quedando así los nombres y extensiones para los tres ficheros:

- ConfiguraciónGlobal.stkgc.
- ConfiguraciónEntornosVirtuales.stkvec.
- ConfiguraciónPrefijos.stkpc.

2.1.1.2.1 Fichero de configuración global

Este fichero es el más sencillo entre todos, dado que cuenta con un solo bloque de datos, en el cual almacena la información referente a la configuración global que va a tener el entorno virtual a la hora de ser cargado y visualizado por la biblioteca base de esta herramienta. Este va a ser el encargado de brindar toda la información de los estados de render a nivel global que tendrá el entorno virtual. La estructura de este fichero se describe en la siguiente figura.

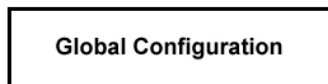


Figura 1: Estructura del fichero de configuración global.

2.1.1.2.2 Fichero de configuración de entorno virtual

La organización del fichero está dada en bloques de datos, los cuales se muestran en la siguiente figura.

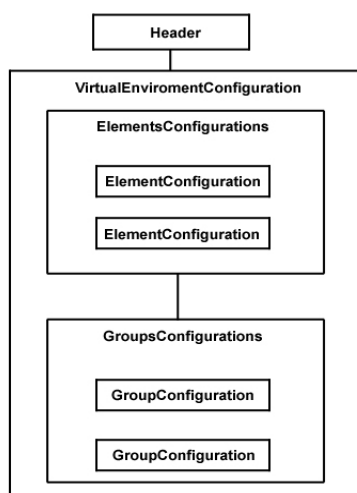


Figura 2: Estructura del fichero de configuración de entorno virtual.

El fichero estará organizado de manera que, primeramente un encabezado (Header), seguidamente un bloque de datos para la Configuración de Elementos (ElementsConfigurations), el cual es contenedor de los bloques de Configuración de Elemento (ElementConfiguration), seguido al bloque para la Configuración de Elementos un bloque de Configuración de Grupos (GroupsConfigurations) contenedor de los grupos de Configuración de Grupo de Elementos (GroupConfiguration).

2.1.1.2.3 Fichero de configuración de prefijos

Al igual que el fichero Configuración de Entornos Virtuales, este fichero estará constituido por bloques de información, los cuales se muestran en la siguiente imagen:

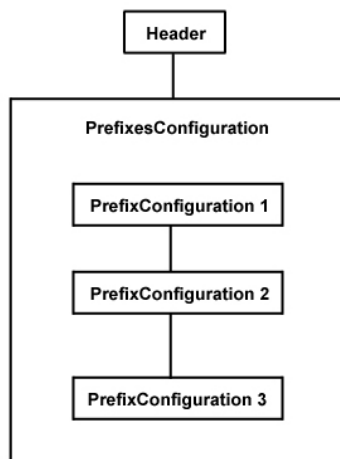


Figura 3: Estructura del fichero de configuración de prefijos.

Este fichero estará organizado de manera que, la primera sección perteneciente a un encabezado (Header), seguido de un bloque de Configuración de Prefijos (PrefixesConfiguration), el cual será contenedor de los bloques de Configuración de Prefijo (PrefixConfiguration).

2.1.1.3 Aspectos generales

Para exportar el formato se hará uso de las facilidades que brinda la Biblioteca de Desarrollo de Interfaces Visuales Qt, ya que cuenta con un conjunto de clases para el manejo del formato XML.

2.1.2 Método de selección de objetos en escenas 3d

Como algoritmo de selección de los objetos contenidos en las escenas 3D, está compuesto por dos partes:

- Primeramente la proyección de los puntos 3D de cada Boundingbox referente a cada elemento en el entorno virtual, y que a su vez cumpla con la condición que se encuentren en el frustrum de la cámara activa. Seguidamente se proyectan estos vértices 3D a un plano 2D en función de la Cámara activa que esta visualizando el entorno. Se toman estos puntos, y se construye, para cada elemento la envoltura convexa que constituye el Bounding 2D del Objeto.
- Se toma este polígono conformado por la envoltura convexa y se verifica el polígono trazado por el usuario como rectángulo de selección, y todos los elementos que intercepten su Boundingbox con el polígono de selección serán seleccionados. Sí el usuario solamente situó

un punto con un clic y no un polígono de selección, se comprueba que el punto marcado por el usuario esté dentro de los Boundingbox de los elementos, de ser así estos quedan seleccionados.

2.1.3 Herramientas de desarrollo y lenguaje utilizado

El uso de estas herramientas estará presente durante todos y cada una de las fases de desarrollo de esta herramienta u Interfaz Gráfica de Usuario, así como el lenguaje de programación que se referencia.

2.1.3.1 Code::Block

Code::Block constituye un IDE de programación en el lenguaje C++, desarrollado con el fin de encontrar la mayoría de las necesidades y exigencias de los usuarios, además está diseñado de manera muy extensible y configurable.

Finalmente constituye un IDE con todos los rasgos que requieran los usuarios, manteniendo una mirada consecuente, sensación y operación a través de plataformas.

Este es desarrollado alrededor de un framework de plugins, lo que lo hace estar dotado de una excelente extensibilidad. Cualquier tipo de funcionalidad que se desee puede serle añadida a este a través de plugins.

Actualmente este constituye el IDE principal de programación en el Proyecto Herramienta de Desarrollo de Sistemas de Realidad Virtual, que es para el cual se desarrolla esta Interfaz Gráfica de Usuario. Por lo que constituye un requisito la utilización del mismo para la programación de esta herramienta, además de que brinda una buena integración con la Biblioteca de Desarrollo de Interfaces Gráficas de Usuario Qt, que es la que se utilizara para el diseño de esta Interfaz Gráfica.

2.1.3.2 Qt Designer

Actualmente Qt constituye una amplia plataforma de desarrollo que incluye un amplio conjunto de clases, librerías y herramientas para la producción de Interfaces Gráficas de Usuario, y hace uso del lenguaje C++ como lenguaje de programación, a su vez es capaz de operar en varias plataformas. Con esta biblioteca se pueden llegar a desarrollar enriquecidas aplicaciones gráficas, incluye soporte para

nuevas tecnologías como OpenGL, XML, Bases de Datos, Implementación de Aplicaciones de Redes, internacionalización, entre otras.

Esta biblioteca está provista de una amplia gama de herramientas que facilitan la creación de formas, botones y ventanas de diálogo con el uso del ratón por citar algunos ejemplos. Las aplicaciones desarrolladas bajo Qt son muy elegantes, se ven y se operan mejor que las aplicaciones nativas.

Está provista de tres grandes ventajas ante otras librerías similares y que la rivalizan a su vez:

1. Qt es completamente gratuito para Aplicaciones de Código Abierto.
2. Es una biblioteca que está disponible para casi todas las plataformas, ya sea Sistemas Unix, Linux, MacOS, Solaris, así como para la familia Windows, por lo que una aplicación puede ser compilada y utilizada en cualquier plataforma sin necesidad de cambiar el código, siendo así, que esta se visualizará y actuará mejor que una aplicación nativa.
3. Qt hace uso de una extensa biblioteca de clases y herramientas para la creación de Aplicaciones Gráficas, estas bibliotecas de clases están bien documentadas, así como de fácil uso y tienen una gran herencia de programación orientada a objetos, lo cual hace de la programación de interfaces gráficas de usuario una aventura placentera.

2.1.3.3 Visual Paradigm

Visual Paradigm es una herramienta UML de tipo profesional, la cual soporta el ciclo completo de desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Este software de modelado UML constituye una gran ayuda en la rápida construcción de aplicaciones de calidad a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generador de código desde diagramas y generar documentación. Esta herramienta CASE (Computer Aided Software Engineering) como comúnmente se conoce esta provista de un gran número de tutoriales de modelado UML, demostraciones interactivas de UML, así como proyectos de este tipo.

Algunas de sus principales características son:

1. Diagrama de Procesos de Negocio – Proceso, Decisión, Actor de Negocio, Documento Modelado colaborativo con CVS y Subversion (nueva característica).
2. Ingeniería de Ida y Vuelta.

3. Ingeniería Inversa, Código a modelo, código a diagrama;
4. Generación de Código, Modelo a Código, Diagrama a Código.
5. Diagrama de flujo de datos.
6. Generación de Bases de Datos, Transformación de Diagramas Entidad-Relación en tablas de Bases de Datos.
7. Generador de informes para elaboración de documentación.
8. Distribución automática de diagramas, Reorganización de las figuras y conectores de los diagramas UML.
9. Importación y Exportación de ficheros XMI, así como archivos de proyectos Rational Rose MDL y CAT.
10. Carácter multiplataforma, ya que podemos hacer uso de esta herramienta tanto en sistemas Linux, Unix o Windows.

2.1.3.4 C++

Actualmente existen básicamente tres lenguajes de programación que son utilizados en la programación de aplicaciones gráficas profesionales 3D, Ensamblador, C y C++, ya que estos son los que más rápidamente logran la ejecución de código, es decir los que tienen menor costo de ejecución de programas. Más recientemente el lenguaje Java ha estado mostrando significativos avances en este sector. Actualmente en el Proyecto Herramienta de Desarrollo de Sistemas de Realidad Virtual, el lenguaje que se utiliza es C++, mostrando a la vez significativos resultados, además de que es el lenguaje en el que mayor experiencia se tiene por parte del equipo de desarrollo de este proyecto. Es por esto que se eligió este lenguaje para el desarrollo de la Interfaz Gráfica de Usuario que se desea obtener mediante el transcurso de este trabajo.

2.2 Diseño del sistema

2.2.1 Reglas del negocio

La aplicación de las configuraciones se realizará siguiendo un orden específico, debido a su nivel de importancia y jerarquía.

- Configuración específica a cada elemento.
- Configuración de grupos.
- Configuración de prefijos.

En el documento XML, la configuración de los valores que no se especifican se asumirá como no existente. Ejemplo: En el archivo de Configuración de Entorno Virtual, si el <alfastate> no se encuentra definido es que no está activado por tanto se asumirá así en la lectura.

2.2.2 Modelo de dominio

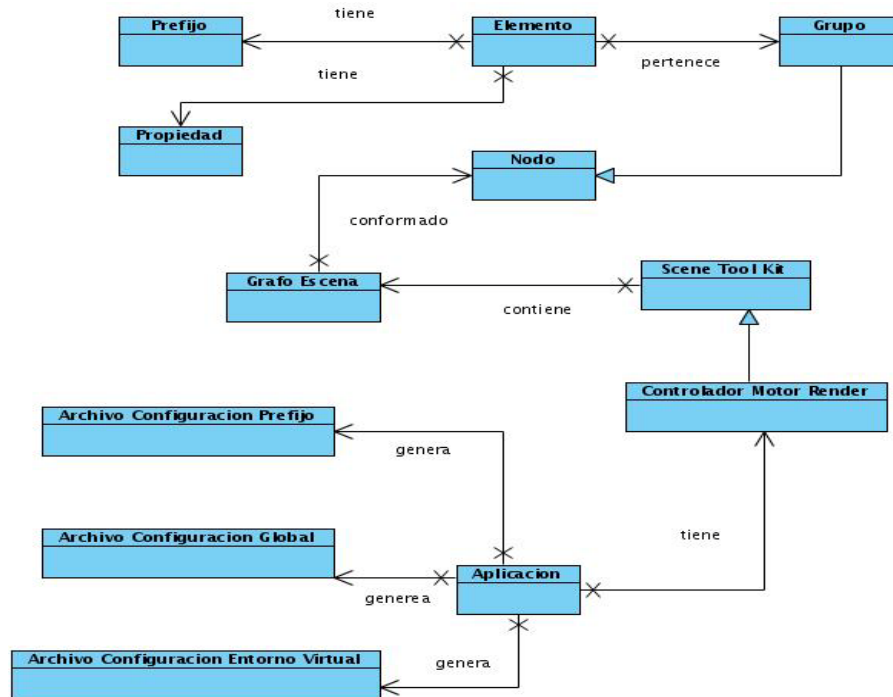


Figura 4: Modelo de dominio del sistema.

2.2.3 Captura de requisitos

- **Requisitos funcionales:**

1. Cargar ficheros 3DX.
2. Realizar selecciones de objetos en escenas 3D.
3. Gestionar configuración de los elementos seleccionados.
4. Gestionar la configuración global.
5. Gestionar prefijos.
6. Gestionar configuración de prefijos.
7. Realizar agrupaciones por elementos.
8. Desagrupar elementos.
9. Realizar vista previa.
10. Generar archivo de configuración de entorno virtual.
11. Generar archivo de configuración global.
12. Generar archivo de configuración de prefijos.
13. Cargar archivo de configuración de entorno virtual.
14. Cargar archivo de configuración de prefijos.
15. Cargar archivo de configuración global.
16. Filtrar los elementos seleccionados.

- **Requisitos no funcionales:**

- **Requerimientos de Software:**

La aplicación fue desarrollada en el Sistema Operativo GNU/Linux Debian. Se utilizó Code::Block como IDE de programación, así como la librería Qt 4 como librería de desarrollo de Interfaces Gráficas de Usuarios en Sistemas Operativos GNU/Linux.

- **Requerimientos de Hardware:**

Es necesaria una PC con CPU Intel Pentium IV, 1 Gb de HDD, 520 Mb de RAM.

- **Restricciones en el Diseño y la Implementación:**

- Estándar de Codificación:
- Lenguaje de programación: C++.
- Herramientas de Desarrollo: Code::Block.
- Biblioteca de Clases: SceneToolKit.

- **Requerimientos de Usabilidad:**

Los futuros usuarios de la aplicación, no necesariamente tienen que ser programadores. Solo se necesita tener conocimiento acerca de los diferentes estados de render de OpenGL.

- **Requerimientos de Soporte:**

La aplicación debe de ser compatible con el Sistema Operativo GNU/Linux Debian.

2.2.4 Modelos de casos de uso

2.2.4.1 Definición de los actores del sistema

Actores	Justificación
---------	---------------

Usuario	Es el que se desempeña en la labor de realizar las diferentes configuraciones que se le realizarán al Entorno Virtual en función.
---------	---

Tabla 1: Definición del actor del sistema.

2.2.4.2 Casos de uso del sistema

Dada la extensión de los casos de uso presentaremos solo las descripciones de los casos de usos más significativos para la arquitectura y funcionalidad del software. Para un completo análisis de los casos de uso se podrán encontrar en el ANEXO 1.

CU-3	Cargar Archivo Configuración Entorno Virtual
Actor	Usuario
Descripción	Lee el archivo Configuración Entorno Virtual y aplica estas configuraciones al entorno virtual.
Referencia	RF13

Tabla 2: Caso de Uso Cargar Archivo Configuración Entorno Virtual.

CU-4	Generar Archivo Configuración Entorno Virtual
Actor	Usuario
Descripción	Crea un archivo con la información que será utilizada para configurar los entornos virtuales.
Referencia	RF10

Tabla 3: Caso de Uso Generar Archivo Configuración Entorno Virtual.

CU-5	Cargar Archivo Entorno Virtual
-------------	---------------------------------------

Actor	Usuario
Descripción	El usuario selecciona el entorno virtual que desea cargar y se muestra en STKEditor para posteriores modificaciones.
Referencia	RF1

Tabla 4: Caso de Uso Cargar Archivo Entorno Virtual.

CU-6	Cargar Archivo Configuración Global
Actor	Usuario
Descripción	Lee el archivo Configuración Global, y aplica estas configuraciones al entorno virtual.
Referencia	RF15

Tabla 5: Caso de Uso Cargar Archivo Configuración Global.

CU-7	Generar Archivo Configuración Global
Actor	Usuario
Descripción	Crea un archivo con la información que será utilizada para configurar parámetros globales de los entornos virtuales que serán utilizados por estos.
Referencia	RF11

Tabla 6: Caso de Uso Generar Archivo Configuración Global.

CU-8	Gestionar Grupo Elementos
Actor	Usuario

Descripción	Gestiona los grupos de elementos creándolos modificándolos o eliminándolos.
Referencia	RF7, RF8

Tabla 7: Caso de Uso Gestionar Grupo Elementos.

CU-11	Gestionar Configuración Global
Actor	Usuario
Descripción	Gestiona la configuración de los entornos virtuales, es global porque configura parámetros que afectan de forma global a todos los elementos del entorno.
Referencia	RF4

Tabla 8: Caso de Uso Gestionar Configuración Global.

CU-12	Gestionar Prefijos
Actor	Usuario
Descripción	Agrega un prefijo nuevo a los conocidos por el sistema.
Referencia	RF5

Tabla 9: Caso de Uso Gestionar Prefijos.

CU-13	Gestionar Configuración Especifica
Actor	Usuario
Descripción	Gestiona la configuración de un elemento en el entorno virtual.
Referencia	RF3

Tabla 10: Caso de Uso Gestionar Configuración Específica.

CU-21	Vista Previa
Actor	Usuario
Descripción	Crea una vista previa de lo que será el entorno virtual con las configuraciones actuales.
Referencia	RF9

Tabla 11: Caso de Uso Vista Previa.

CU-22	Cargar Archivo Configuración Prefijos
Actor	Usuario
Descripción	Lee el archivo Configuración Prefijos, y aplica estas configuraciones al entorno virtual.
Referencia	RF14

Tabla 12: Caso de Uso Cargar Archivo Configuración Prefijos.

CU-23	Generar Archivo Configuración Prefijos
Actor	Usuario
Descripción	Crea un archivo con la información que será utilizada para configurar prefijos que serán utilizados por estos.
Referencia	RF12

Tabla 13: Caso de Uso Generar Archivo Configuración Prefijos.

2.2.4.3 Diagrama de casos de uso

Dada la extensión en números de casos de uso, solo se tratarán los más significativos de los mismos, siendo estos los siguientes. Para realizar un análisis mas profundo los diagramas que no se presentan a continuación se encontraran en el ANEXO1

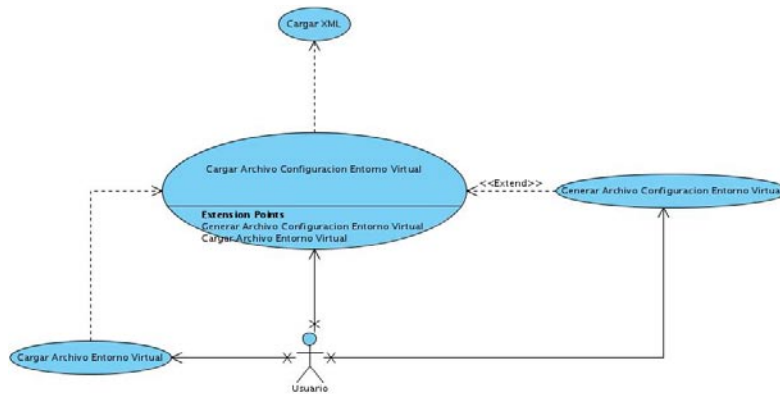


Figura 5: Diagrama de caso de uso, cargar archivo entorno virtual.

