# Universidad Nacional Autónoma de Nicaragua-León (UNAN-León)

# Facultad de Ciencias y Tecnologías Departamento de Computación



Entorno virtual en 3D del Centro de Iniciativas Medioambientales de la Ciudad de León aplicando las tecnologías del servidor de aplicaciones 3D OpenSimulator.

Monografía para optar al título de

# Ingeniería en Sistemas de la Información

#### **Autores:**

- **& Br.** Juan José Espinoza Ruiz.
- ❖ Br. Mayerling Antonia Flores Acuña.
- \* Br. Gabriela María Parrales Méndez.

#### **Tutor:**

M.Sc. Eduardo Santiago Molina

León, Nicaragua Septiembre del 2015.

"A La Libertad Por La Universidad"



# **Agradecimiento**

Agradezco en primera instancia a mi padre celestial por irradiar la luz que alumbra el sendero hacia el camino del bien y del éxito camino que me ha llevado a culminar con éxito mis estudios dejando atrás los tropezones y barreras que pudiesen afectar el llegar a la meta.

Agradezco de manera incondicional a mi madre Lucia Andrea Ruiz quien ha estado a mi lado desde que fui concebido en su vientre y que sin importar esfuerzos o sacrificios me ha sabido formar y dirigirme en el trayecto de mi vida.

A mi padre terrenal José Antonio Espinoza López por ser el pilar de mi fortaleza quien a pesar de estar tan largo, al mismo tiempo siempre ha estado muy cerca y al pendiente de dirigirme y guiarme por los buenos pasos.

A mi abuela María de los Ángeles Narváez López quien ha estado velando desde cerca por mi formación y mi buena conducta a lo largo de mi desarrollo estudiantil.

A Nuestro tutor M.Sc. Eduardo Santiago Molina quien nos dedicó su tiempo amistad sabiduría y conocimiento para colaborar en el aprendizaje y la realización de este reto que nos propusimos.

A mi novia Silvia Elena Díaz que me ha dado su compañía en todo este transcurso de mi carrera apoyándome y ayudándome en el acontecer.

En fin quiero agradecer a todas las personas y amistades que de una u otra manera hicieron de nuestro trabajo una realidad la realización de nuestro trabajo.

Juan José Espinoza



# Agradecimiento

Agradezco a Dios primeramente por haberme brindado la salud, brindarme las fuerzas necesarias que me permitieron culminar con éxito mis estudios y la sabiduría para por entender cada uno de los obstáculos que se nos presentaron en este largo camino.

A mis padres Minin del Carmen Acuña García y Erling Flores Portobanco por su amor incondicional, los consejos necesarios para poder seguir adelante y por su apoyo económico que nunca falto.

A M.Sc. Eduardo Santiago Molina por aceptar la tutoría de este trabajo, por el apoyo incondicional durante el transcurso de la misma y por haber compartido con nosotros cada uno de sus conocimientos.

A mis amigos Gabriela Parrales y Juan José Espinoza porque siempre estuvimos juntos sin importar los obstáculos de este trabajo y no darnos por vencido.

# **Mayerling Antonia Flores**



# Agradecimiento

Agradezco en primera instancia a nuestro Señor Dios por haber estado a mi lado en cada una de las etapas que pase en mi carrera por haberme dado fuerzas cuando las necesite en el momento en que mis estudios se tornaban más complejos, El me consoló cuando necesite consuelo y me dio la sabiduría necesaria para vencer cada uno de los obstáculos que se me presentaron.

Gracias a mi mamá Isidra Antonia Méndez y a mi abuela Eusebia Antonia Méndez por su apoyo incondicional a lo largo de mis estudios, por sus consejos los cuales me hacían reflexionar cada vez que sentía que mis sueños flaqueaban, por su amor que siempre me llena cuando me siento vacía y sobre todo por el apoyo económico que nunca falto, que siempre me dieron a pesar de que habían dificultades.

A M.Sc. Eduardo Santiago Molina por haber aceptado ser nuestro tutor, por su apoyo en el transcurso de nuestro trabajo, por el tiempo que nos brindó para aconsejarnos y corregirnos pero principalmente por haber compartido sus conocimientos con nosotros los cuales fueron de gran ayuda.

A mis compañeros de estudios, compañeros de tesis pero sobre todo grandes amigos e incondicionales Mayerling Antonia Flores Acuña y Juan José Espinoza Ruiz los cuales estuvieron siempre a mi lado brindándome su apoyo, su amistad y compartiendo sus conocimientos para que pudiéramos llegar hasta el punto donde estamos hoy, día con día estuvimos luchando para superarnos como grupo teniendo acuerdos y desacuerdo, altas y bajas pero al final siempre llegábamos a la conclusión más satisfactoria y beneficiosa para todos.

A mi novio Juan Luis Midence Pérez porque antes de ser mi novio fue mi amigo, mi mejor amigo, un amigo incondicional que siempre estaba ayudándome cuando más necesitaba ayuda, aconsejándome cada vez que necesitaba un consejo y ahora está a mi lado como más que amigo pero siempre apoyándome en todas las dificultades que se me han presentado en el transcurso de mi trabajo de tesis siendo así mi apoyo, mi fuerza, mi mejor amigo!.