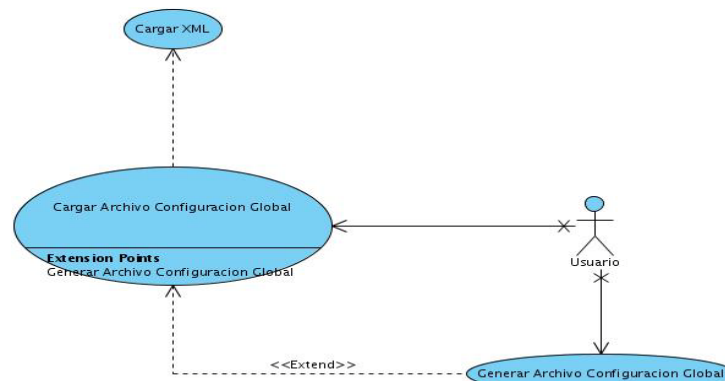


Figura 6: Diagrama de caso de uso, cargar archivo configuración global.

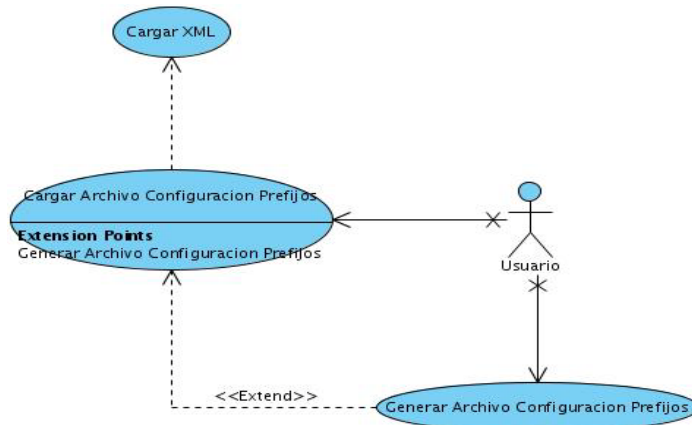


Figura 7: Diagrama de caso de uso, cargar archivo de configuración prefijos.

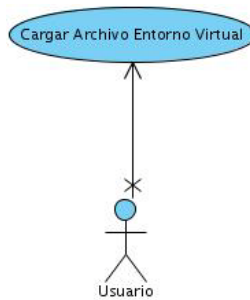


Figura 8: Diagrama de caso de uso, cargar archivo entorno virtual.

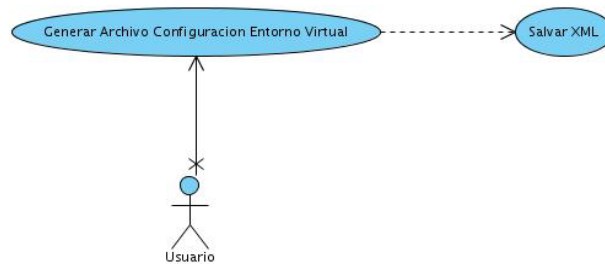


Figura 9: Diagrama de caso de uso, generar archivo configuración entorno virtual.

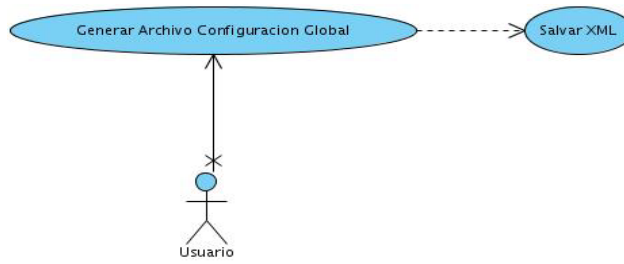


Figura 10: Diagrama de caso de uso, generar archivo configuración global.

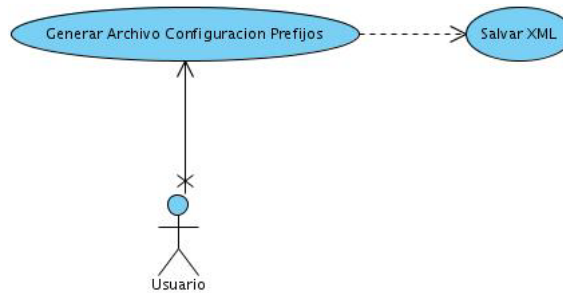


Figura 11: Diagrama de caso de uso, generar archivo configuración prefijos.

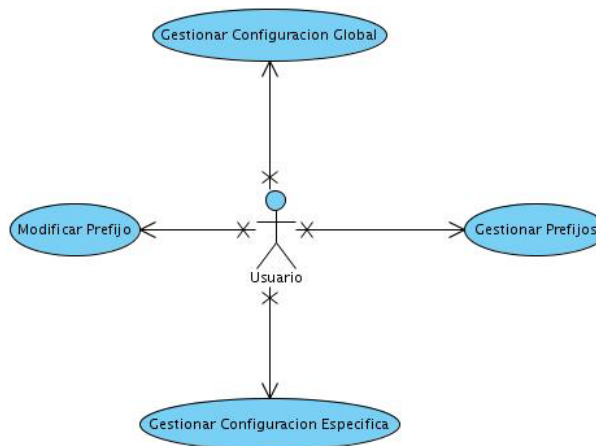


Figura 12: Diagrama de caso de uso, gestionar configuración.

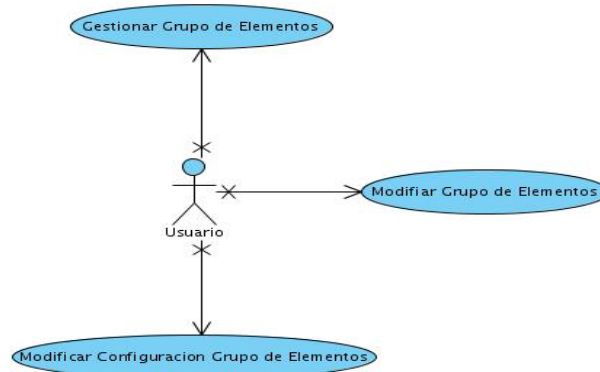


Figura 13: Diagrama de caso de uso, gestionar grupo de elementos.

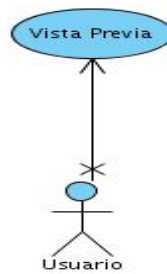


Figura 14: Diagrama de caso de uso, vista previa.

2.2.4.4 Descripción de los casos de uso en formato expandido

Para lograr un mayor entendimiento de las funcionalidades asociadas a cada caso de uso, no es suficiente con la representación gráfica de los diagramas de casos de uso, es por ello que se realiza la expansión de los mismos, donde se muestra más detalladamente sus funcionalidades, logrando así un mejor entendimiento de los mismos. Para realizar un análisis más profundo las descripciones que no se presentan a continuación se encontraran en el ANEXO1.

Caso de Uso:	Cargar Archivo Configuración Entorno Virtual.
Actores:	Usuario
Trabajadores:	-

Características de Sistema

Resumen:	Lee el archivo Configuración Entorno Virtual y aplica estas configuraciones al entorno virtual.	
Referencia:	R13	
Precondiciones:	Existe el Archivo y el entorno virtual que configura se encuentra cargado.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Selecciona la opción Cargar Configuración Entorno Virtual.	<p>1.1 Si existe configuración anterior ir a flujo alterno Configuración Anterior.</p> <p>1.2 Invoca caso de uso Cargar Archivo Entorno Virtual.</p> <p>1.3 Solicita archivo a cargar.</p>	
2. Especifica archivo a cargar.	<p>2.1 Crea una estructura donde guardar la información.</p> <p>2.2 Invoca al caso de uso Cargar XML.</p> <p>2.3 Solicita entorno virtual.</p>	
3. Facilita el entorno virtual.	<p>3.1 Limpia las estructuras donde se almacenarán los datos.</p> <p>3.2 Si existe entorno virtual cargado lo descarga.</p> <p>3.3 Carga el entorno virtual y crea todas las estructuras correspondientes a la configuración leída.</p>	
Flujos Alternos		

Acción del Actor		Respuesta del Negocio
1.1 Especifica una opción.		1.1.1 Pregunta si desea guardar la configuración anterior. 1.1.2 Si no desea guardar continua en el paso 3 del Flujo Normal de los Eventos. 1.1.3 Invoca caso de uso Generar Archivo Configuración Entorno Virtual. 1.1.4 Continúa en el paso 3 del Flujo Normal de los Eventos.
Pos condiciones	El entorno virtual queda configurado.	
Prioridad	Crítico.	

Tabla 14: Expansión del Caso de Uso Cargar Archivo Configuración Entorno Virtual.

Caso de Uso:	Generar Archivo de Configuración Entorno Virtual.	
Actores:	Usuario.	
Trabajadores:	-	
Resumen:	Crea un archivo con la información que será utilizada para configurar los entornos virtuales.	
Referencia:	R10	
Precondiciones:	Información disponible.	
Flujo Normal de Eventos		
Acción del Actor		Respuesta del Negocio
1. Seleccionar la opción generar archivo		1.1 Solicita nombre del archivo y dirección

Características de Sistema

Configuración Entorno Virtual.	donde se desea almacenar.
2. Especifica nombre del archivo y la dirección donde se desea almacenar.	2.1 Guarda la información a almacenar en el archivo en una estructura. 2.2 Invoca al caso de uso Salvar XML.
Pos condiciones	Archivo Configuración Entorno Virtual queda creado con la información deseada en él.
Prioridad	Crítico.

Tabla 15: Expansión del Caso de Uso Generar Archivo de Configuración Entorno Virtual.

Caso de Uso:	Cargar Archivo Entorno Virtual.	
Actores:	Usuario.	
Trabajadores:	-	
Resumen:	El usuario selecciona el entorno virtual que desea cargar y se muestra en STKEditor para posteriores modificaciones.	
Referencia:	R1	
Precondiciones:	Existe el archivo y STKEditor se encuentra disponible.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Selecciona la opción Cargar Entorno Virtual.	1.1 Solicita archivo a cargar.	
2. Selecciona archivo a cargar.	2.1 Se procede a cargar el archivo. 2.2 Se crea el árbol genealógico representando el grafo de escena.	

Características de Sistema

Pos condiciones	STKEditor muestra el entorno virtual seleccionado.
Prioridad	Crítico.

Tabla 16: Expansión del Caso de Uso Cargar Archivo Entorno Virtual.

Caso de Uso:	Cargar Archivo de Configuración Global.	
Actores:	Usuario.	
Trabajadores:	-	
Resumen:	Lee el archivo Configuración Global, y aplica estas configuraciones al entorno virtual.	
Referencia:	R15	
Precondiciones:	Existe el Archivo.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Selecciona la opción Cargar Configuración Global.	1.1 Si existe una configuración ir flujo alternativo. 1.2 Existe Configuración. 1.3 Solicita archivo a cargar.	
2. Especifica el archivo.	2.1 Crea una estructura donde guardar la información. 2.2 Invoca al caso de uso Cargar XML. 2.3 Limpia las estructuras donde se guardará la información. 2.4 Almacena los datos leídos en las	

Características de Sistema

	estructuras correspondientes.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
1.1 Selecciona su opción.	<p>1.1.1 Pregunta si se desea guardar la configuración anterior.</p> <p>1.1.2 Si seleccionó no guardar continua en el paso 3 del Flujo Normal de los Eventos.</p> <p>1.1.3 Invoca caso de uso Generar Archivo Configuración Global.</p> <p>1.1.4 Continúa en el paso 3 del Flujo Normal de los Eventos.</p>
Pos condiciones	Queda configurada la herramienta para futuros o actuales entornos virtuales.
Prioridad	Crítico.

Tabla 17: Expansión del Caso de Uso Cargar Archivo Configuración global.

Caso de Uso:	Generar Archivo de Configuración Global.
Actores:	Usuario.
Trabajadores:	-
Resumen:	Crea un archivo con la información que será utilizada para configurar parámetros globales de los entornos virtuales que serán utilizados por estos.
Referencia:	R11

Características de Sistema

Precondiciones:	Información disponible.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Seleccionar la opción generar archivo Configuración Global.	1.1 Solicita nombre del archivo y dirección donde se desea almacenar.	
2. Especifica nombre del archivo y la dirección donde se desea almacenar.	2.1 Almacena la información a guardar en el archivo en una estructura. 2.2 Invoca al caso de uso Salvar XML.	
Pos condiciones	Archivo Configuración Global queda creado con la información deseada en el.	
Prioridad	Crítico.	

Tabla 18: Expansión del Caso de Uso Generar Archivo de Configuración global.

Caso de Uso:	Gestionar Grupo de Elementos.	
Actores:	Usuario.	
Trabajadores:	-	
Resumen:	Gestiona los grupos de elementos creándolos modificándolos o eliminándolos.	
Referencia:	R7, R8	
Precondiciones:	Aplicación funcionando correctamente.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El usuario solicita una de las opciones del	1.1 Si selecciona la opción Crear Grupo ir a	

Características de Sistema

caso de uso.	sección Crear Grupo. 1.2 Si selecciona la opción Eliminar Grupo ir a sección Eliminar Grupo. 1.3 Si selecciona la opción Modificar Grupo ir a sección Modificar Grupo.
Flujo Normal de Eventos	
Sección "Crear Grupo de Elementos "	
Acciones del Actor	Respuestas del Negocio
1. Especifica Nombre Único del grupo.	1.1 Verifica si el nombre del grupo es único 1.2 Si no es único ir a flujo alternativo Nombre No Único. 1.3 Verifica si el elemento o alguno de los elementos seleccionados pertenece a algún grupo. 1.4 Si se encuentra algún elemento en otro grupo ir caso alternativo Elemento en Otro Grupo 1.5 Crea una estructura con el(los) id(s) de el(los) elemento(s) seleccionado(s) y asigna el Nombre Único especificado por el usuario.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
1.4.1 Se especifica si se continúa con la	1.4.1 Comunica que existe al menos un elemento de los seleccionados en otro grupo y se pregunta si desea continuar con la

Características de Sistema

operación de crear el grupo o no.	operación de crear un grupo nuevo. 1.4.2 Si no se desea continuar con la operación, ir a flujo alternativo Elementos en Ningún Grupo. 1.4.3 Se elimina el(los) elemento(s) en otro grupo de ese respectivo grupo. 1.4.4 Continúa en el paso 6 del Flujo Normal de los Eventos.
1.2 Nombre no único.	1.2.1 Informa que el nombre del grupo no es único.
Elementos en ningún grupo. 1.4.2 Especifica su elección.	1.4.2.1 Pregunta si desea crear el grupo con los elementos que no pertenecen a ningún grupo. 1.4.2.2 Busca los elementos que no pertenecen a ningún grupo. 1.4.2.3 Si no existen elementos que no pertenezcan a ningún grupo ir a flujo alternativo No Elementos. 1.4.2.4 Continúa en el paso 6 del Flujo Normal de los Eventos.
No elementos 1.4.3 Especifica su elección.	1.4.3.1 Informa que no existen elementos que no pertenezcan a ningún grupo y que de crearse el nuevo grupo se crearía vacío y pregunta si desea continuar creando el grupo. 1.4.3.2 Si desea crear el grupo continúa en el

Características de Sistema

	paso 6 del Flujo Normal de los Eventos.
Flujo Normal de Eventos	
Sección " Modificar Grupo de Elementos "	
Acciones del Actor	Respuestas del Negocio
1. Se escoge la opción Modificar Grupo de Elementos.	<p>1.1 Verifica si existen más de un grupo seleccionado.</p> <p>1.2 Si existe más de un grupo seleccionado ir a flujo alternativo Varios Grupos.</p> <p>1.3 Ir a flujo alternativo Modificar Nombre.</p> <p>1.4 Modifica el(los) nombre(s) de el(los) grupo(s) implicado(s).</p>
Flujo Alternativo de Eventos	
Acciones del Actor	Respuesta del Negocio
<p>Varios Grupos</p> <p>1.2 Especifica su elección.</p>	<p>1.2.1 Pregunta si desea cambiarle el nombre a todos.</p> <p>1.2.2 Si desea cambiarle el nombre a todos para cada grupo ir a flujo alternativo Modificar Nombre.</p>
<p>Modificar Nombre</p> <p>1.3 Especifica el nuevo nombre.</p>	<p>1.3.1 Solicita el nuevo nombre del grupo.</p> <p>1.3.2 Comprueba si es único.</p> <p>1.3.3 Si no es único ir a flujo alternativo Nombre No Único.</p> <p>1.3.4 Almacena el viejo y nuevo nombre del</p>

Características de Sistema

	grupo en una estructura.
Nombre no único. 1.4 Especifica su elección.	1.4.1 Informa que el nombre del grupo no es único y pregunta si desea intentar otro nombre. 1.4.2 Si desea probar otro nombre ir a flujo alterno Modificar Nombre.
Pos condiciones	Grupos disponibles.
Prioridad	Crítico.

Tabla 19: Expansión del Caso de Uso Gestionar Grupo de Elementos.

Caso de Uso:	Modificar Configuración Grupo de Elementos.	
Actores:	Usuario.	
Trabajadores:	-	
Referencia:	R3	
Resumen:	Modifica la configuración del grupo y con la de los elementos pertenecientes al grupo.	
Referencia:		
Precondiciones:	Grupo seleccionado.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Selecciona grupo de elementos.	1.1 Muestra la configuración actual del grupo.	
2. El usuario selecciona una opción de las del caso de uso.	2.1 Si selecciona la opción Cancelar ir a sección Cancelar.	

Características de Sistema

	<p>2.2 Si selecciona la opción Restablecer ir sección Restablecer.</p> <p>2.3 Selecciona la opción Aceptar</p> <p>2.4 Actualiza la configuración de todos los elementos pertenecientes al grupo.</p>
Sección "Cancelar"	
Acción del Actor	Respuesta del Negocio
	1. No aplica ningún cambio
Sección "Restablecer"	
Acción del Actor	Respuesta del Negocio
	1. Muestra la configuración por defecto del grupo.
Pos condiciones	Configuración del grupo actualizada.
Prioridad	Crítico.

Tabla 20: Expansión del Caso de Uso Modificar Configuración Grupo de Elementos.

Caso de Uso:	Gestionar Configuración Global.
Actores:	Usuario.
Trabajadores:	-
Resumen:	Gestiona la configuración de los entornos virtuales, es global porque configura parámetros que afectan de forma global a todos los elementos del entorno.
Referencia:	R4

Características de Sistema

Precondiciones:	Aplicación funcionando correctamente.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Selecciona opción Configuración Global.	1.1 Muestra la configuración global actual.	
2. Si selecciona opción Restablecer.	2.1 Ir a Sección Restablecer Configuración.	
3. Selecciona la opción Aceptar.	3.1 Almacena la información de la configuración.	
Sección "Restablecer Configuración"		
Acción del Actor	Respuesta del Negocio	
	1. Restablece la configuración.	
Pos condiciones	Configuración disponible.	
Prioridad	Crítico.	

Tabla 21: Expansión del Caso de Uso Gestionar Configuración Global.

Caso de Uso:	Gestionar Prefijos.
Actores:	Usuario.
Trabajadores:	-
Resumen:	Agrega un prefijo nuevo a los conocidos por el sistema.
Referencia:	R5
Precondiciones:	Aplicación funcionando correctamente.
Flujo Normal de Eventos	

Características de Sistema

Acción del Actor	Respuesta del Negocio
1. Selecciona la opción gestionar prefijos.	1.1 Muestra los prefijos actuales.
2. Si selecciona Agregar.	2.1 Ir a sección Agregar Prefijo.
3. Si selecciona Eliminar.	3.1 Ir a sección Eliminar Prefijo.
4. Selecciona un prefijo.	
5. Si selecciona Modificar	5.1 Ir a sección Modificar Prefijo.
Sección "Insertar Prefijo"	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción Agregar Prefijo.	1.1 Solicita el nombre del prefijo.
2. Especifica el nombre del prefijo.	2.1 Verifica si el nombre es único. 2.2 Si no es único ir a flujo alternativo Nombre no Único. 2.3 Agrega el prefijo a los conocidos por el sistema.
Sección "Insertar Prefijo"	
Acción del Actor	Respuesta del Negocio
2.2 Nombre no único.	2.2.1 Informa que el nombre no es único.
Sección "Modificar Prefijo"	
Acción del Actor	Respuesta del Negocio
	1.1 Solicita nuevo nombre para el prefijo.
2. Especifica el nuevo nombre.	2.1 Verifica que el nombre sea único.