#### Gabriela Maria Parrales



# Contenido

1. Resumen	10
2. Plante amiento del problema	12
3. Justificación del tema	13
4. Producto	14
5. Obje tivos	15
5.1 Objetivo General	15
5.2 Objetivos específicos	15
6. Marco Teórico	16
6.1 DNS Dinámico (DDNS)	16
6.2 Mundos Virtuales	16
6.2.1 Origen de los mundos virtuales en la web	)16
6.2.2 Servicios de OpenSim	16
6.3 Bases de Datos	17
6.4 Avatar	17
6.5 Script	17
6.6 Simulación	17
6.7 OpenSimulator	19
6.7.1 Historia de OpenSim	20
6.7.2 Características	21
6.7.3 Hippo OpenSim Viewer	21
6.7.4 Vivox	21
6.7.5 Grid	22
6.7.6 Archivos IARs	23
6.7.7 Archivos OARs	23
6.8 Archivos de configuración de OpenSimulator	
6.8.1 OpenSim.ini	
6.8.2 Regions.ini	
6.9 Lenguaje de Programación utilizado en OpenSin	



6.9.1 Linden Scripting Language (LSL)	24
6.10 Construcción Básica de los objetos	25
6.10.1 Crear	25
6.10.2 Borrar	25
6.10.3 Mover	25
6.10.4 Girar	25
6.10.5 Cambiar de tamaño	25
6.10.6 Duplicar	25
6.10.7 Vaciar	26
6.10.8 Recortar	26
6.10.9 Retorcer	26
6.10.10 Disminuir la parte superior o inferior	26
6.10.11 Dar flexibilidad	26
6.10.12 Unir varios objetos	26
6.10.13 Aplicar texturas	26
6.10.14 Permisos del objeto	27
6.11 CIMAC León (Centro de Iniciativas Medio Ambientales)	27
6.11.1 Bosque de Sabana	28
6.11.2 Bosque Húmedo	28
6.11.3 Bosque Seco	28
6.11.4 Bosque de Ribera	28
7. Diseño metodológico	28
7.1 Materiales utilizados	28
7.1.1Hardware	28
7.1.2 Software	28
7.2 Etapas del proyecto	29
7.2.1 Etapa I: exploración	29
7.2.2 Etapa II: Instalación de OpenSim 0.8.0.4	30
7.2.3 Etapa III: Instalación de Hippo OpenSim Viewer	30
7.2.4 Etapa IV: Construcción y diseño de objetos que conforman el mundo virtual.	30



7.2.5 Etapa V: Realización de pruebas de conexión local y remota	30
7.2.6 Etapa VI: Redacción del informe final	32
8. Desarrollo	32
8.1 Como crear un script	32
8.2 Creando el primer script	32
8.3 Variables	35
8.3.1Tipos de variables	35
8.3.2 Declaración de variables	35
8.4 Operadores	36
8.4.1 Ope rador cast	36
8.4.2 Evento timer	39
8.4.3 Comunicación entre scripts	40
8.4.4 Eventos más comunes	42
8.4.5 Funciones más comunes	43
8.5 El Chat	43
8.6 Encuesta	43
8.6.1 Descripción de los pasos para realizar la encuesta	43
8.6.3 Análisis de Resultados	44
9. Conclusiones	50
10. Recomendaciones	51
11. Bibliografía	52
12. Anexos	54
12.1 Configuración para permitir el acceso remoto al mundo virtual	54
12.2 Acceso	56
12.3 Instalación de OpenSimulator	61
12.4 Pasos para crear un Grid	63
12.5 Personalización del Avatar	65
12.6 Subir Texturas	70
12.7 Configuración de chat de voz	72
12 & Formato de la enguesta	75



# Índice de Figuras

Figura 1 Composición de LSL	24
Figura 2 Declaración de variables	36
Figura 3 Resultado de la encuesta, pregunta 1	44
Figura 4 Resultado de la encuesta, pregunta 2	45
Figura 5 Resultado de la encuesta, pregunta 3	
Figura 6 Resultado de la encuesta, pregunta 4	46
Figura 7 Resultado de la encuesta, pregunta 5	46
Figura 8 Resultado de la encuesta, pregunta 6.	47
Figura 9 Resultado de la encuesta, pregunta 7.	47
Figura 10 Resultado de la encuesta, pregunta 8	47
Figura 11 Resultado de la encuesta, pregunta 9.	49
Figura 12 Logo	54
Figura 13 Configuración cuenta no-ip	54
Figura 14 Configuración para crear DDNS	56
Figura 15 Configuración del router para crear DDNS	58
Figura 16 Configuración del router para obtener direcciones ip fijas	59
Figura 17 Configuración del router para habilitar los puertos de acceso a OpenSim	60
Figura 18 Instalación de OpenSimulator	61
Figura 19 Creación del Grid	
Figura 20 Acceso al mundo virtual	64
Figura 21 Apariencia del avatar	65
Figura 22 Texturas	71
Figura 23 Configuración VozIp	73



# Índice de Tablas

Tabla 1 Materiales	Hardware	28
Tabla 2 Materiales	Software	29



#### 1. Resumen

El actual proyecto tecnológico que se implementó consiste en recrear el ambiente real mediante un entorno virtual en 3D del Centro de Iniciativas medio ambientales de la Ciudad de León (CIMAC) aplicando la herramienta OpenSim y algunos de sus complementos, uno de los motivos que condujo a elaborar dicho proyecto fue el querer indagar más sobre nuevas tendencias tecnológicas que se están llevando a cabo actualmente en otros países desarrollados en el campo de la informática, los mundos virtuales se han transformados en verdaderas redes sociales útiles para la interacción entre personas de diferentes lugares pueden socializar, aprender, entretenerse, etc. Debido al potencial social de los mundos virtuales se han convertido en un atractivo para instituciones, empresas e investigadores. Esto beneficiará en gran manera al CIMAC.

El uso de entornos virtuales ha crecido considerablemente en los últimos años, por la gran cantidad de posibilidades que presentan. Universidades como Harvard, Cincinnati, empresas como IBM, entre otras, han apostado al uso de entornos virtuales con diferentes propósitos, tales como la educación a distancia, e-learning, realización de conferencias y reuniones a distancia, simulación de entornos para su estudio, etc.

Existen diferentes plataformas para la simulación de entornos virtuales, como por ejemplo SecondLife y OpenWonderland, pero en este caso se utilizará OpenSim, debido a que es una herramienta de código abierto, gratuita, multiplataforma y de licencia libre.

En el documento se presenta la herramienta de creación de entornos virtuales OpenSim, orientado principalmente en su uso en los ambientes educativos, mediante este documento podrán adquirir nuevos conocimientos acerca de la instalación, configuración y manipulación de meta verso en 3D. Esta información aplica la metodología en cascada por lo que se irá mostrando tanto el nombre de las aplicaciones que serán usadas como las respectivas configuraciones que se tienen que realizar para la construcción y el correcto funcionamiento del mismo, de esa manera se dará cuenta como el paso anterior esta intrínsecamente relacionado con el posterior.

Es una aplicación Cliente-Servidor en la que se utiliza un Servidor de aplicaciones 3D llamado OpenSimulator quien es propietario tanto de comandos, como archivos de configuración junto a la herramienta compiladora de gráficos llamada Hippo OpenSim Viewer hacen posible la creación y manipulación de un conjunto de objetos que se adaptan a la necesidad del creador para dar origen a un meta verso, todos estos objetos se almacenan en una base de datos, OpenSimulator es compatible con MySQL pero la base de datos que trae por defecto es SQLite la cual se utilizará por lo que es más liviana y por lo



tanto permite una comunicación más rápida al momento de realizar las consultas necesarias.

Como toda aplicación que su principio sea el de socializar a un grupo de usuarios, OpenSimulator ofrece la posibilidad de establecer la conexiones tanto local como remota, vamos de lo sencillo a lo complejo lo anteriormente mencionado dará paso a usar complementos de chat mediante texto, además VozIp o chat de voz es decir, poder hablar en tiempo real con otros usuarios en general o con algún grupo en particular, configuraciones que estarán descritas posteriormente en este documento.



### 2. Planteamiento del problema

En la actualidad el Centro de Iniciativas Medioambientales de la Ciudad de León no posee registros de una manera digital de cada una de las especies de plantas que se encuentran en los diferentes tipos de bosques con los que cuenta esta institución dentro de sus instalaciones los cuales trata de preservar, esto provoca que la manipulación de los datos no se lleven de una manera organizada y se pierda el control de la misma, aun siendo de gran importancia para llevar a cabo el correcto funcionamiento del centro.

Por este motivo se diseñará un entorno virtual en 3D de una parte de las instalaciones, implementando tecnología adecuada y a bajo costo para crear una biblioteca virtual que posea información de relevante importancia acerca de cada una de las plantas, Haciéndolo de una manera didáctica y poco convencional y así despertar el interés de los usuarios que se conecten al mundo virtual. Debido a que se sumergirán en el recorrido virtual por lo que este es una simulación de la realidad así tendrán mayor interacción entre ellos porque no solo van a adquirir nuevos conocimientos sino que podrán compartir sus ideas comunicándose entre sí.´

Al contar el CIMAC con una biblioteca virtual serán beneficiados los usuarios que estén interesados en el estudio del bosque de Sabana y las plantas que se encuentran en las instalaciones del centro pertenecientes a él.

OpenSimulator interactúa junto a Hippo OpenSim Viewer el cual es el visor que nos muestra de una manera gráfica todas las configuraciones que se realizan en el servidor, este brinda grandes beneficios ya que en él se puede diseñar objetos virtuales similares a los reales, cada uno de estos objetos pueden tener diferentes animaciones dependiendo de la programación con la que cuenten, además posee una base de datos en la cual se almacenará la información que deseemos esto permite tener registros específicos de cada una de la plantas, también cuenta con servicios de comunicación así como chat público, chat privado, y servicio de VozIp o chat por voz ,lo que permitirá a los usuarios expresar sus comentarios, dudas, sugerencias o simplemente una comunicación para socializar de acuerdo a temas que sean de su interés.