Características de Sistema

	2.2 Si no es único ir a Flujo Alterno Nombre No Único.
	2.3 Cambia el nombre del prefijo.
Sección "Modificar Prefijo"	
Acción del Actor	Respuesta del Negocio
2.1 Nombre no único.	2.1.1 Informa que el nombre no es único.
Sección "Eliminar Prefijo"	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción Agregar Prefijo.	1.1 Solicita el nombre del prefijo.
2. Especifica el nombre del prefijo.	2.1 Verifica si el nombre es único. 2.2 Si no es único ir a flujo alternativo Nombre no Único. 2.3 Agrega el prefijo a los conocidos por el sistema.
Sección "Eliminar Prefijo"	
Acción del Actor	Respuesta del Negocio
2.1 Nombre no único.	2.1.1 Informa que el nombre no es único.
Pos condiciones	Prefijo actualizados.
Prioridad	Crítico.

Tabla 22: Expansión del Caso de Uso Gestionar Prefijos.

Caso de Uso:	Gestionar Configuración Específica.
--------------	-------------------------------------

Características de Sistema

Actores:	Usuarios.
Trabajadores:	-
Resumen:	Gestiona la configuración de un elemento en el entorno virtual.
Referencia:	R3
Precondiciones:	Entorno virtual cargado y elemento seleccionado.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Si selecciona la opción Restablecer Configuración.	1.1 Ir a sección Configuración Por Defecto.
2. Selecciona la opción Aceptar.	2.1 Almacena la información referente al elemento en cuestión.
Sección "Configuración por Defecto"	
Acción del Actor	Respuesta del Negocio
	1. Muestra la configuración por defecto de un elemento del entorno virtual.
Pos condiciones	Elemento configurado.
Prioridad	Crítico.

Tabla 23: Expansión del Caso de Uso Gestionar Configuración Específica.

Caso de Uso:	Vista Previa
Actores:	Usuarios.

Características de Sistema

Trabajadores:	-
Resumen:	Crea una vista previa de lo que será el entorno virtual con las configuraciones actuales.
Referencia:	R9
Precondiciones:	Entorno virtual existe.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. Selecciona la opción vista previa.	<p>1.1 Actualiza la configuración global del entorno virtual.</p> <p>1.2 Actualiza la configuración de cada elemento del entorno virtual, primero la específica y después la de cada grupo.</p> <p>1.3 Actualiza la configuración de cada elemento del entorno virtual según su prefijo.</p>
Pos condiciones	Entorno virtual mostrado con todas sus configuraciones.
Prioridad	Crítico.

Tabla 24: Expansión del Caso de Uso Vista Previa.

Caso de Uso:	Cargar Archivo Configuración Prefijos.
Actores:	Usuarios.
Trabajadores:	-
Resumen:	Lee el archivo Configuración Prefijos, y aplica estas configuraciones al entorno virtual.

Características de Sistema

Referencia:	R14		
Precondiciones:	Existe el archivo.		
Flujo Normal de Eventos			
Acción del Actor			Respuesta del Negocio
1. Selecciona la opción Cargar Configuración de Prefijos.			1.1 Si existe una configuración anterior ir a flujo alternativo Configuración Anterior. 1.2 Solicita el archivo.
2. Especifica el archivo.			2.1 Crea una estructura donde guardar la información. 2.2 Invoca al caso de uso Cargar XML. 2.3 Limpiar las estructuras donde se almacenaran los datos. 2.4 Configura la herramienta con la información leída.
Flujos Alternos			
Acción del Actor			Respuesta del Negocio
Configuración Anterior. 1.1 Especifica su elección.			1.1.1 Pregunta si desea guardar la configuración. 1.1.2 Si no desea guardar la configuración continua en el paso 3 del Flujo Normal de los Eventos. 1.1.3 Invoca caso de uso Generar Archivo Configuración Prefijos.

Características de Sistema

	1.1.4 Continúa en el paso 3 del Flujo Normal de los Eventos.
Pos condiciones	Entorno virtual configurado.
Prioridad	Critico.

Tabla 25: Expansión del Caso de Uso Cargar Archivo Configuración Prefijos.

Caso de Uso:	Generar Archivo Configuración Prefijos.	
Actores:	Usuarios.	
Trabajadores:	-	
Resumen:	Crea un archivo con la información que será utilizada para configurar prefijos que serán utilizados por estos.	
Referencia:	R12	
Precondiciones:	Información disponible.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. Seleccionar la opción generar archivo Configuración Prefijos.	1.1 Muestra los prefijos actuales.	
2. Selecciona los prefijos a generar.		
3. Selecciona la opción Generar Archivo.	3.1 Solicita nombre del archivo y dirección donde se desea almacenar.	
4. Especifica nombre del archivo y la dirección donde se desea almacenar.	4.1 Almacena la información a almacenar en el archivo en una estructura. 4.2 Invoca al caso de uso Salvar XML.	

Pos condiciones	Archivo Configuración Prefijos queda creado con la información deseada en el.
Prioridad	Crítico.

Tabla 26: Expansión del Caso de Uso Generar Archivo Configuración Prefijos.

CAPITULO 3: ANÁLISIS Y DISEÑO DEL SOFTWARE

3.1 Modelo de clases de análisis

Por la extinción de las clases presentaremos solo las más relevantes para la arquitectura. Para un análisis con mayor profundidad de la arquitectura del software todas las clases se colocaran en el ANEXO1.

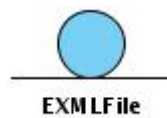


Figura 15: Diagrama de clases de análisis, acceso a datos.

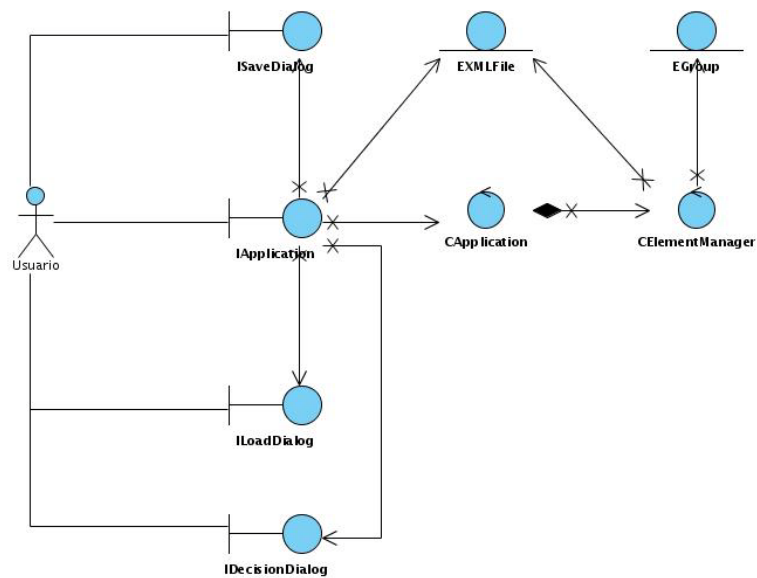


Figura 16: Diagrama de clases de análisis, cargar archivo configuración entorno virtual.

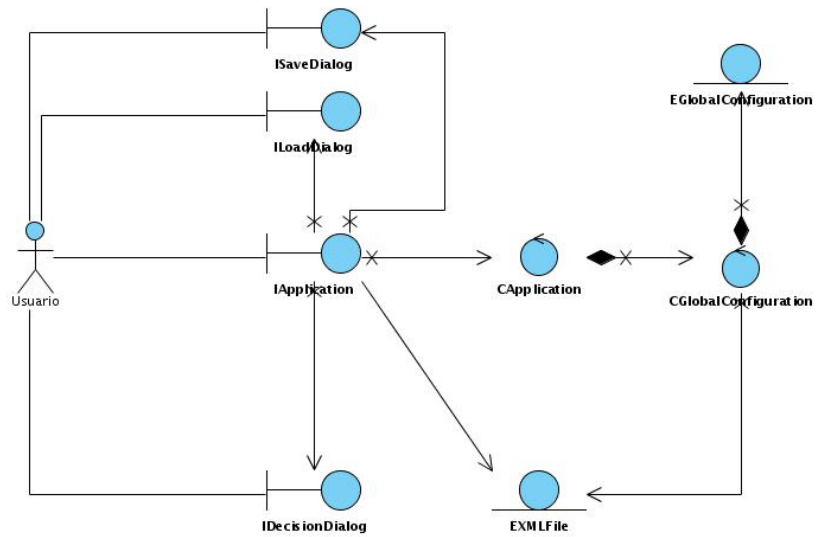


Figura 17: Diagrama de Clases de Análisis Cargar Archivo Configuración Global.

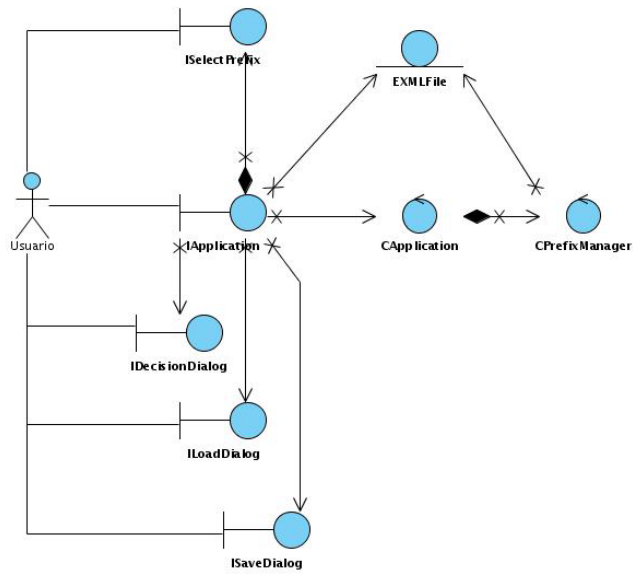


Figura 18: Diagrama de Clases de Análisis Cargar Archivo Configuración Prefijos.

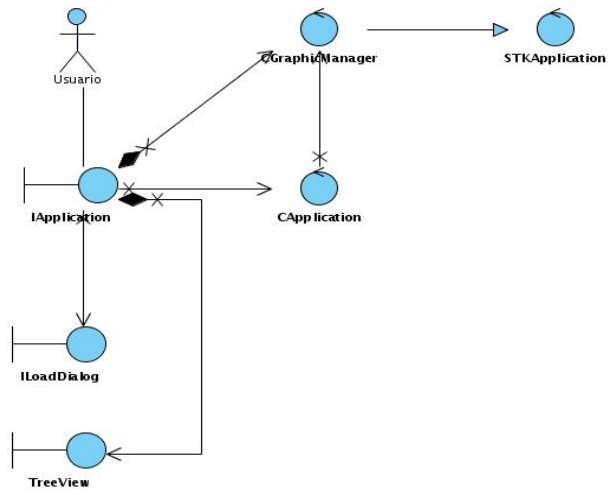


Figura 19: Diagrama de Clases de Análisis Cargar Archivo Entorno Virtual.

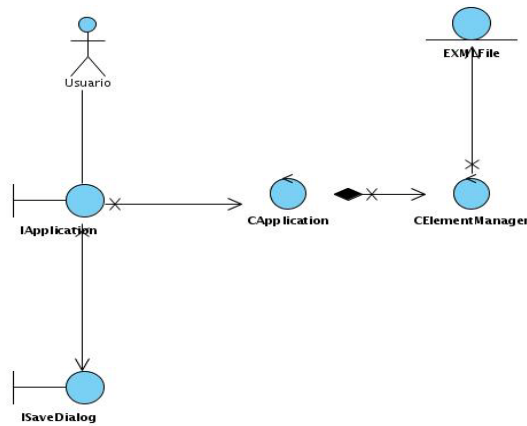


Figura 20: Diagrama de Clases de Análisis Generar Archivo Configuración Entorno Virtual.

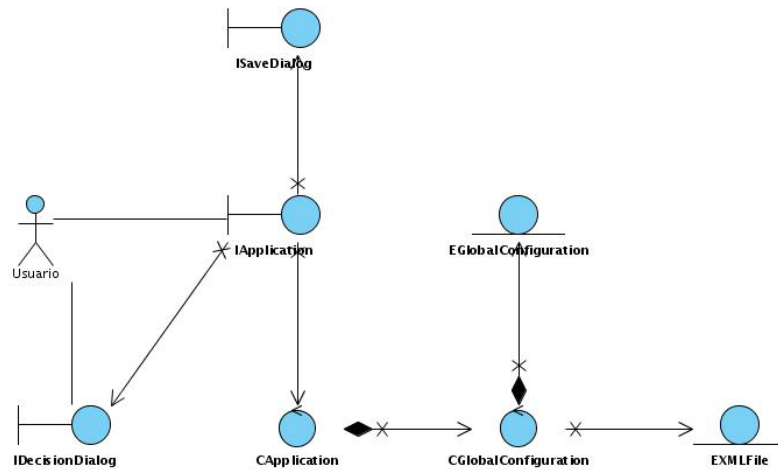


Figura 21: Diagrama de Clases de Análisis Generar Archivo Configuración Global.

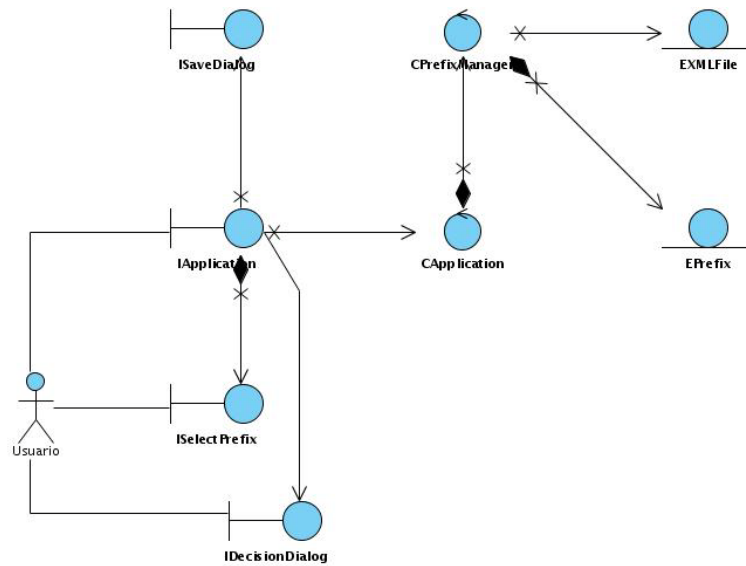


Figura 22: Diagrama de Clases de Análisis Generar Archivo Configuración Prefijos.

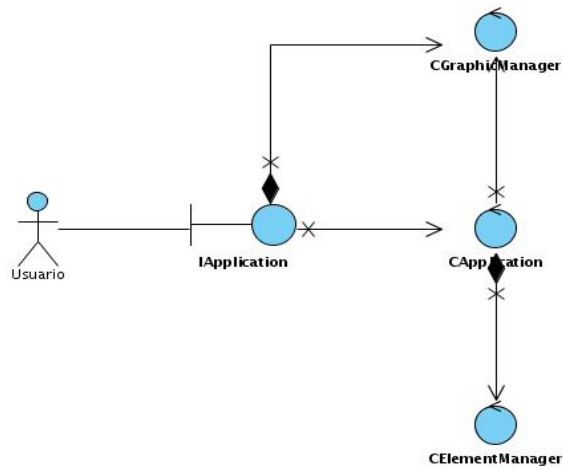


Figura 23: Diagrama de Clases de Análisis Gestionar Configuración Específica.

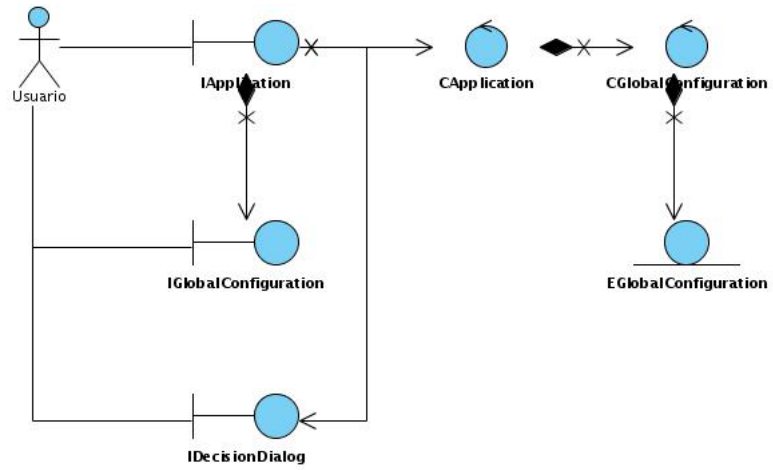


Figura 24 Diagrama de Clases de Análisis Gestionar Configuración Global.

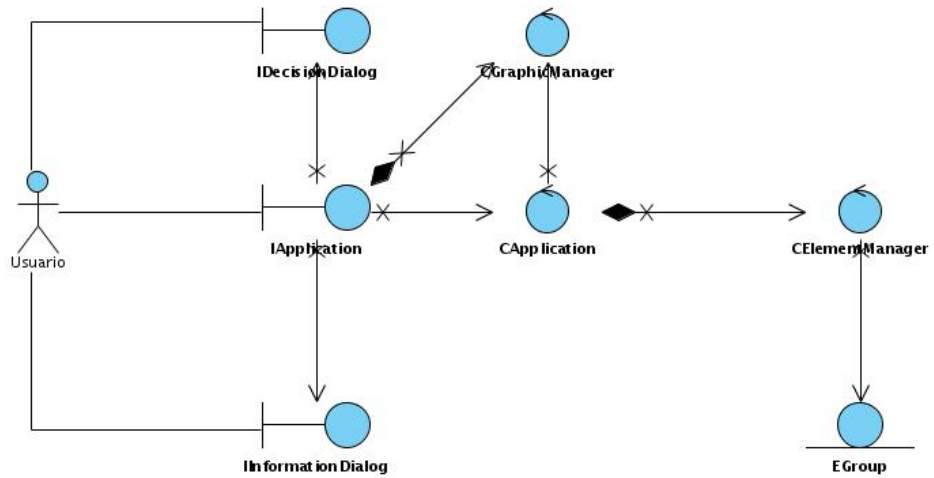


Figura 25 Diagrama de Clases de Análisis Gestionar Grupo de Elementos.

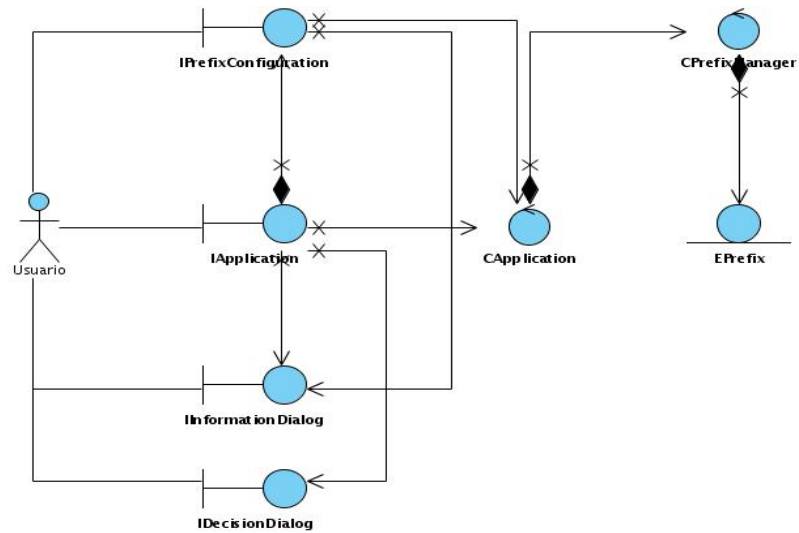


Figura 26: Diagrama de Clases de Análisis Gestionar Prefijos.

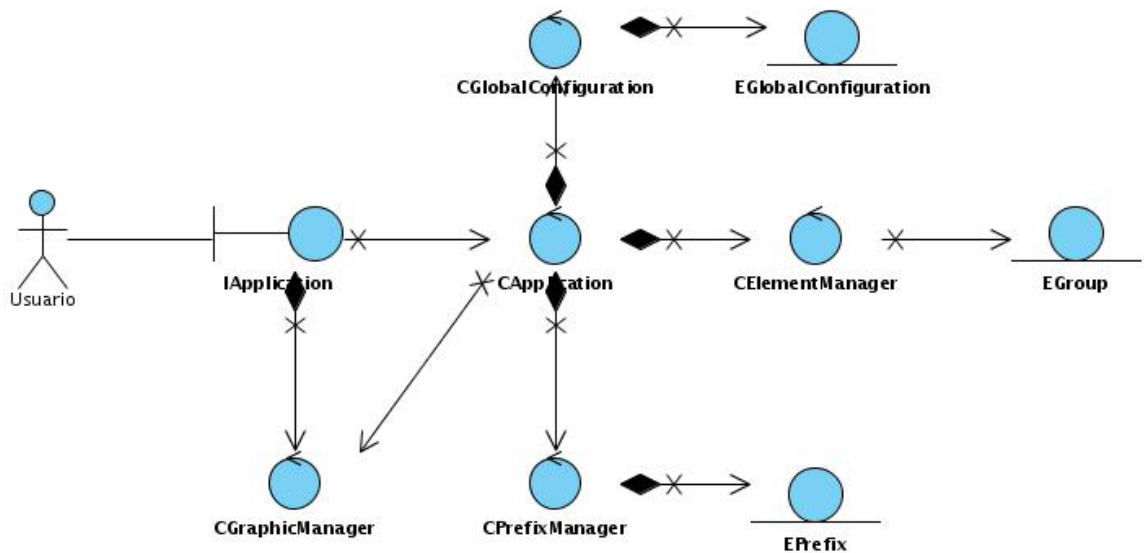


Figura 27: Diagrama de Clases de Análisis Vista Previa.

3.2 Diagrama de interacción

Dado que los diagramas de secuencia son muy extensos, solo se representarán los diagramas más significativos y de mayor importancia para el desarrollo de la aplicación. Para un análisis con mayor profundidad de la arquitectura del software los diagramas que no se exponen a continuación se colocaran en el ANEXO1.

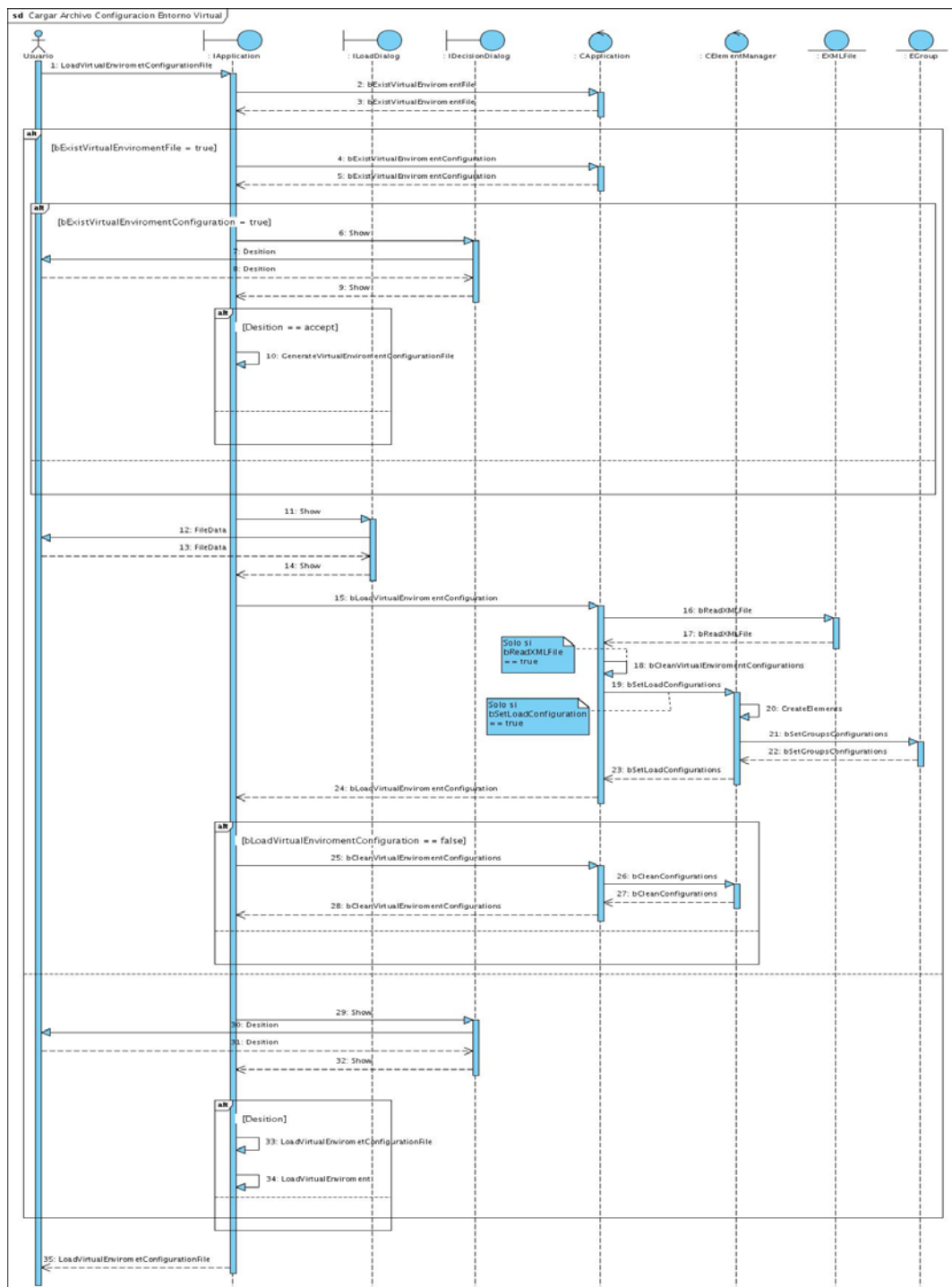


Figura 28: Diagrama de Secuencia Cargar Archivo Configuración Entorno Virtual.

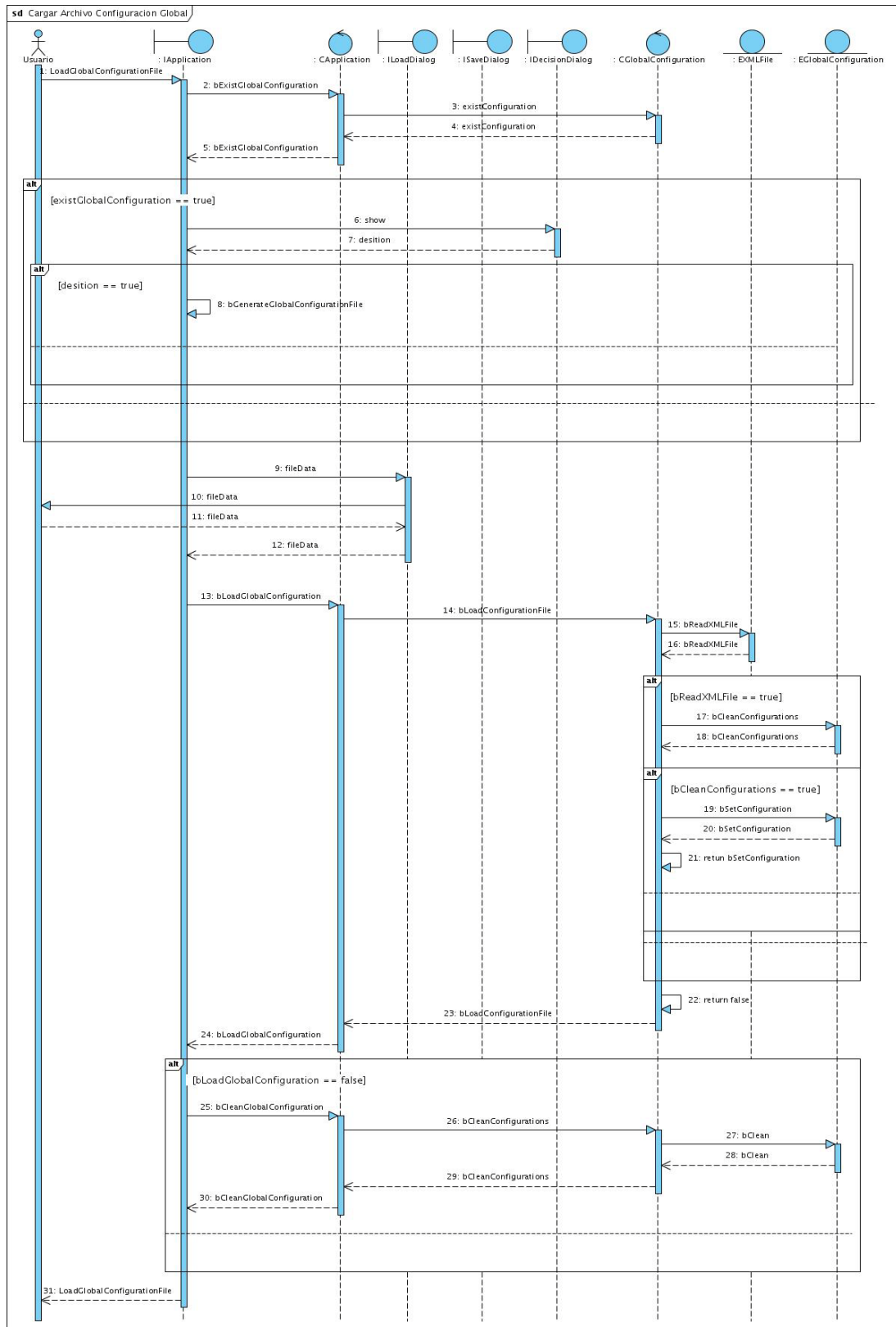


Figura 29: Diagrama de Secuencia Cargar Archivo Configuración Global.

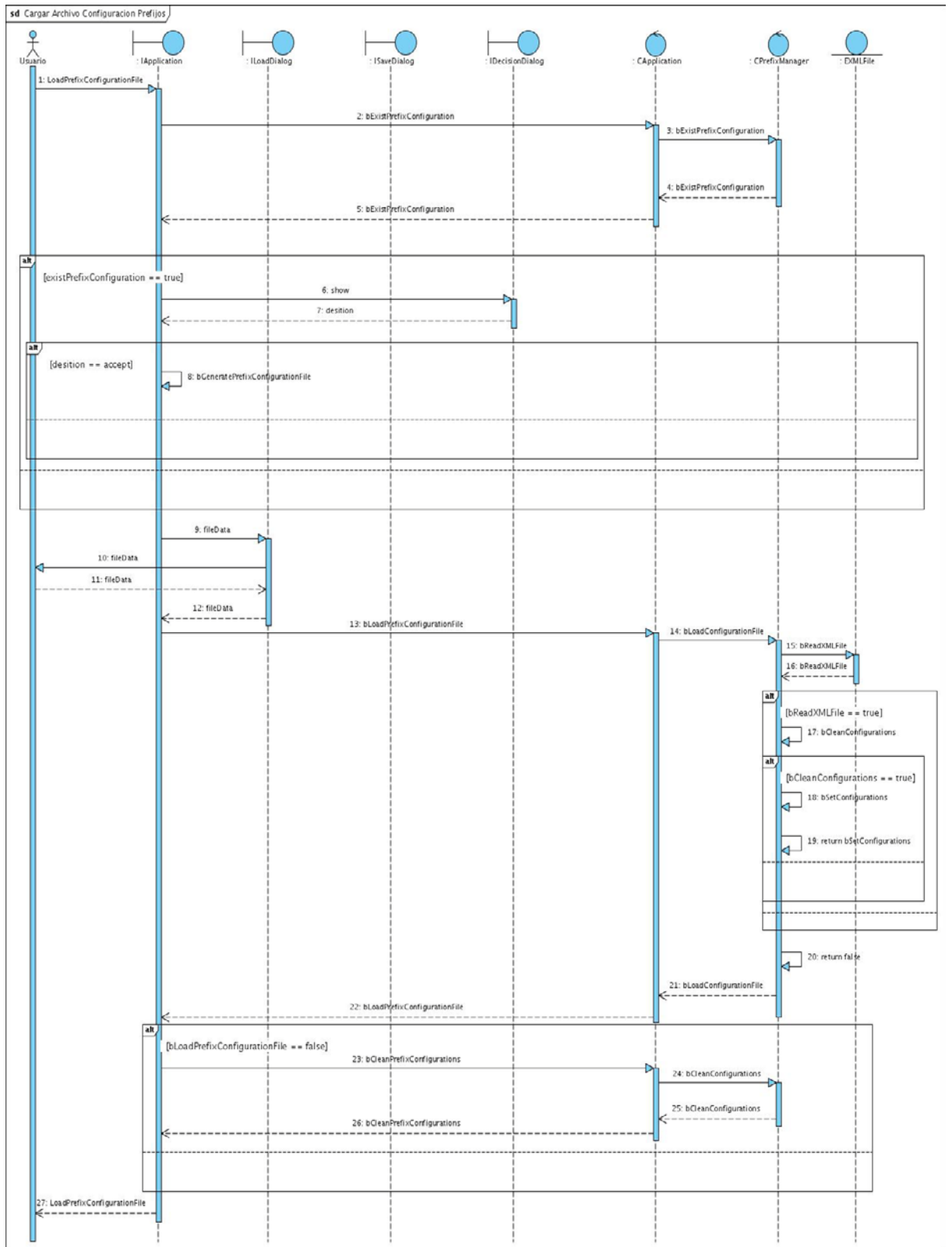


Figura 30: Diagrama de Secuencia Cargar Archivo Configuración Prefijos.

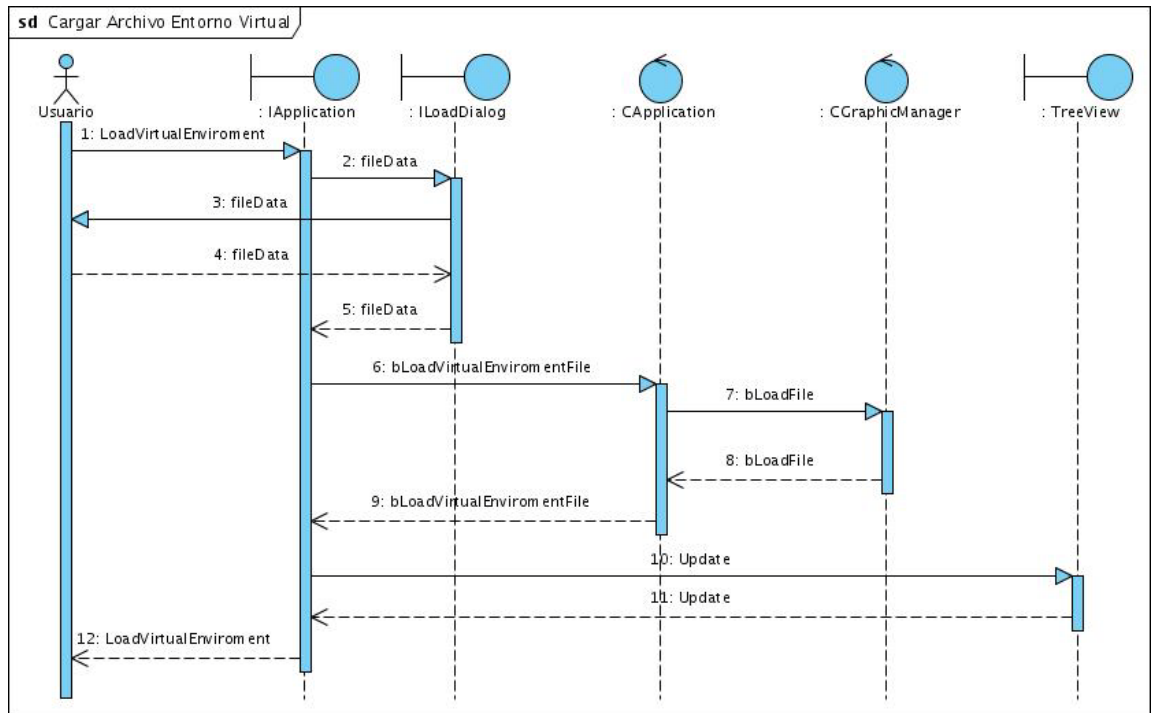


Figura 31: Diagrama de Secuencia Cargar Archivo Entorno Virtual.