#### 3. Justificación del tema

El motivo por el cual se diseñara el entorno virtual en 3D de las instalaciones de CIMAC en OpenSim es que a diferencia de otros programas utilizados para la creación de objetos en tres dimensiones este servidor no solo proporciona herramientas para diseñar objetos sino que asigna un determinado espacio o región para una completa manipulación de estos, además proporciona diferentes herramientas que trae por defecto como es el servicio de chat y otras que se tienen que activar como la VozIp no obstante también brinda la posibilidad de que los objetos tengan diferentes animaciones por medio de programación de Script.

Al implementar el proyecto para promover el desarrollo del CIMAC también se brindará al sitio la ventaja de que por medio del entorno virtual se cuente con la información digitalizada de las diversas especies de árboles que pertenecen al bosque de Sabana, esto ayudará a que el usuario adquiera nuevos conocimientos de la flora nacional y sociabilicen con quien lo deseen.

Existen diferentes programas que permiten la creación de objetos en 3D por ejemplo de ellos están Sketchup, AutoCAD, Unity, Blender etc. Que a pesar de que cada una posee características especiales para la creación de estos; cuentan con ciertas limitaciones siendo una de ellas que son software de paga, otra desventaja sería que a pesar que permite la modelación de objetos estos se convierten en su mayoría en archivos de bodega estáticos ya que permanecen individuales y al momento de querer utilizarlos se tiene que pasar por diferentes procesos.

OpenSim permite configuraciones para que dentro del entorno virtual creado en su visor Hippo OpenSim puedan tener acceso de manera remota diferentes usuarios interesados en experimentar un recorrido virtual.

Una vez realizadas todas las configuraciones necesarias, la plataforma OpenSimulator le brindará al usuario los siguientes servicios:

- ❖ Podrá acceder al entorno virtual en 3D del CIMAC, ya sea fuera o dentro del País.
- ❖ Desplazarse fácilmente por las áreas que tenga autorizadas tales como: Recepción, Auditorio y Bosque de Sabana que estarán diseñados en el entorno virtual de las instalaciones del CIMAC.
- ❖ Tendrá comunicación de Chat con los demás usuarios que estén en el entorno virtual, donde esto permitirá si desea agregar algún otro usuario como amigo y tener una comunicación más personal.



❖ Permitirá también comunicación de VozIp con los avatares que se encuentren dentro del mundo virtual, con el objetivo de que tengan una comunicación más real.

#### 4. Producto

El producto entregable será, un software el cual es un entorno virtual del CIMAC-León, en donde el usuario podrá acceder fácilmente a diversas áreas y disfrutar de las instalaciones virtuales del sitio. El software posee las siguientes características:

- Completo: Dispone de las herramientas Chat, Vivox (VozIp) y el diseño virtual.
- Confidencialidad: Los usuarios podrán conocer el sitio virtual, chatear con sus amigos y de igual forma hablar con ellos sin ningún temor de que otra persona intervenga en sus conversaciones.



# 5. Objetivos

#### 5.1 Objetivo General

➤ Diseñar un entorno virtual en 3D para la construcción de las especies de árboles que se encuentran en el Bosque de Sabana ubicado en las instalaciones del CIMAC-León utilizando las tecnologías del servidor de aplicaciones 3D OpenSimulator.

#### 5.2 Objetivos específicos

- Facilitar a los usuarios adquirir nuevos conocimientos acerca del Bosque de Sabana en Nicaragua y las diferentes especies de árboles que se encuentran en él.
- > Solventar las necesidades educativas de los usuarios en las áreas de ciencias naturales y biología de las plantas para despertar su interés en ellas.
- Sensibilizar a los usuarios sobre el cuido de los bosques por medio del entorno virtual diseñado.
- Adquirir nuevos conocimientos sobre el diseño y modelado de objetos en 3D.



#### 6. Marco Teórico

#### 6.1 DNS Dinámico (DDNS)

El DNS dinámico (DDNS) es un servicio que permite la actualización en tiempo real de la información sobre nombres de dominio situada en un servidor de nombres. El uso más común que se le da es permitir la asignación de un nombre de dominio de Internet a un dispositivo con dirección IP variable (dinámica) (Web, 2014).

#### **6.2 Mundos Virtuales**

Son un campo de la tecnología muy relacionado con la inteligencia artificial ya que consiste en la simulación de mundos o entornos a los cuales se les denomina virtuales, los usuarios tiene la impresión de estar en ese mundo junto con la habilidad de navegar y manipular objetos en él (Victor Lara Arana, 2011).

#### 6.2.1 Origen de los mundos virtuales en la web

El origen de los mundos virtuales en la world wide web (www), se remota al inicio de la década de los 90, con el desarrollo del lenguaje de Modelado de realidad Virtual (VRML) estándar. Las principales características de este es que permitían la definición de objetos 3D en lugares remotos, la definición de animaciones, la inclusión de conexiones a otros mundos o sitios web y la modificación del entorno real para convertirlo en virtual (Victor Lara Arana, 2011).

#### 6.2.2 Servicios de OpenSim

#### 6.2.2.1 Chat

El término chat es un anglicismo que significa charla. Es uno de los métodos de comunicación digital surgida con las nuevas tecnologías. Consiste en la conversación simultánea entre dos o más personas conectadas a la red.

Los mensajes escritos se publican instantáneamente en la pantalla del ordenador. El receptor tiene acceso a ellos sin ningún tipo de retardo y puede contestarlos de igual manera.

Los chats pueden ser públicos o privados. En los chats públicos todos los usuarios conectados a él pueden participar en la conversación. En un chat privado sólo los participantes invitados exclusivamente pueden hablar (Fotonostra).

#### 6.2.2.2 VozIp

VozIp proviene del inglés Voice Over Internet Protocol, que significa "voz sobre un protocolo de internet". Básicamente VozIp es un método por el cual tomando señales de audio analógicas del tipo de las que se escuchan cuando uno habla por teléfono se las



transforma en datos digitales que pueden ser transmitidos a través de internet hacia una dirección IP determinada (VozIp).

#### **6.3 Bases de Datos**

Es el conjunto de informaciones almacenadas en un soporte legible por ordenador y organizadas internamente por registros (formado por todos los campos referidos a una entidad u objeto almacenado) y campos (cada uno de los elementos que componen un registro). Permite recupera cualquier clase de información: referencias, documentos textuales, imágenes, datos estadísticos, etc.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. En este sentido; una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, y por ende se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos (Molina).

#### 6.4 Avatar

En Informática, como avatar se denomina la representación gráfica que, en el ámbito de internet y las nuevas tecnologías de la comunicación, se asocia a un usuario para su identificación en el mundo virtual. Los avatares pueden ser fotografías, dibujos o, incluso, representaciones tridimensionales (Significados).

#### 6.5 Script

Los Scripts son programas, usualmente pequeños o simples, para realizar generalmente tareas muy específicas. Los Scripts son un conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben de ser interpretados línea a línea en tiempo real para su ejecución; esto los distingue de los programas (compilados), pues estos deben de ser convertidos a un archivo binario ejecutable para poder correrlos (ALEGSA).

#### 6.6 Simulación

La simulación es un método por el cual se puede probar, experimentar o ver qué sucede en determinado proceso, escenario u objeto sin el riesgo que haya consecuencias reales. Luego permite comparar diferentes soluciones ante un problema, probarlas y ver cuál es la mejor, posteriormente, se podría aplicar esa solución que funcionó adecuadamente en la simulación, en el mundo real y esperar que las consecuencias que el modelo de simulación muestre, sean las mismas en el mundo real (EtnasSoft).

-

<sup>&</sup>lt;sup>1</sup> **Avatar:** Identidad virtual.



#### Entre las principales ventajas de simular un sistema se encuentran:

- ➤ Una vez construido, el modelo puede ser modificado de manera rápida con el fin de analizar diferentes políticas o escenarios.
- ➤ Generalmente es más barato mejorar el sistema vía simulación, que hacerlo directamente en el sistema real.
- Es mucho más sencillo comprender y visualizar los métodos de simulación que los métodos puramente analíticos.
- Los métodos analíticos se desarrollan casi siempre para sistemas relativamente sencillos donde suele hacerse un gran número de suposiciones o simplificaciones, mientras que con los modelos de simulación es posible analizar sistemas de mayor complejidad o con mayor detalle.
- > En algunos casos, la simulación es el único medio para lograr una solución.
- ➤ No siempre es posible realizar pruebas en el sistema real, ya sea porque es económicamente inviable o porque el riesgo de una falla en la prueba pone en peligro al sistema real y a su entorno, así como a los actores del mismo, por lo que se opta por simular el sistema y realizar pruebas en un ambiente controlado, fuera de peligros.

# Como es de esperar, no sólo hay ventajas en la utilización de la simulación en los sistemas, también existen desventajas como por ejemplo:

- ➤ Los modelos de simulación complejos en una computadora son costosos y requieren mucho tiempo para desarrollarse y validarse.
- ➤ Generalmente se requiere gran cantidad de corridas computacionales para encontrar "soluciones óptimas", lo cual repercute en altos costos.
- > Es difícil aceptar los modelos de simulación.
- Los modelos de simulación no siempre dan soluciones óptimas.

**Script:** Archivo de órdenes, archivo de procesamiento por lotes o guion, programa usualmente simple, que por lo regular se almacena en un archivo de texto plano.



#### 6.7 OpenSimulator

OpenSimulator es un servidor multi-plataforma (OpenSimulator, 2015), de aplicaciones 3D multi-usuario de código abierto. Se puede utilizar para crear un entorno virtual (o el mundo) que se puede acceder a través de una variedad de clientes, en múltiples protocolos. También cuenta con una instalación opcional (el Hypergrid) para permitir a los usuarios visitar otras instalaciones OpenSimulator a través de la web de su instalación OpenSimulator 'casa'. De esta manera, es la base de un metaverso distribuido naciente.

OpenSimulator permite a los desarrolladores de mundos virtuales personalizar sus mundos usando las tecnologías que sienten que funcionan mejor, se ha diseñado el marco para ser fácilmente extensible. OpenSimulator está escrito en C#, funcionando tanto en Windows a través de .Net Framework y en las máquinas Unix sobre el Mono marco. El código fuente se distribuye bajo una licencia comercial amigable para incrustar OpenSimulator en los productos.