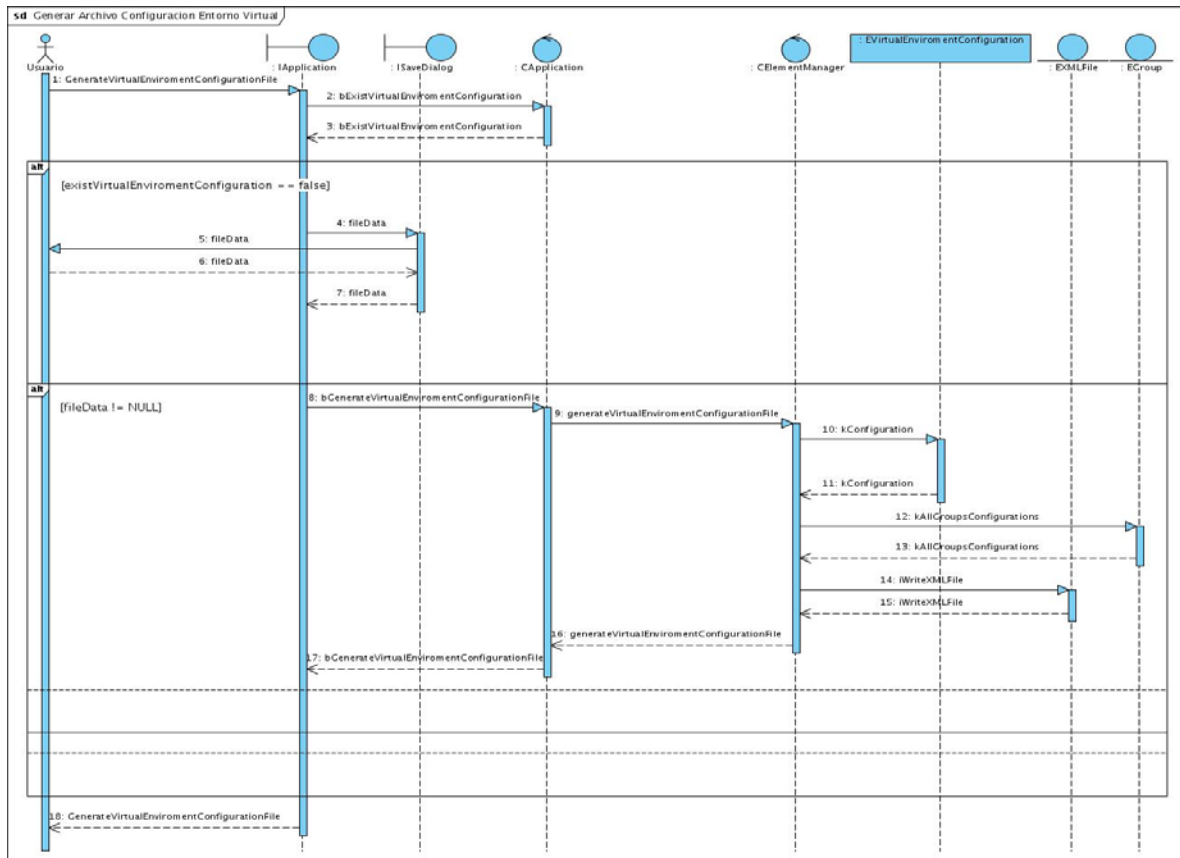


Figura 32: Diagrama de Secuencia Generar Archivo Configuración Entorno Virtual.

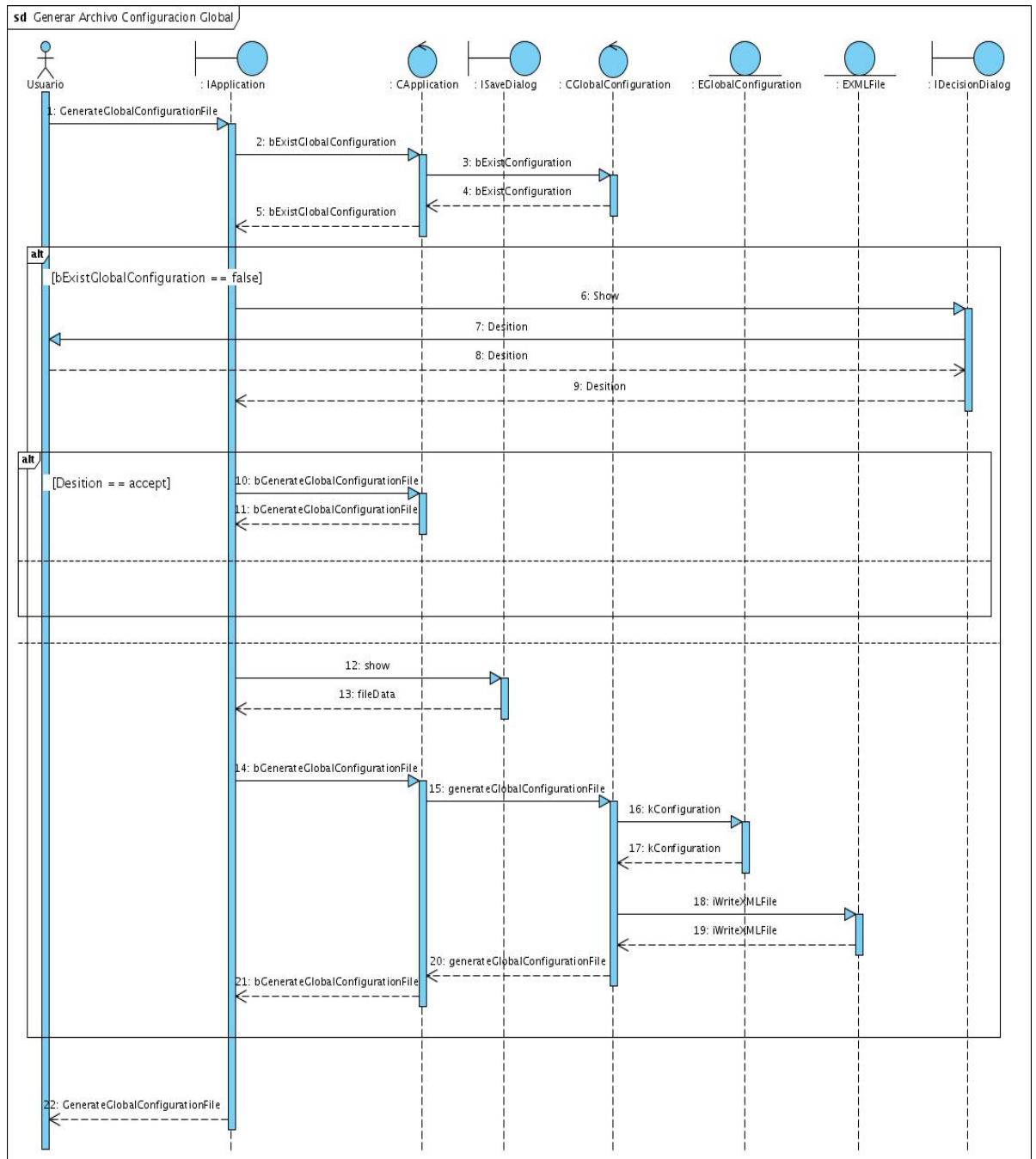


Figura 33: Diagrama de Secuencia Generar Archivo Configuración Global.

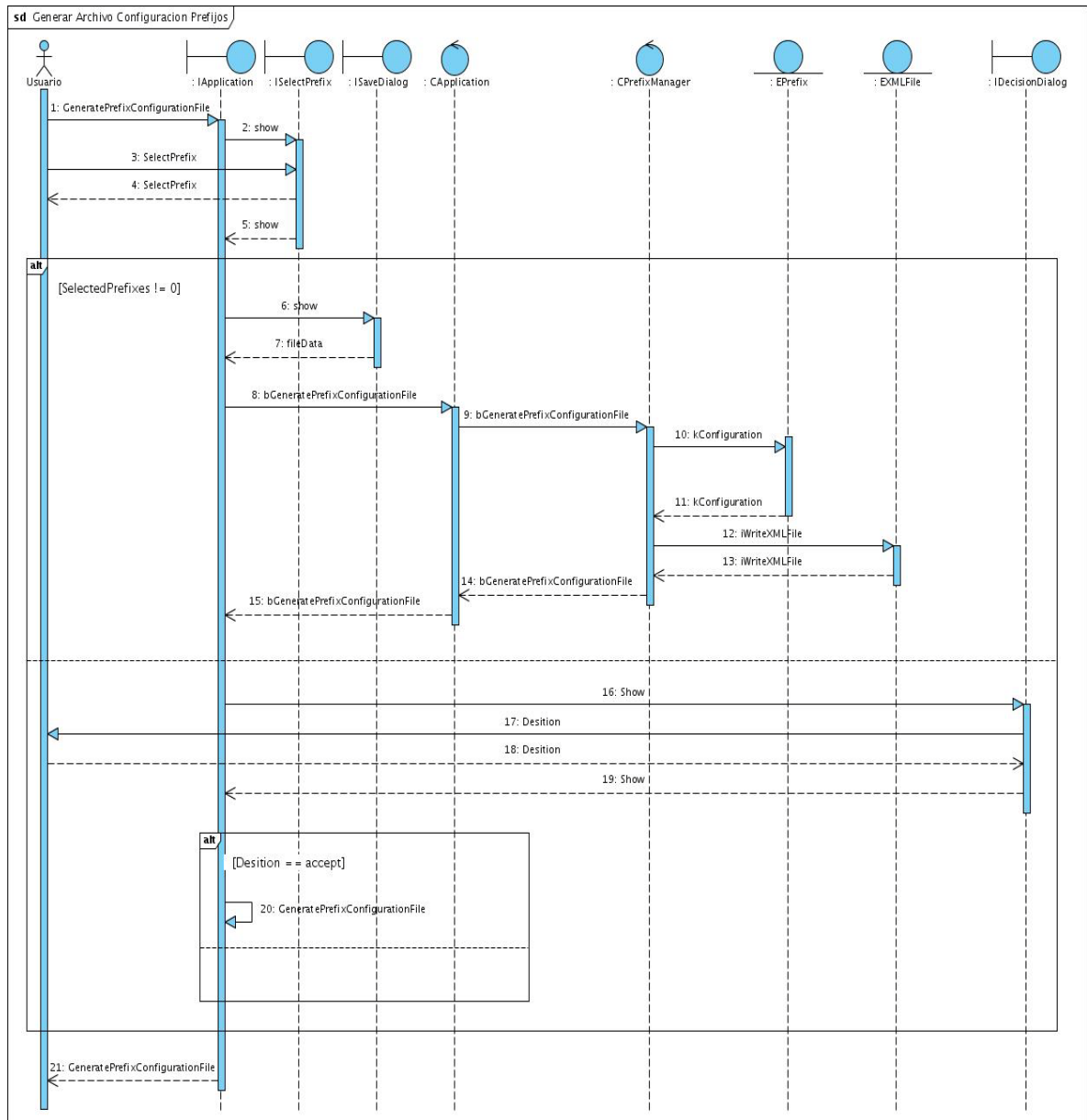


Figura 34: Diagrama de Secuencia Generar Archivo Configuración Prefijos.

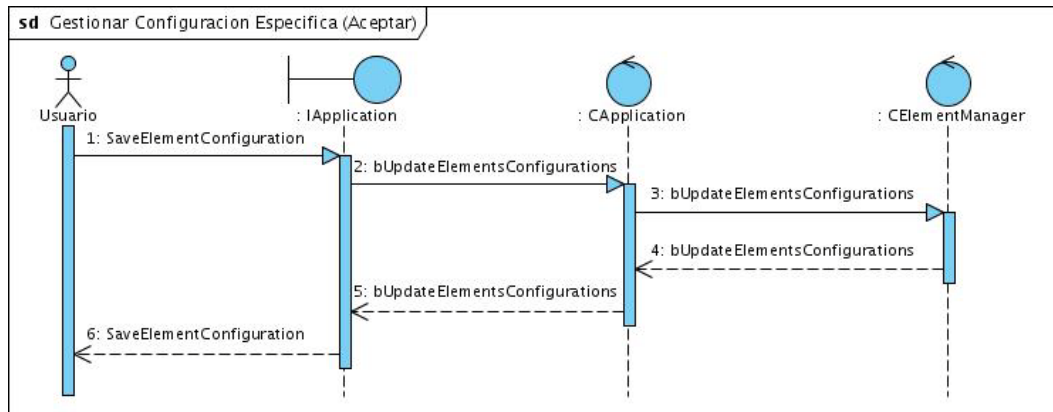


Figura 35: Diagrama de Secuencia Gestionar Configuración Específica.

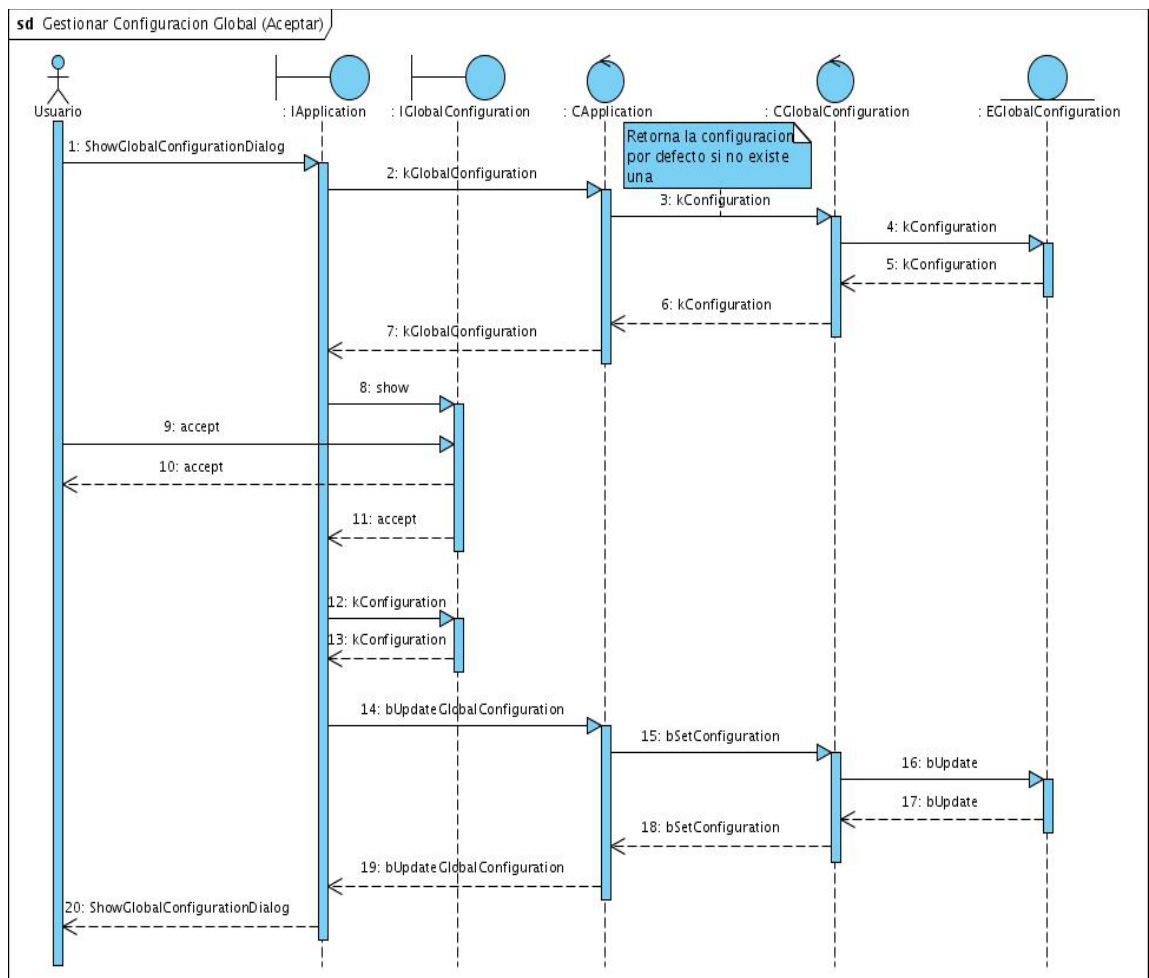


Figura 36: Diagrama de Secuencia Gestionar Configuración Global (Aceptar).

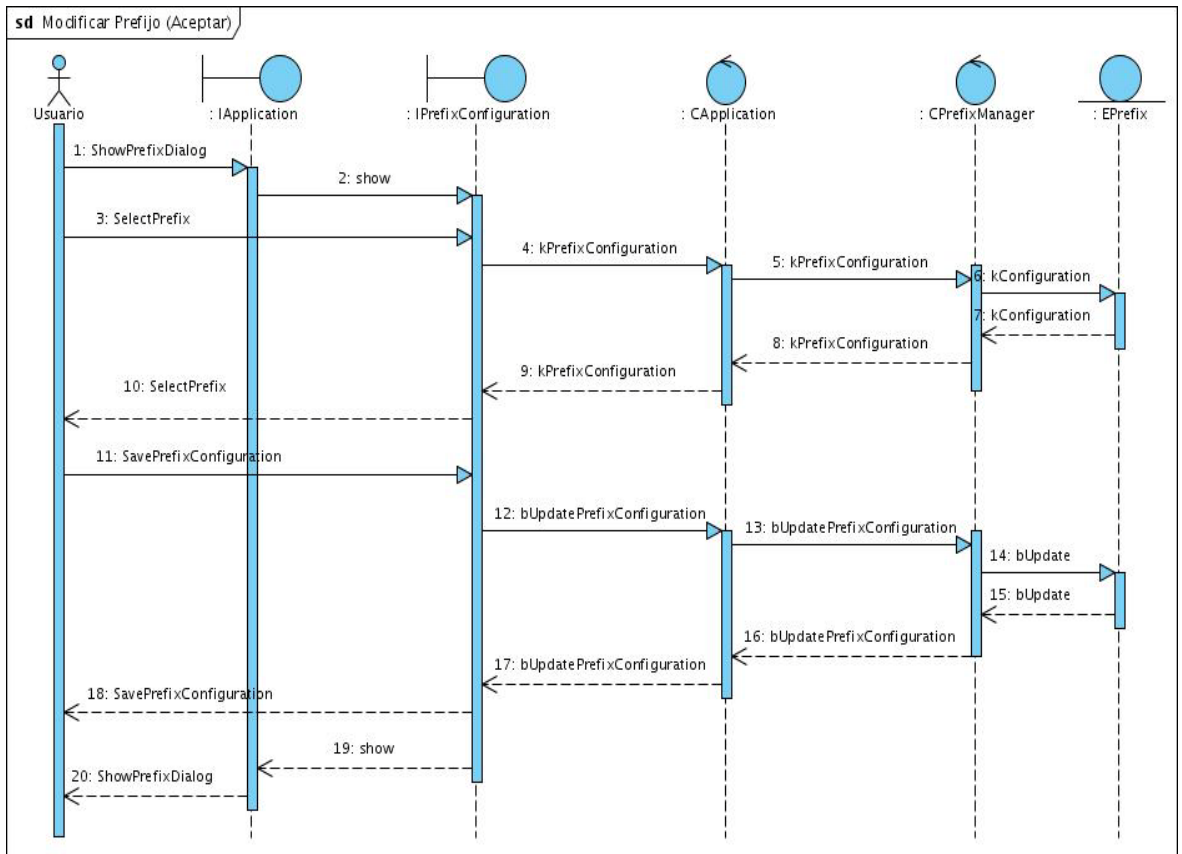


Figura 37: Diagrama de Secuencia Modificar Prefijo (Aceptar).