Para incursionar en el mundo virtual que se construye con OpenSim, el usuario debe contar con un visor, Viewer o cliente (software) que se instala en la computadora personal. Los visores compatibles de OpenSimulator se basan en el código fuente del navegador de código abierto utilizado para SecondLife.

Para ejecutar correctamente los visores se debe disponer, como mínimo, 1 GB de RAM y una tarjeta gráfica con 256 MB (DDR2 o DDR3) de memoria de vídeo dedicada.

Las Computadoras portátiles de nueva generación incluyen tarjetas de gráficos compatibles. Aquellas equipadas con tarjetas de gráficos integrados tipo Intel, etc. pueden soportar la carga respectiva pero será incomodo utilizar el software con la velocidad de despliegue de las imágenes. Para explotar al máximo los recursos gráficos disponibles se necesita una tarjeta gráfica con al menos 512 MB (DDR3 y DDR4) y 2 GB de RAM.

Durante la utilización de OpenSim necesitamos crear un Grid o una rejilla la cual nos permitirá que múltiples usuarios puedan acceder a nuestro servidor por medio de una cuenta de usuario, de la cual el administrador del mundo tendrá control y podrá dar los privilegios que quiera a los diferentes usuarios entre ellos está permitirle al usuario que pueda diseñar el entorno que el desee para ello el usuario que quiera acceder a nuestro mundo necesita crear un Grid y acceder por medio de la cuenta de usuario que se le sea asignada.

Para conectarse a un mundo virtual basado en OpenSimulator, el usuario deberá crear una cuenta en este. En la mayoría de los casos, el administrador del mundo virtual controla la posible inscripción y, en su caso, el nivel de privilegios que se otorga. Dependiendo de los privilegios, el residente podrá modificar objetos, construir o establecer grupos de amigos.

OpenSimulator posee una página web muy importante que podrá encontrar dentro de los anexos, en ella podemos descargar muchos objetos los cuales posteriormente los exportaremos al Hippo OpenSim Viewer, OpenSim trabaja con objetos los cuales se tiene la opción de descargar, por ejemplo Arboles, Mesas, Sillas, Floreros etc., esta es un pequeña ventaja que logramos obtener. De esta manera no tenemos que estar diseñando uno a uno los objetos más pequeños que se vayan a utilizar para el diseño del entorno virtual.



OpenSim posee diferentes archivos de configuración, dentro de estos tenemos uno llamado ObjectsAssetSet en ella es donde se guardan todos los objetos que se utilizarán o se creará en el mundo virtual, el Hippo OpenSim Viewer solo acepta objetos de tipo de extensión XML o IAR en la página de OpenSim podemos descargar algunos objetos con esa extensión, en el caso de diseñar otros objetos se utilizarán las mismas herramientas que brinda Hippo OpenSim.

#### 6.7.1 Historia de OpenSim

El proyecto OpenSimulator (OpenSimulator, 2015) fue fundada en enero de 2007 por Darren Guardia (también conocido como MW), que, como tantas otras personas, vio el potencial de una fuente abierta 3D servidor de entornos virtuales que podría ser utilizado para muchas aplicaciones diferentes. También al igual que muchos otros, Darren había visto muchos otros intentos de servidores del mundo virtual de código abierto que fallan a menudo debido a la enorme tarea de escribir tanto un servidor y un cliente a la vez.

Luego, en enero de 2007, el cliente de SecondLife fue lanzado como código abierto, y libsl (una biblioteca de código abierto para crear clientes personalizados que podrían conectarse a SecondLife (tm)), estaba llegando al punto de ser estable. Así que la idea de OpenSim nació, con el objetivo inicial de crear una prueba de concepto de servidor que el cliente SL podría conectarse y permitir algunas funciones básicas. La idea era que con el tiempo el alcance del proyecto, se espera pueda llegar a ser mucho más que sus humildes comienzos. Esto ha sucedido, con el objetivo actual de desarrollar una plataforma de entorno virtual estándar que cualquier aplicación puede utilizar como marco. Mientras se sigue manteniendo la compatibilidad con el cliente de SecondLife, se ha estado trabajando para apoyar a varios otros clientes. En el futuro también se espera apoyar protocolos y entornos que son completamente independientes de la de SecondLife.

Fuera de la caja, OpenSimulator se puede utilizar para simular entornos virtuales similares a SecondLife, ya que soporta el núcleo del protocolo de mensajería de SL. Como tales, estos mundos virtuales se pueden acceder con los regulares espectadores SL. Sin embargo, OpenSimulator no aspira a convertirse en un clon de la plataforma de servidores de SecondLife. Más bien, el objetivo del proyecto es permitir que él desarrolle características innovadoras para entornos virtuales y el Meta verso en general.

OpenSimulator es cada vez más estable en el tiempo, pero todavía es un sistema de software complejo que puede sufrir varios errores y peculiaridades; manejar con cuidado!

2

**BSD:** Licencia Comercial amigable.

<sup>&</sup>lt;sup>2</sup> **OpenSimulator:** Servidor de aplicaciones 3D multi-usuario de código abierto.

**Servidor:** Aplicación en ejecución (software) capaz de atender las peticiones de un cliente y devolverle una respuesta en concordancia.



#### 6.7.2 Características

- > Soporta entornos 3D multi-usuario en línea tan pequeñas como 1 simulador o tan grandes como miles de simuladores.
- > Soporta múltiples clientes y tienen acceso al mismo mundo en el mismo tiempo, a través de múltiples protocolos.
- > Soporta scripting incluyendo LSL y C #.
- > Proporciona capacidad ilimitada para personalizar las aplicaciones del mundo virtual a través del uso de módulos.

#### 6.7.3 Hippo OpenSim Viewer

Es un visor de SecondLife modificado (Soft 112, 2015), dirigido a los usuarios de OpenSim, Permite construir hasta una altura de 10,000 metros, Hippo OpenSim es el entorno gráfico el cual presenta OpenSim en el que podremos diseñar nuestro mundo, en este se presenta un avatar el cual se utilizará para realizar cada una de las construcciones teniendo un conocimiento profundo acerca del funcionamiento de esta plataforma, y de esta manera poder manipular las funciones que se necesitan para satisfacer las necesidades.

#### **6.7.4 Vivox**

Una herramienta imprescindible en el uso de entornos virtuales orientado a la educación es el chat de voz, es decir, poder hablar en tiempo real con otros usuarios en general o con algún grupo en particular. Esta característica por defecto no viene configurada en OpenSim, por lo que se tiene que configurar. Para esto se utilizará un servicio de voz que se puede integrar a OpenSim llamado Vivox.

Para poder utilizar el servicio de voz en OpenSim, se ha escogido el servicio Vivox. Vivox es una compañía americana que desde su nacimiento en el 2005 viene trabajando con soluciones de voz sobre IP y que en el 2007 realizó una alianza con SecondLife para proveer el servicio de voz para este mundo virtual (Vivox, 2015).

#### 6.7.4.1 Pasos para poder utilizar el VIVOX

- 1. Para poder hablar en OpenSim mediante la voz, se solicitará el Servicio de voz Mundial de Vivox. Para ello se debe entrar a la siguiente dirección web: http://support.vivox.com/opensim/ y rellenar los datos solicitados.
- 2. Luego de rellenar el formulario, envían un correo electrónico con una dirección web para confirmar la solicitud, después hay que esperar alrededor de un día para que envíen la información que se necesita para conectar OpenSim con la red de voz.
- 3. Se copia la información recibida en el correo, en el archivo OpenSim.ini (Archivo alojado en la capeta: opensim-0.8.0.4 > bin > OpenSim.ini)



- 4. Iniciar OpenSimulator.
- 5. Para hablar, haz clic en el botón "Talk" (Hablar)

Si la voz está deshabilitado en tu mundo, y desea activarla, entonces usted necesita asegurarse en las casillas de verificación:

1. Habilitar voz en Estate.

En el menú, seleccione: World > Region/Estate

#### 6.7.5 Grid

Es el nivel que organiza las regiones y las posiciona en el mundo, maneja cosas que deben de existir en las regiones, como el inventario del usuario una forma de verlo es con el mapa del mundo.

#### 6.7.5.1 Creación del Grid en OpenSimulator

El Grid se utiliza para crear una nueva conexión a tu mundo, una conexión clienteservidor, este funciona de una manera sencilla una vez que el administrador del mundo virtual haya creado una cuenta de usuario y una contraseña para este, dentro de esta misma configuración el administrador tendrá que asignar los permisos que desea que el usuario tenga dentro del mundo, para que el usuario pueda acceder deberá tener el visor de OpenSim instalado en su máquina, una vez instalado se procede a la configuración del Grid o rejilla de red.

La conexión de una computadora a otra funciona a través de la red, esto es, atreves de la dirección IP de la maquina Servidor, en la parte del Quick Grid Select hay dos opciones, la de metrópolis la cual es un mundo virtual, ya existente, y el otro es la conexión a nivel local, el q nos permite conectarnos a nivel local (a nuestro propio mundo).

Para conectar dos computadoras en un mundo que se esté creando, simplemente se modifica la dirección de IP del visor al igual que se hace al configurar el servidor, para ello ir a edit -> Preferences

3

Luego se crea un nuevo Select a Grid y se inserta todas las credenciales iguales a la q se modificó cuando se configuró el servidor.

<sup>&</sup>lt;sup>3</sup> **Hippo OpenSim Viewer:** Visor Hipo de OpenSim.



#### 6.7.6 Archivos IARs

OpenSimulator Inventory Archives (IARs) (OpenSimulator, 2015) es un elemento que permite guardar, trasladar, cargar carpetas y cosas que se tienen en el inventario de las cuentas. Este archivo con formato IAR permitirá guardar plenamente objetos, texturas y scripts que se tengan en el inventario.

#### 6.7.7 Archivos OARs

Los archivos OARs (OpenSimulator, 2015) son regiones enteras que ya han sido diseñadas completamente y se guardan todos esos objetos en un archivo con esa extensión, estos elementos pueden ser guardados, trasladados, cargados y modificados.