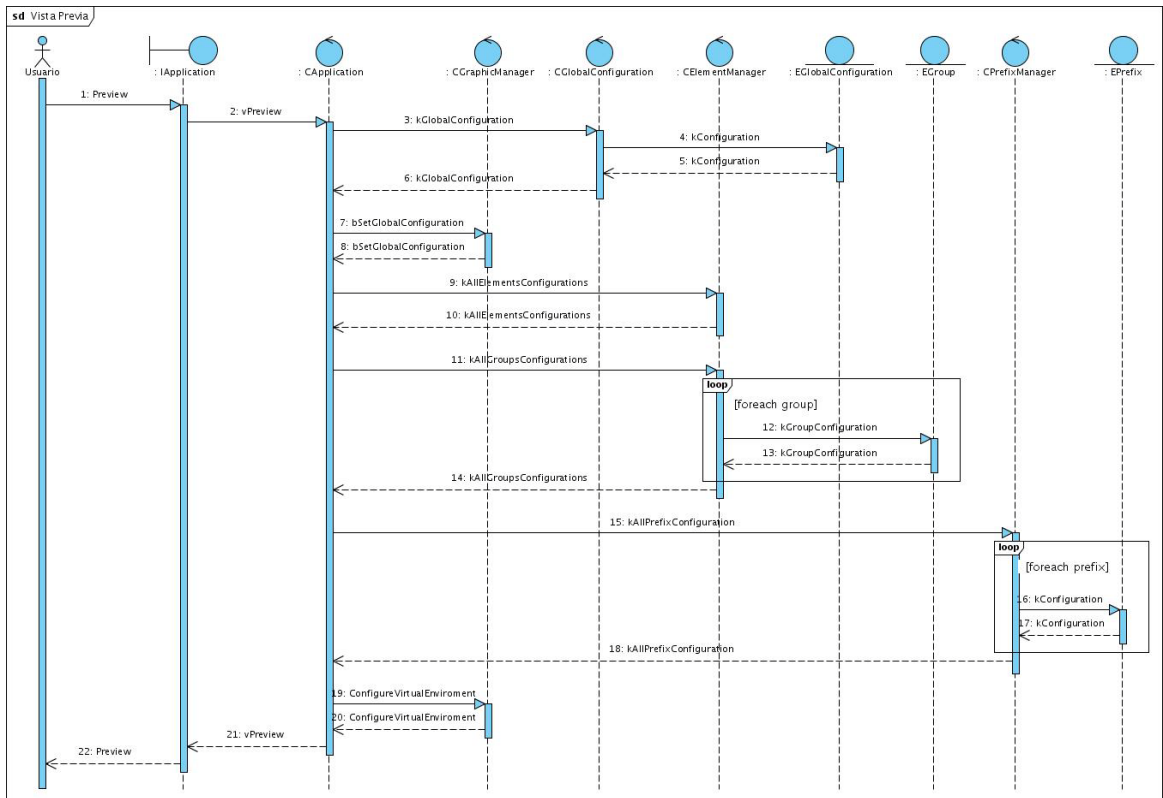


Figura 38: Diagrama de Secuencia Vista Previa.

3.4 Descripción de las clases de diseño en formato expandido

Nombre: CPrefix	
Tipo de clase: entidad	
Atributo	Tipo
m_acNamer	QString
m_pkConfiguration	CElementConfiguration*
Para cada responsabilidad:	
Nombre:	bExistConfiguration
Descripción:	Retorna TRUE si existe configuración para este prefijo y FALSE si no existe configuración.
Nombre:	bClean
Descripción:	Elimina la configuración del prefijo, retorna TRUE si es realizado con éxito o FALSE si falla en el proceso.
Nombre:	bUpdate
Descripción:	Actualiza la configuración del prefijo y retorna TRUE si es realizado con éxito o FALSE si falla en el proceso.
Nombre:	kConfiguration
Descripción:	Retorna la configuración actual del prefijo y si no existe una retorna la configuración por defecto.

Tabla 27: Descripción de la Clase de Diseño CPrefix.

Nombre: CPrefixManager	
Tipo de clase: controladora	
Atributo	Tipo
m_apkPrefix	std::vector<CPrefix*>
Para cada responsabilidad:	

Nombre:	bUpdatePrefixName
Descripción:	Actualiza el nombre de un prefijo especificado por su nombre que es único, retorna TRUE si el nombre es actualizado al nuevo y FALSE si ocurre algún error en el proceso.
Nombre:	bExistPrefixConfiguration
Descripción:	Retorna TRUE si existe al menos un prefijo configurado y FALSE si no existe ninguno.
Nombre:	bCleanConfigurations
Descripción:	Elimina todos los prefijos y con ellos sus configuraciones, retorna TRUE si se eliminan todos los prefijos y FALSE si no se pueden eliminar u ocurre algún error en el proceso.
Nombre:	bSetConfigurations
Descripción:	Agrega nuevos prefijos con sus configuraciones, retorna TRUE si todos son agregados con éxito o FALSE si no se pueden agregar.
Nombre:	bLoadConfigurationFile
Descripción:	Carga un archivo configuración de prefijos limpiando las configuraciones anteriores y agregando las cargadas del archivo. Retorna TRUE si el proceso de carga se realiza con éxito y FALSE si no se puede cargar el archivo.
Nombre:	bGeneratePrefixConfigurationFile
Descripción:	Genera un archivo .stkpc con la configuración de los prefijos en una dirección dada retornando TRUE si el proceso se realiza con éxito o FALSE si no puede generarse el archivo.
Nombre:	bCheckUnicPrefixName
Descripción:	Comprueba si un nombre es único dentro de los prefijos retorna TRUE si es único y FALSE si no es único.
Nombre:	bAddPrefix
Descripción:	Crea un nuevo prefijo, retorna TRUE si es agregado con éxito y FALSE si no puede agregarse. Por seguridad debe comprobarse antes que el nombre del prefijo que se creará sea único ya que de no serlo, creará conflicto en procesos posteriores.

Nombre:	bRemovePrefix
Descripción:	Elimina un prefijo dado por su nombre único retorna TRUE si es correctamente eliminado o FALSE si no se puede eliminar.
Nombre;	kPrefixConfiguration
Descripción:	Retorna la configuración de un prefijo dado por su nombre único si este existe, de no ser así retorna la configuración por defecto
Nombre:	bUpdatePrefixConfiguration
Descripción:	Actualiza la configuración de un prefijo dado por su nombre único, retorna TRUE si el proceso es realizado con éxito o FALSE si no se pudo realizar.
Nombre:	kDefaultPrefixConfiguration
Descripción:	Retorna la configuración por defecto.
Nombre:	kAllPrefixConfiguration
Descripción:	Retorna la configuración de todos los prefijos.

Tabla 28: Descripción de la Clase de Diseño CPrefixManager.

Nombre: CElementConfiguration	
Tipo de clase: entidad	
Atributo	Tipo
m_kConfiguration	QDomDocument*
Para cada responsabilidad:	
Nombre:	kConfiguration
Descripción:	Retorna la configuración que almacena.
Nombre:	bExistConfiguration
Descripción:	Retorna TRUE si existe configuración y FALSE si no existe configuración.
Nombre:	bClean
Descripción:	Elimina la configuración que almacena, retorna TRUE si es correctamente

	eliminada o FALSE si no puede ser eliminada.
Nombre:	bUpdate
Descripción:	Actualiza la configuración que almacena, retorna TRUE si ha sido actualizada con éxito o FALSE si no se ha podido actualizar.

Tabla 29: Descripción de la Clase de Diseño CElementConfiguration.

Nombre: CGroup	
Tipo de clase: entidad	
Atributo	Tipo
m_acGroupName	QString
m_auiElements	std::vector<unsigned int>
m_pkConfiguration	CElementConfiguration*
Para cada responsabilidad:	
Nombre:	bRemoveElements
Descripción:	Elimina varios elementos del grupo, retorna TRUE si son correctamente eliminados todos los elementos deseados o FALSE si falló al eliminar a uno de ellos
Nombre:	bAddElements
Descripción:	Agrega varios elementos al grupo, retorna TRUE si son correctamente agregados al grupo o FALSE si falla al agregar a uno de ellos
Nombre:	kConfiguration
Descripción:	Retorna la configuración que almacena el grupo si esta existe, de lo contrario retorna la configuración por defecto.
Nombre:	bAnyElementInside
Descripción:	Comprueba si existe alguno de los elementos que recibe por parámetro en el grupo, retorna TRUE si encuentra al menos uno o FALSE si no encuentra ninguno.
Nombre:	bUpdateConfiguration

Descripción:	Actualiza la configuración del grupo, retorna TRUE si la actualización se realizó con éxito o FALSE si falló.
--------------	---

Tabla 30: Descripción de la Clase de Diseño CGroup.

Nombre: CGroupManager	
Tipo de clase: controladora	
Atributo	Tipo
m_apkGroups	std::vector<CGroup*>
Para cada responsabilidad:	
Nombre:	bClean
Descripción:	Elimina todos los grupos que contiene y con ellos las configuraciones que les pertenecen, retorna TRUE si eliminó con éxito a todos los grupos o FALSE si falló en el proceso.
Nombre:	bCheckUnicGroupName
Descripción:	Comprueba que el nombre dentro de los grupos actuales sea único retorna TRUE si es único el nombre o FALSE si no es único.
Nombre:	bCheckElementsInAnyGroup
Descripción:	Comprueba si los elementos que recibe por parámetro pertenecen a algún grupo, retorna TRUE si al menos uno de los elementos pertenece a alguno de los grupos o FALSE si ninguno de los elementos se encuentra en alguno de los grupos.
Nombre:	bRemoveElementsFromGroup
Descripción:	Remueve los elementos que recibe por parámetro del grupo que también recibe por parámetro, retorna TRUE si el proceso es ejecutado con éxito o FALSE si no es ejecutado satisfactoriamente.
Nombre:	bCreateGroup
Descripción:	Crea un nuevo grupo con el nombre que recibe por parámetro, retorna TRUE si es creado con éxito o FALSE si no se puede crear. El nombre del grupo debe ser único ya que de no ser así causaría conflicto en procesos posteriores.

Nombre:	bRemoveGroups
Descripción:	Elimina los grupos que corresponden a los nombres que recibe por parámetro, retorna TRUE si los elimina a todos con éxito o FALSE si falla al eliminar al menos uno.
Nombre:	bAddElementsToGroup
Descripción:	Agrega al grupo que coincida con el nombre recibido por parámetro los elementos que son también recibidos por parámetro, retorna TRUE si son agregados todos con éxito o FALSE si falla el proceso.
Nombre:	auiFilterElementsNotInGroup
Descripción:	Retorna de los elementos recibidos por parámetros los que no pertenezcan a ningún grupo.
Nombre:	kAllGroupsConfiguration
Descripción:	Retorna todas las configuraciones de los grupos.
Nombre:	kDefaultConfiguration
Descripción:	Retorna la configuración por defecto.
Nombre:	bUpdateConfiguration
Descripción:	Actualiza la configuración del grupo que corresponda con el nombre que es recibido por parámetro.
Nombre:	kGroupConfiguration
Descripción:	Retorna la configuración del grupo que corresponda con el nombre recibido por parámetro
Nombre:	auiElementsInGroup
Descripción:	Retorna los elementos que pertenecen al grupo que corresponda con el nombre recibido por parámetro.
Nombre:	bSetGroupsConfigurations
Descripción:	Crea grupos con los nombres recibidos por parámetros y los actualiza con las configuraciones también recibidas por parámetro, retorna TRUE si se crean todos con éxito o FALSE si falla el proceso.
Nombre:	bExistConfiguration

Descripción:	Retorna TRUE si existe configuración o FALSE si no existe
--------------	---

Tabla 31: Descripción de la Clase de Diseño CGroupManager.

Nombre: CElementManager	
Tipo de clase: controladora	
Atributo	Tipo
m_pkGroupManager	CGroupManager*
m_apkCElementConfiguration	std::vector<CElementConfiguration*>
Para cada responsabilidad:	
Nombre:	bExistVirtualEnviromentConfiguration
Descripción:	Retorna TRUE si existe configuración o FALSE si no existe.
Nombre:	bCleanConfiguration
Descripción:	Elimina todas las configuraciones y retorna TRUE si el proceso se realizó con éxito o FALSE si falló.
Nombre:	bUpdateElementsConfigurations
Descripción:	Actualiza las configuraciones de los elementos recibidos por parámetro con las configuraciones recibidas también por parámetro.
Nombre:	bGenerateVirtualEnviromentConfigurationFile
Descripción:	Genera un archivo .stkvec con el nombre recibido por parámetro en la dirección también recibida por parámetro, retorna TRUE si el proceso se realizó con éxito o FALSE si falló.
Nombre:	kElementConfiguration
Descripción:	Retorna la configuración del elemento que corresponda con el recibido por parámetro, de no existir configuración para este se retorna la configuración por defecto.
Nombre:	kAllElementsConfigurations
Descripción:	Retorna todas las configuraciones de los elementos.

Nombre:	bLoadConfigurationFile
Descripción:	Carga un fichero .stkvec limpiando todas las configuraciones y actualizando con las cargadas del archivo, retorna TRUE si el proceso fue realizado con éxito y FALSE si falló.

Tabla 32: Descripción de la Clase de Diseño CElementManager.

Nombre: CConfigurationEditor	
Tipo de clase: controladora	
Atributo	Tipo
m_pkElementManager	CElementManager*
m_pkGlobalConfigurationManager	CGlobalConfigurationManager*
m_pkGraphicManager	IGraphicManager*
m_pkPrefixManager	CPrefixManager*
Para cada responsabilidad:	
Nombre:	bLoadApplicationConfigurationFile
Descripción:	Carga datos que necesita la aplicación para ejecutarse, retorna TRUE si la carga fue satisfactoria o FALSE si falló.

Tabla 33: Descripción de la Clase de Diseño CConfigurationEditor.

Nombre: CGraphicManager	
Tipo de clase: controladora	
Atributo	Tipo
m_acFileAddress	QString
m_auSelectedElements	unsigned int[]
Para cada responsabilidad:	
Nombre:	DrawSelectedElements

Descripción:	Renderea los boundingbox de los elementos seleccionados.
Nombre:	bSetGlobalConfiguration
Descripción:	Aplica la configuración global y retorna TRUE si el proceso se realizó con éxito o FALSE si falló al aplicarse.
Nombre:	ConfigureVirtualEnviroment
Descripción:	Aplica la configuración de entorno virtual.
Nombre:	auiElementsWithPrefix
Descripción:	Retorna los elementos que tengan como prefijo la cadena recibida por parámetro.
Nombre:	ResetElementsConfiguration
Descripción:	Restaura por defecto la configuración de los elementos que correspondan con los recibidos por parámetros.

Tabla 34: Descripción de la Clase de Diseño CGraphicManager.

Nombre: CConfigurationController	
Tipo de clase: controladora	
Para cada responsabilidad:	
Nombre:	OnMouseDown
Descripción:	Ejecuta acciones cuando algunos de los botones del mouse son presionados sobre la ventana de render.
Nombre:	OnMouseUp
Descripción:	Ejecuta acciones cuando algunos de los botones del mouse son presionados y después liberados sobre la ventana de render.
Nombre:	OnMouseMove
Descripción:	Ejecuta acciones cuando el mouse se mueve sobre la ventana de render y esta está activada
Nombre:	OnKeyDown

Descripción:	Ejecuta acciones cuando alguna tecla del teclado es presionada mientras la ventana de render esta activada
Nombre:	OnKeyUp
Descripción:	Ejecuta acciones cuando alguna tecla del teclado después de ser presionada y liberada mientras la ventana de render está activada.
Nombre:	OnWheel
Descripción:	Ejecuta acciones cuando se acciona la ruedita del mouse sobre la ventana de render mientras está activa.

Tabla 35: Descripción de la Clase de Diseño CConfigurationController.

Nombre: CGlobalConfigurationManager	
Tipo de clase: controladora	
Atributo	Tipo
m_pkGlobalConfiguration	CGlobalConfiguration*
Para cada responsabilidad:	
Nombre:	bExistConfiguration
Descripción:	Retorna TRUE si existe una configuración o FALSE si no existe.
Nombre:	bCleanConfigurations
Descripción:	Elimina la configuración global actual, retorna TRUE si es proceso se realizo con éxito o FALSE si falló
Nombre:	bSetConfiguration
Descripción:	Actualiza la configuración global, retorna TRUE si el proceso se realiza con éxito o FALSE si falló.
Nombre:	bLoadConfigurationFile
Descripción:	Carga un archivo .stkgc que se encuentra en la dirección recibida por parámetro y guarda la configuración que almacena en la configuración global. Retorna TRUE si el proceso de carga se realizó con éxito o FALSE si falló.

Nombre:	bGenerateGlobalConfigurationFile
Descripción:	Genera archivo llamado según el nombre recibido por parámetro y en la dirección también recibida por parámetro y retorna TRUE si el proceso se realiza con éxito o FALSE si falla.
Nombre:	kConfiguration
Descripción:	Retorna la configuración global.
Nombre:	kDefaultConfiguration
Descripción:	Retorna la configuración por defecto.

Tabla 36: Descripción de la Clase de Diseño CGlobalConfigurationManager.

Nombre: CGlobalConfiguration	
Tipo de clase: entidad	
Atributo	Tipo
m_kConfiguration	QDomDocument
Para cada responsabilidad:	
Nombre:	bClean
Descripción:	Elimina la configuración.
Nombre:	bUpdate
Descripción:	Actualiza la configuración.
Nombre:	kConfiguration
Descripción:	Retorna la configuración.

Tabla 37: Descripción de la Clase de Diseño CGlobalConfiguration.

Nombre: CXMLFile	
Tipo de clase: controladora	
Atributo	Tipo

m_pkSingleton	CXMLFile*
Para cada responsabilidad:	
Nombre:	bReadXMLFile
Descripción:	Lee un archivo XML que se encuentra en la dirección recibida por parámetro, retorna en una variable por referencia los datos leídos del XML, retorna TRUE si el proceso de lectura se realizó con éxito o FALSE si falló.
Nombre:	bWriteXMLFile
Descripción:	Escribe los datos recibidos por parámetros en un archivo XML llamado según el nombre recibido también por parámetro en la dirección especificada por el mismo parámetro, retorna TRUE si el proceso de escritura se realizó con éxito o FALSE si falla.

Tabla 38: Descripción de la Clase de Diseño CXMLFile.

CAPITULO 4: IMPLEMENTACION Y PRUEBAS

4.1 Estándar de codificación

Como estándar de codificación se escogió el mismo estándar utilizado en el desarrollo de la biblioteca base de esta aplicación, es decir, la biblioteca Scene Tool Kit (STK). Para un mayor entendimiento del mismo, en el manual de usuario de esta biblioteca se describe detalladamente dicho estándar de codificación.

4.2 Diagrama de despliegue

Dada la simplicidad del diagrama de despliegue para este sistema, ya que incluye solo una PC, se decidió que no era necesario realizar una representación de dicho diagrama.

4.3 Diagrama de componentes

Para lograr un mayor entendimiento y organización del diagrama de componentes se dividió en, una vista binaria donde se muestra la relación entre los componentes binarios, y una vista de código fuente, la cual está diseñada siguiendo la estructura física de los archivos de código fuente.

La vista binaria está representada en la siguiente figura.

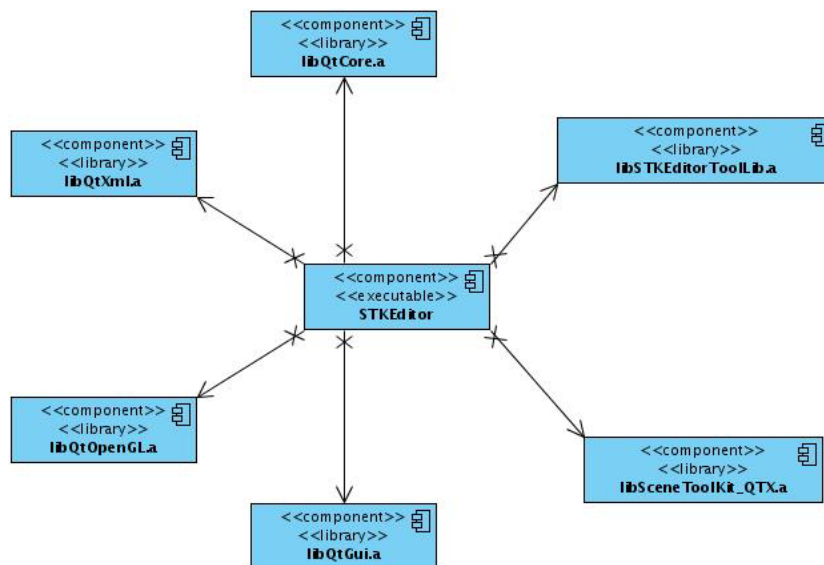


Figura 40: Diagrama de Componentes de la Vista Binaria.

La estructura física de los ficheros de código está representada en la siguiente imagen.

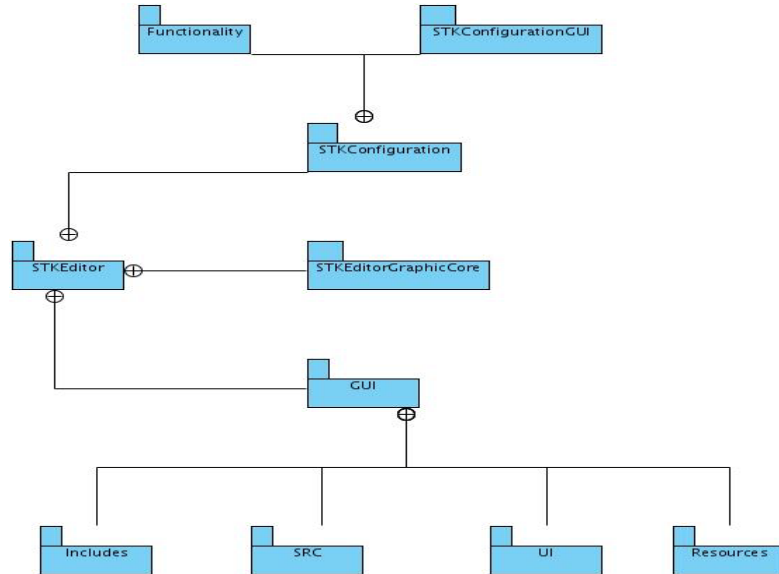


Figura 41: Estructura física de los ficheros de código fuente.

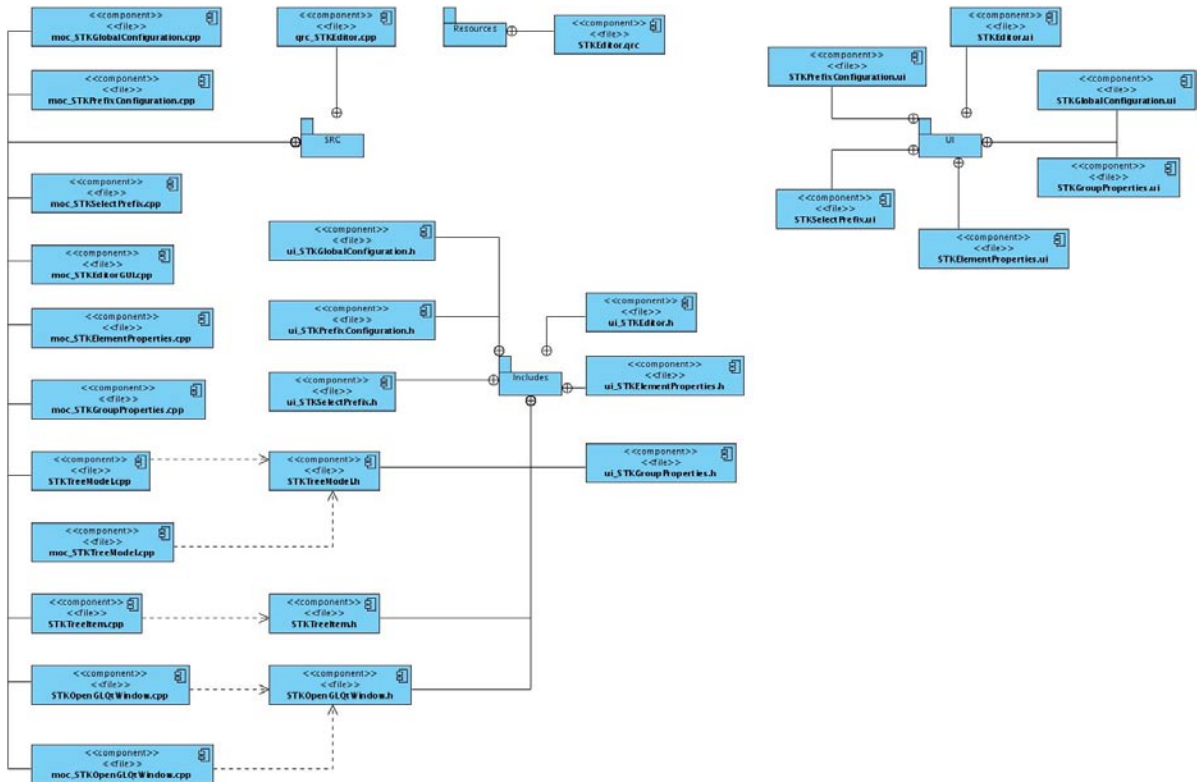


Figura 42: Diagrama de Componente de la carpeta GUI.

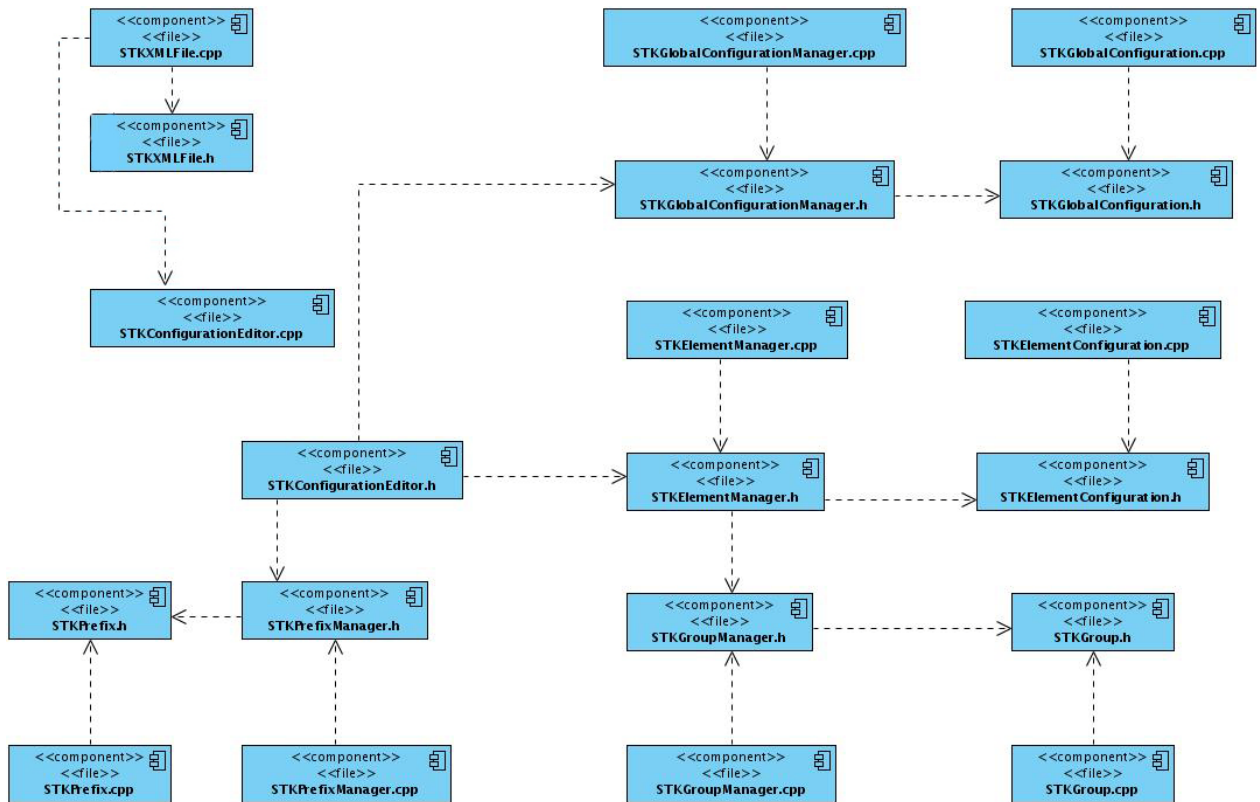


Figura 43: Diagrama de Componente de la Carpeta Functionality.

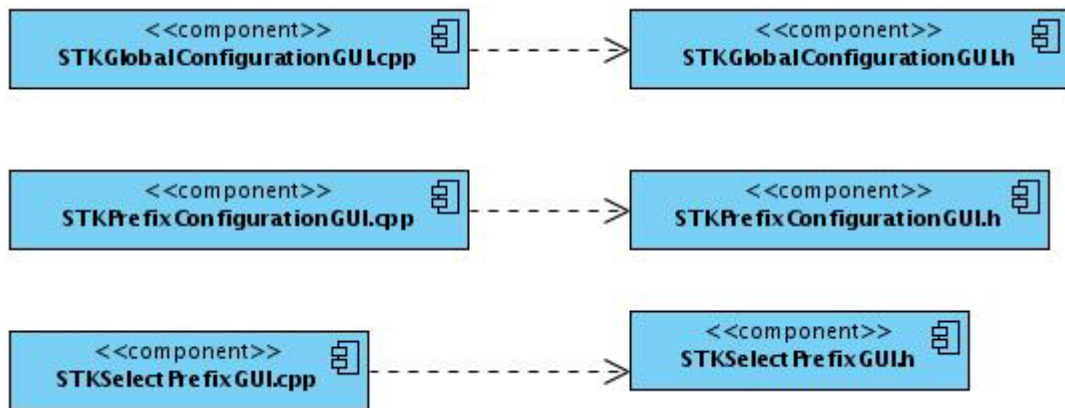


Figura 44: Diagrama de Componente de la Carpeta STKConfigurator GUI.

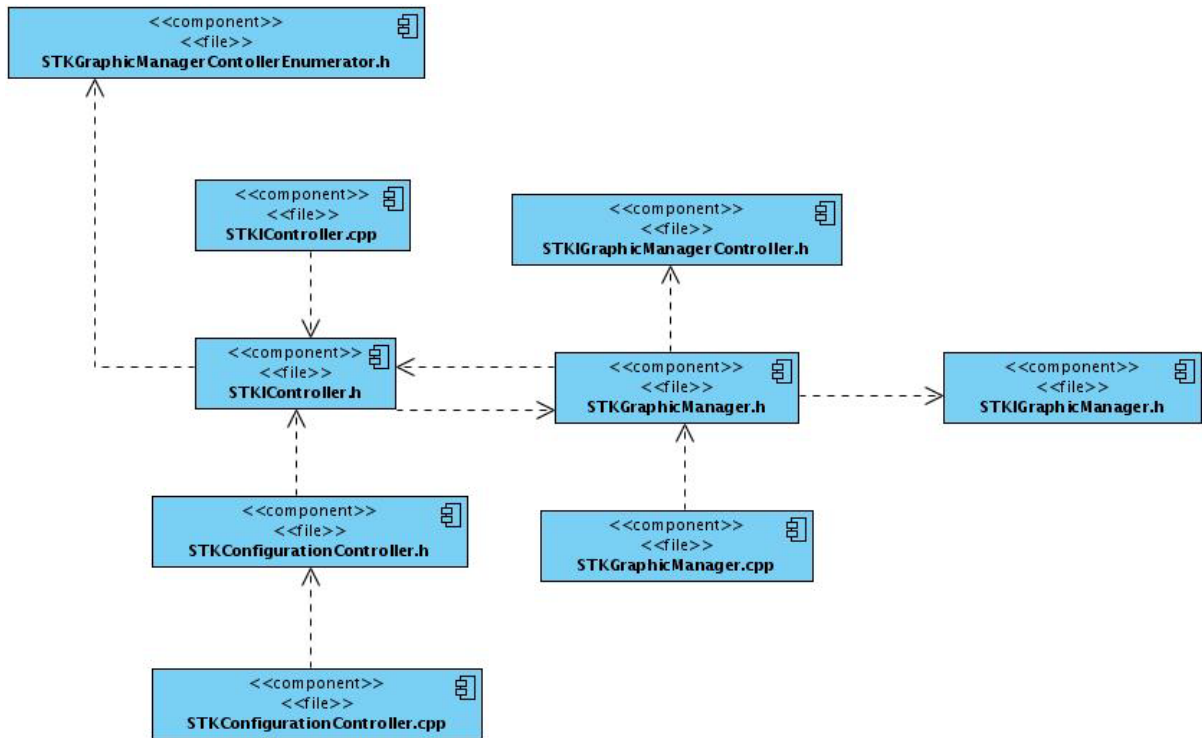


Figura 45: Diagrama de Componente de la Carpeta STKEditorGraphicCore.

4.4 Modelos de pruebas

Caso de Uso: Generar Archivo Configuración Entorno Virtual.

Descripción General: Verifica que el proceso de generación de la configuración del entorno virtual sea realizado con éxito.

Condiciones de Ejecución: Entorno Virtual Cargado.

Secciones a aprobar en el caso de uso.

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Generar archivo configuración entorno virtual.	1.1 Existe configuración.	Genera archivo configuración entorno virtual correctamente.	Seleccionar acción "Generar Archivo configuración Entorno Virtual". Entra el nombre y dirección donde se creará el archivo. Se genera el archivo correctamente.

	1.2 No existe configuración.	Muestra mensaje informativo sobre la configuración del entorno virtual.	Seleccionar acción “Generar Archivo configuración Entorno Virtual”. Muestra mensaje informativo, “Debe configurar un entorno virtual primero para guardar su configuración”.
--	------------------------------	---	---

Tabla 39: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Entorno Virtual.

Caso de Uso: Generar Archivo Configuración Prefijos.

Descripción General: Verifica que el proceso de generación de la configuración de los prefijos sea realizada con éxito.

Condiciones de Ejecución: Aplicación funcionando correctamente.

Secciones a probar en el caso de uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Generar archivo configuración prefijos.	1.1 Existe al menos un prefijo.	Genera archivo configuración prefijos correctamente.	Seleccionar acción “Generar Archivo configuración Prefijos”. Entra el nombre y dirección donde se creará el archivo. Se genera el archivo correctamente.
	1.2 No existen prefijos.	Muestra mensaje informativo sobre la configuración de los prefijos.	Seleccionar acción “Generar Archivo configuración Prefijos”. Muestra mensaje informativo “Debe primero configurar los prefijos”.

Tabla 40: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Prefijos.

Caso de Uso: Generar Archivo Configuración Global.

Descripción General: Verifica que el proceso de generación de la configuración global sea realizado con éxito.

Condiciones de Ejecución: Aplicación funcionando correctamente.

Secciones a probar en el caso de uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Generar archivo configuración global.	1.1 Existe configuración.	Genera archivo configuración global correctamente.	<p>Seleccionar acción “Generar Archivo configuración Global”.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se genera el archivo correctamente.</p>
	1.2 No existe configuración y genera configuración por defecto.	Muestra mensaje interrogativo sobre la configuración global.	<p>Seleccionar acción “Generar Archivo configuración Global”.</p> <p>Muestra mensaje interrogativo “No existe una configuración. Desea guardar un archivo con la configuración por defecto”.</p> <p>La respuesta es acertada.</p> <p>Se genera el archivo correctamente con la configuración por defecto.</p>
	1.3 No existe configuración.	Muestra mensaje interrogativo sobre la configuración global.	<p>Seleccionar acción “Generar Archivo configuración Global”.</p> <p>Muestra mensaje interrogativo “No existe una configuración. Desea guardar un archivo con la configuración por defecto”.</p> <p>La respuesta no es acertada.</p>

Tabla 41: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Global.

Caso de Uso: Cargar Archivo Configuración Entorno Virtual.

Descripción General: Verifica que el proceso de carga de la configuración del entorno virtual sea realizado con éxito.

Condiciones de Ejecución: Entorno Virtual Cargado.

Secciones a probar en el caso de uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Cargar archivo configuración entorno virtual.	1.1 Existe configuración, se genera archivo configuración entorno virtual y se carga archivo configuración entorno virtual.	Informa que existe una configuración entorno virtual y pregunta si desea almacenarla. La respuesta es acertada y se genera el archivo configuración entorno virtual. Se carga el archivo configuración entorno virtual.	<p>Seleccionar acción "Cargar Archivo configuración Entorno Virtual".</p> <p>Informa que existe configuración de entorno virtual y pregunta si desea almacenar esta configuración.</p> <p>La respuesta es acertada.</p> <p>Genera archivo configuración entorno virtual.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración entorno virtual.</p>
	1.2 Existe configuración, no se genera archivo configuración entorno virtual y se carga archivo configuración entorno virtual.	Informa que existe una configuración entorno virtual y pregunta si desea almacenarla. La respuesta no es acertada. Se carga el archivo configuración entorno virtual.	<p>Seleccionar acción "Cargar Archivo configuración Entorno Virtual".</p> <p>Informa que existe configuración de entorno virtual y pregunta si desea almacenar esta configuración.</p> <p>La respuesta no es acertada.</p> <p>Entra el nombre y dirección donde se creará el archivo</p> <p>Se carga el archivo configuración entorno virtual.</p>
	1.3 No existe configuración y se carga archivo configuración entorno virtual	Se carga el archivo configuración entorno virtual.	<p>Seleccionar acción "Cargar Archivo configuración Entorno Virtual".</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración entorno virtual.</p>

Tabla 42: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Entorno Virtual.

Caso de Uso: Cargar Archivo Configuración Prefijos.

Descripción General: Verifica que el proceso de carga de la configuración de los prefijos sea realizado con éxito.

Condiciones de Ejecución: Aplicación funcional.

Secciones a probar en el caso de uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Cargar archivo configuración prefijos.	1.1 Existe configuración , se genera archivo configuración prefijos y se carga archivo configuración prefijos.	Informa que existe configuración de prefijos y pregunta si desea almacenarla. La respuesta es acertada y se genera el archivo configuración prefijos. Se carga el archivo configuración prefijos.	<p>Seleccionar acción “Cargar Archivo configuración Prefijos”.</p> <p>Informa que existe configuración de prefijos y pregunta si desea almacenar esta configuración.</p> <p>La respuesta es acertada.</p> <p>Genera archivo configuración prefijos</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración prefijos.</p>
	1.2 Existe configuración , no se genera archivo configuración prefijos y se carga archivo configuración prefijos.	Informa que existe una configuración de prefijos y pregunta si desea almacenarla. La respuesta no es acertada. Se carga el archivo configuración prefijos.	<p>Seleccionar acción “Cargar Archivo configuración Prefijos”.</p> <p>Informa que existe configuración de prefijos y pregunta si desea almacenar esta configuración.</p> <p>La respuesta no es acertada.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración prefijos.</p>

	1.3 No existe configuración y se carga archivo configuración prefijos.	Se carga el archivo configuración prefijos.	<p>Seleccionar acción “Cargar Archivo configuración Prefijos”.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración. prefijos</p>
--	--	---	--

Tabla 43: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Prefijos.

Caso de Uso: Cargar Archivo Configuración Global.

Descripción General: Verifica que el proceso de carga de la configuración global sea realizado con éxito.

Condiciones de Ejecución: Aplicación funcional.

Secciones a probar en el caso de uso:

Nombre de la sección	Escenarios de la sección	Descripción de la funcionalidad	Flujo Central
1. Cargar archivo configuración global.	1.1 Existe configuración, se genera archivo configuración global y se carga archivo configuración global.	Informa que existe una configuración global y pregunta si desea almacenarla. La respuesta es acertada y se genera el archivo configuración global. Se carga el archivo configuración global.	<p>Seleccionar acción “Cargar Archivo configuración Global”.</p> <p>Informa que existe configuración de global y pregunta si desea almacenar esta configuración.</p> <p>La respuesta es acertada.</p> <p>Genera archivo configuración global.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración global.</p>

	<p>1.2 Existe configuración, no se genera archivo configuración global y se carga archivo configuración global.</p>	<p>Informa que existe una configuración global y pregunta si desea almacenarla. La respuesta no es acertada. Se carga el archivo configuración global.</p>	<p>Seleccionar acción "Cargar Archivo configuración Global".</p> <p>Informa que existe configuración de global y pregunta sí desea almacenar esta configuración.</p> <p>La respuesta no es acertada.</p> <p>Entra el nombre y dirección donde se creará el archivo.</p> <p>Se carga el archivo configuración global.</p>
	<p>1.3 No existe configuración y se carga archivo configuración global.</p>	<p>Se carga el archivo configuración global.</p>	<p>Seleccionar acción "Cargar Archivo configuración Global".</p> <p>Entra el nombre y dirección donde se creara el archivo.</p> <p>Se carga el archivo configuración global.</p>

Tabla 44: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Global.

CONCLUSIONES

Para el cumplimiento de los objetivos de este proyecto, en concordancia con las necesidades de la herramienta base para la cual este fue concebido, se realizó un estudio en varias ramas del desarrollo de los Sistemas de Realidad Virtual, así como la construcción de Interfaces Gráficas de Usuario. Primeramente efectuamos un estudio sobre la construcción de interfaces gráficas de usuario, las principales técnicas y tendencias en cuanto al diseño de las mismas, así como los principales pasos y principios que se requieren a la hora de desarrollar estas interfaces. También fue necesario hacer un estudio acerca de las principales bibliotecas de desarrollo de sistemas de realidad virtual, enfatizando en sus principales características, así como un estudio sobre aquellas bibliotecas que poseen interfaces gráficas de usuario, que apoyan el desarrollo de estos sistemas de realidad virtual. Junto a esto se requirió realizar una investigación acerca de las principales características del formato XML, así como las ventajas que brinda para el almacenamiento de información. Proponiendo así un conjunto de soluciones técnicas, que posibilitarían llevar una medida de lo que se obtendría al final del recorrido del proyecto.

Seguidamente se constituyó el proceso de Ingeniería de Software, haciendo uso de la Metodología de Desarrollo de Software RUP (Proceso Unificado Software). Se realizó el proceso de captura de requisitos, así como la identificación de los casos de usos del sistema junto con su descripción en formato expandido, se diseñaron las clases y se creó el diagrama de componentes, contenedor de las clases del sistema.

Finalmente, se obtuvo una Interfaz Gráfica de Usuario, la cual cumple con todos los requisitos que se propusieron, permitiéndole al usuario final, una fácil configuración de los entornos virtuales, guardar dicha configuración en un fichero con formato XML, llegando así a reducir en gran medida el esfuerzo requerido por parte de los programadores a la hora de realizar dichas configuraciones a los diferentes entornos virtuales.

RECOMENDACIONES

Se le recomiendan los siguientes aspectos al trabajo:

- Estudiar algoritmos de selección 3d para optimizar y lograr mayor eficiencia en el algoritmo aplicado por los autores.
- Introducir arquitectura de plugins para crear mayor flexibilidad e integración entre las distintas herramientas.
- Convertir en plugin a STKConfigurator para mejorar su flexibilidad en desarrollo y cumplir con la proposición anterior.
- Hacer más comfortable la interfaz visual de usuario siguiendo criterios de los usuarios finales para lograr mayor productividad y aceptación.

BIBLIOGRAFÍA CONSULTADA

1. **Roman, Yanoski Camacho.** *SceneToolKit, Herramienta Básica para Desarrollo de Sistemas de Realidad Virtual.* Noviembre 2007.

2. Qt Reference Documentation (Open Source Edition). *trolltech.com.* [Online]
<http://doc.trolltech.com/4.0/index.html>.

REFERENCIAS BIBLIOGRÁFICAS

1. **Toledo, Carlos Aimacaña.** Interfaz de Usuario. *monografias.com*. [Online] [Cited: 12 6, 2007.] <http://www.monografias.com/trabajos6/inus/inus.shtml>.
2. Interfaz Grafica de Usuario. *wikipedia.org*. [Online] [Cited: 12 10, 2007.] http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario.
3. What is Ogre. *ogre3d.org*. [Online] [Cited: 12 10, 2007.] http://www.ogre3d.org/index.php?option=com_content&task=view&id=19&Itemid=105.
4. DirectX. *wikipedia.org*. [Online] [Cited: 12 08, 2007.] <http://es.wikipedia.org/wiki/DirectX>.
5. Features. *ogre3d.org*. [Online] [Cited: 12 10, 2007.] http://www.ogre3d.org/index.php?option=com_content&task=view&id=13&Itemid=128.
6. Documentation and Support. *g3d-cpp.sourceforge.net*. [Online] [Cited: 12 06, 2007.] <http://g3d-cpp.sourceforge.net/html/>.
7. Documentation and Support. *g3d-cpp.sourceforge.net*. [Online] [Cited: 12 06, 2007.] <http://g3d-cpp.sourceforge.net/html/guideintro.html>.
8. OgreStudio. *ogre3d.org*. [Online] [Cited: 12 06, 2007.] http://www.ogre3d.org/wiki/index.php/OGRE_Studio.
9. GameStudio. *conitec.com*. [Online] [Cited: 12 06, 2007.] <http://www.conitec.com/english/gstudio/index.php>.
10. Qt (biblioteca). *wikipedia.org*. [Online] [Cited: 01 15, 2008.] [http://es.wikipedia.org/wiki/Qt_\(biblioteca\)](http://es.wikipedia.org/wiki/Qt_(biblioteca)).
11. Gtk. *wikipedia.org*. [Online] [Cited: 01 14, 2008.] <http://es.wikipedia.org/wiki/GTK%2B>.
12. XML. *wikipedia.org*. [Online] [Cited: 01 14, 2008.] <http://es.wikipedia.org/wiki/XML>.

13. GameStudio/A7. *conitec.com*. [Online] [Cited: 12 06, 2007.]
<http://www.conitec.com/english/gstudio/3dgs7.php>.
14. La realidad virtual. *monografias.com*. [Online] [Cited: 11 25, 2007.]
<http://www.monografias.com/trabajos4/realvirtual/realvirtual.shtml> .
15. VRML - Realidad Virtual. *activamente.com.mx*. [Online] [Cited: 11 26, 2007.]
<http://www.activamente.com.mx/vrml/>.
16. Conceptos de presencia e interactividad en objetos de aprendizaje de RV. *docente.ucol.mx*. [Online] [Cited: 11 25, 2007.] <http://docente.ucol.mx/~mgarcia/ponencias.html>.
17. Concepto de Realidad Virtual. *monografias.com*. [Online] [Cited: 11 24, 2007.]
<http://www.monografias.com/trabajos11/realitua/realitua.shtml>.
18. **Expósito, Carlos Marrero**. Interfaz gráfica de usuario: Aproximación semiótica y cognitiva. *www.chr5.com*. [Online] [Cited: 12 09, 2007.]
http://www.chr5.com/investigacion/investiga_igu/index_igu.html.
19. **Gómez, Leopoldo Sebastián M**. Diseño de Interfaces de Usuario: Principios, Prototipos y Heurísticas para Evaluación. *monografias.com*. [Online] [Cited: 12 08, 2007.]
<http://www.monografias.com/trabajos10/diusuar/diusuar.shtml>.

APÉNDICES

Glosario de Abreviaturas

3D: Es la abreviatura de tridimensional y en todo contexto se utiliza para hacer referencia e algo que posea o pueda ser representado en tres dimensiones.

API: Application Programming Interface o Interfaz de Programación de Aplicaciones.

GUI: Graphic User Interface o Interfaz Grafica de Usuario.

CUI: Command Line User Interface o Interfaz de Usuario de Línea de Comando.

XML: Extensible Markup Language o Lenguaje de Marcas Extensible, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

STK: SceneToolKit, herramienta desarrollada en el proyecto "Herramienta de Desarrollo de Sistemas de Realidad Virtual", perteneciente a la Facultad 5 de la Universidad de las Ciencias Informáticas.

KDE: K Desktop Environment o Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

GPL: La GNU General Public License en inglés: Licencia Pública General, es una licencia creada por la Free Software Foundation y orientada principalmente a los términos de distribución, modificación y uso de software.

GTK: GTK+ o The GIMP Toolkit es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI).

SGML: SGML son las siglas de Standard Generalized Markup Language o "Lenguaje de Marcación Generalizado".

IDE: IDE son las siglas de: Integrated Development Environment u Entorno Integrado de Desarrollo.

QT: Es una biblioteca multiplataforma para desarrollar interfaces gráficas de usuario.

Glosario de Términos

B:

Background: Fondo según su interpretación para el español. Según el contexto se habla de un proceso en segundo plano cuando este se ejecuta sin nuestra interactividad o se pasa a modo suspendido.

Boundary: Es una frontera que incluye un espacio o un concepto abstracto.

BoundingBox: Es una frontera que define un dominio mínimo rectangular para un objeto 3D en particular.

C:

Cedega: El programa Cedega está específicamente diseñado para ejecutar juegos de ordenador escritos para Microsoft Windows bajo GNU/Linux. Como tal, su principal objetivo es implementar la API de DirectX.

D:

DirectX: Es una colección de APIs creadas y recreadas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo en la plataforma Microsoft Windows.

E:

Engine: Se usa en el lenguaje de los videojuegos para definir el tipo de entorno gráfico de un juego.

Estados de Render: Información de estados de dibujo utilizados por el renderer para representar las mallas de la escena.

G:

GNU/Linux: Nombre por el que se conoce al sistema operativo formado por el conjunto de utilidades de sistema GNU y el núcleo Linux más otras aplicaciones libres creadas por terceros.

L:

Lite-C: Es un lenguaje de programación dedicado a la creación de aplicaciones multimedia y juegos de computadoras, con una sintaxis similar a al lenguaje C.

M:

Multitextura: Es el hecho de aplicarle varias texturas a un objeto que luego será renderizado en una escena 3D.

Multi-Threading: Es una forma de paralelismo donde múltiples hilos son controlados simultáneamente y se comunican mediante la memoria compartida.

O:

Octree: Es una estructura de datos en forma de árbol, en la cual cada nodo interno contiene hasta 8 nodos hijos. Entre sus usos más frecuentes se encuentra el de dividir un espacio tridimensional en ocho octantes.

P:

Pipelines: Los Pipelines o Tuberías Gráficas, son unidades de cálculo especializadas, que determinan el número de píxeles que puede procesar una tarjeta por ciclo de reloj.

Plugins: Un plugin o plug-in en inglés, es una aplicación informática que interactúa con otra aplicación para aportarle una función o utilidad específica, como por ejemplo servir como driver (controlador) en una aplicación, para hacer así funcionar un dispositivo en otro programa.

R:

Renderizar: Es un proceso de cálculo complejo desarrollado por un ordenador destinado a generar una imagen 2D a partir de una escena 3D.

S:

Shaders: Un Shader, es un conjunto de instrucciones gráficas destinadas para el acelerador gráfico, estas instrucciones dan el aspecto final de un objeto. Los Shaders determinan materiales, efectos, color, luz, sombra y etc.

Sistema de Realidad Virtual: Sistema informático interactivo que ofrece una percepción sensorial al usuario de un mundo tridimensional sintético que suplanta al real.

T:

Terragen: Terragen es un programa de software diseñado para generar escenarios y paisajes en diferentes entornos.

Textura: Imagen que sirve de “piel” a los modelos en un mundo virtual.

W:

Widget: Window Gadget, término genérico para referirse a los elementos de una interfaz gráfica con el usuario, como botones, barras de desplazamiento, campos de entrada de texto, etc.

ÍNDICE DE FIGURAS

<i>Figura 1: Estructura del fichero de configuración global.</i>	36
<i>Figura 2: Estructura del fichero de configuración de entorno virtual.</i>	36
<i>Figura 3: Estructura del fichero de configuración de prefijos.</i>	37
Figura 4: Modelo de dominio del sistema.	41
<i>Figura 5: Diagrama de caso de uso, cargar archivo entorno virtual.</i>	48
<i>Figura 6: Diagrama de caso de uso, cargar archivo configuración global.</i>	48
<i>Figura 7: Diagrama de caso de uso, cargar archivo de configuración prefijos.</i>	49
Figura 8: Diagrama de caso de uso, cargar archivo entorno virtual.	49
Figura 9: Diagrama de caso de uso, generar archivo configuración entorno virtual.	49
<i>Figura 10: Diagrama de caso de uso, generar archivo configuración global.</i>	50
<i>Figura 11: Diagrama de caso de uso, generar archivo configuración prefijos.</i>	50
Figura 12: Diagrama de caso de uso, gestionar configuración.	50
Figura 13: Diagrama de caso de uso, gestionar grupo de elementos.	51
Figura 14: Diagrama de caso de uso, vista previa.	51
Figura 16: Diagrama de clases de análisis, acceso a datos.	71
Figura 16: Diagrama de clases de análisis, cargar archivo configuración entorno virtual.	71
Figura 17: Diagrama de Clases de Análisis Cargar Archivo Configuración Global.	72
Figura 18: Diagrama de Clases de Análisis Cargar Archivo Configuración Prefijos.	72
Figura 19: Diagrama de Clases de Análisis Cargar Archivo Entorno Virtual.	73
Figura 20: Diagrama de Clases de Análisis Generar Archivo Configuración Entorno Virtual.	73
Figura 21: Diagrama de Clases de Análisis Generar Archivo Configuración Global.	73
Figura 22: Diagrama de Clases de Análisis Generar Archivo Configuración Prefijos.	74
Figura 23: Diagrama de Clases de Análisis Gestionar Configuración Específica.	74
Figura 24 Diagrama de Clases de Análisis Gestionar Configuración Global.	75
Figura 25 Diagrama de Clases de Análisis Gestionar Grupo de Elementos.	75
Figura 26: Diagrama de Clases de Análisis Gestionar Prefijos.	76
Figura 27: Diagrama de Clases de Análisis Vista Previa.	76
Figura 28: Diagrama de Secuencia Cargar Archivo Configuración Entorno Virtual.	77
Figura 29: Diagrama de Secuencia Cargar Archivo Configuración Global.	78
Figura 30: Diagrama de Secuencia Cargar Archivo Configuración Prefijos.	79
Figura 31: Diagrama de Secuencia Cargar Archivo Entorno Virtual.	80
Figura 32: Diagrama de Secuencia Generar Archivo Configuración Entorno Virtual.	81
Figura 33: Diagrama de Secuencia Generar Archivo Configuración Global.	82
Figura 34: Diagrama de Secuencia Generar Archivo Configuración Prefijos.	83
Figura 35: Diagrama de Secuencia Gestionar Configuración Específica.	84
Figura 36: Diagrama de Secuencia Gestionar Configuración Global (Aceptar).	84
Figura 37: Diagrama de Secuencia Modificar Prefijo (Aceptar).	85
Figura 38: Diagrama de Secuencia Vista Previa.	86
Figura 39: Diagrama de clases de diseño.	87
Figura 40: Diagrama de Componentes de la Vista Binaria.	100
Figura 41: Estructura física de los ficheros de código fuente.	101
Figura 42: Diagrama de Componente de la carpeta GUI.	101
Figura 43: Diagrama de Componente de la Carpeta Functionality.	102
Figura 44: Diagrama de Componente de la Carpeta STKConfigurator GUI.	102

Figura 45: Diagrama de Componente de la Carpeta STKEditorGraphicCore. 103

ÍNDICE DE TABLAS

Tabla 1: Definición del actor del sistema.....	44
Tabla 2: Caso de Uso Cargar Archivo Configuración Entorno Virtual.....	44
Tabla 3: Caso de Uso Generar Archivo Configuración Entorno Virtual.....	44
Tabla 4: Caso de Uso Cargar Archivo Entorno Virtual.....	45
Tabla 5: Caso de Uso Cargar Archivo Configuración Global.....	45
Tabla 6: Caso de Uso Generar Archivo Configuración Global.....	45
Tabla 7: Caso de Uso Gestionar Grupo Elementos.	46
Tabla 8: Caso de Uso Gestionar Configuración Global.....	46
Tabla 9: Caso de Uso Gestionar Prefijos.	46
Tabla 10: Caso de Uso Gestionar Configuración Específica.	46
Tabla 11: Caso de Uso Vista Previa.	47
Tabla 12: Caso de Uso Cargar Archivo Configuración Prefijos.	47
Tabla 13: Caso de Uso Generar Archivo Configuración Prefijos.	47
Tabla 14: Expansión del Caso de Uso Cargar Archivo Configuración Entorno Virtual.....	53
Tabla 15: Expansión del Caso de Uso Generar Archivo de Configuración Entorno Virtual.	54
Tabla 16: Expansión del Caso de Uso Cargar Archivo Entorno Virtual.	55
Tabla 17: Expansión del Caso de Uso Cargar Archivo Configuración global.	56
Tabla 18: Expansión del Caso de Uso Generar Archivo de Configuración global.....	57
Tabla 19: Expansión del Caso de Uso Gestionar Grupo de Elementos.	61
Tabla 20: Expansión del Caso de Uso Modificar Configuración Grupo de Elementos.....	62
Tabla 21: Expansión del Caso de Uso Gestionar Configuración Global.....	63
Tabla 22: Expansión del Caso de Uso Gestionar Prefijos.....	65
Tabla 23: Expansión del Caso de Uso Gestionar Configuración Específica.....	66
Tabla 24: Expansión del Caso de Uso Vista Previa.	67
Tabla 25: Expansión del Caso de Uso Cargar Archivo Configuración Prefijos.....	69
Tabla 26: Expansión del Caso de Uso Generar Archivo Configuración Prefijos.....	70
Tabla 27: Descripción de la Clase de Diseño CPrefix.....	88
Tabla 28: Descripción de la Clase de Diseño CPrefixManager.	90
Tabla 29: Descripción de la Clase de Diseño CElementConfiguration.	91
Tabla 30: Descripción de la Clase de Diseño CGroup.	92
Tabla 31: Descripción de la Clase de Diseño CGroupManager.	94
Tabla 32: Descripción de la Clase de Diseño CElementManager.....	95
Tabla 33: Descripción de la Clase de Diseño CConfigurationEditor.	95
Tabla 34: Descripción de la Clase de Diseño CGraphicManager.....	96
Tabla 35: Descripción de la Clase de Diseño CConfigurationController.....	97
Tabla 36: Descripción de la Clase de Diseño CGlobalConfigurationManager.	98
Tabla 37: Descripción de la Clase de Diseño CGlobalConfiguration.....	98
Tabla 38: Descripción de la Clase de Diseño CXMLFile.....	99
Tabla 39: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Entorno Virtual.....	104
Tabla 40: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Prefijos.	104
Tabla 41: Modelo de Prueba del Caso de Uso Generar Archivo Configuración Global.....	105
Tabla 42: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Entorno Virtual.	106
Tabla 43: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Prefijos.....	108
Tabla 44: Modelo de Prueba del Caso de Uso Cargar Archivo Configuración Global.....	109