### 6.8 Archivos de configuración de OpenSimulator

#### 6.8.1 OpenSim.ini

En este archivo se encuentran muchas de las configuraciones esenciales de OpenSim como son habilitar la base de datos con la que se va a trabajar, las modificaciones del servicio de chat por defecto este trae que los avatar que estén a máximo 20 metros de distancia puedan comunicarse vía chat, los comandos para hacer uso de la Voz IP por medio del servidor de Vivox, comandos para habilitar el acceso remoto para administrar el mundo desde otro lugar, comandos para permitir enviar mensajes vía smtp. En pocas palabras este es uno de los archivos de configuración más importantes de OpenSim.

#### 6.8.2 Regions.ini

Este es el archivo donde están todas las configuraciones que se realizan al momento de instalar OpenSim dentro de los campos que se tienen que introducir correctamente durante la instalación de encuentran los siguientes:

El UUID de la región lo cual es un identificador único que posee la región.

Localización de la región.

El tamaño de la región que máximo es 256 metros en el eje X, Y, Z.

La dirección Ip Interna la cual se deja la Ip publica 0.0.0.0.

El puerto interno con el cual trabaja que es el 9000.

4

<sup>&</sup>lt;sup>4</sup> **Grid:** Rejilla.



El nombre del Host externo por el que se tuvo que crear una cuenta en la página de no-ip la cual brinda un servicio de DNS Dinámico que con sus respectivas configuraciones en el router se encarga de permitir acceso remoto al mundo virtual.

Cualquier modificación que se desee realizar en cualquiera de estos campos solamente se hacen en este archivo pero hay que tener mucho cuidado en lo que se está modificando debido a que son campos muy importantes y cualquier error provocaría que se cayera el mundo.

# 6.9 Lenguaje de Programación utilizado en OpenSimulator

### 6.9.1 Linden Scripting Language (LSL)

Linden Scripting Language (LSL) (Guía de iniciación Linden Scripting, 2015) es un lenguaje de programación orientado a eventos, Similar a C y java. Con este lenguaje de programación se pueden crear scripts para controlar el comportamiento de los objetos en el mundo virtual.

Los scripts están compuesto de uno o varios estados, dentro de cada estado hay uno o varios eventos y dentro de estos, una o varias funciones.



Figura 1 Composición de LSL

5

**OAR'S:** Extensión reconocida por OpenSim para cargar regiones

<sup>&</sup>lt;sup>5</sup> IAR'S: Archivo con extensión reconocida por OpenSim para cargar objetos.



#### 6.10 Construcción Básica de los objetos

#### 6.10.1 Crear

Para crear un objeto pulsa el botón construir en la parte baja de la pantalla (al lado del botón de "Buscar" y "Mini mapa") o bien pulsa el botón derecho en el terreno y seleccionar la opción "Crear" o también puede pulsando "B". Aparecerá una ventana con las formas primitivas (Prims) disponibles, si desplazamos el ratón fuera de esta ventana vemos como el cursor se transforma en una varita mágica, pulse el ratón izquierdo sobre cualquier punto del terreno y aparecerá la forma seleccionada.

#### **6.10.2** Borrar

Para eliminar un objeto pulse el botón derecho del ratón sobre él y seleccione la opción "Más" y después la opción "Borrar", o bien pulse el botón derecho del ratón sobre él y seleccione "Editar" y pulse el botón del teclado "SUPR".

#### Editar:

Para editar un objeto pulse el botón derecho del ratón sobre él y seleccione "Editar". Si ya tenía la ventana de edición abierta basta con hacer clic sobre el objeto.

#### 6.10.3 Mover

Edite el objeto y ponga el ratón sobre la línea verde, azul o roja (dependiendo de hacia dónde quiera mover el objeto), y manteniendo pulsado el botón izquierdo del ratón muévalo.

#### 6.10.4 Girar

Para girar un objeto, edite el objeto y en la ventana de edición seleccione "Girar" o mantenga pulsada la tecla "CTRL". El objeto aparecerá rodeado de tres círculos rojo, verde y azul. Ponga el ratón sobre uno de los tres círculos y muévalo manteniendo pulsado el botón izquierdo del ratón

#### 6.10.5 Cambiar de tamaño

Para cambiar de tamaño un objeto active la opción estirar en la ventana de edición de objetos o mantenga pulsadas las teclas "CTRL." y "SHIFT" de su teclado. En el objeto aparecerán unos cubos, dos rojos, dos verdes, dos azules y ocho blancos. Moviendo el ratón manteniendo pulsado el botón izquierdo en uno de esos cubos vemos que el objeto se estira o se contrae tanto de ese lado como del contrario. Si desea estirar o contraer solo un lado desactive la opción "Estirar para los lados"

#### 6.10.6 Duplicar

Para duplicar un objeto muévalo en cualquier dirección mientras mantiene pulsada la tecla "SHIFT" del teclado, veremos que el objeto se desplaza y en su lugar aparecerá una copia del mismo.



#### **6.10.7 Vaciar**

Para ahuecar un objeto pulse "Mas" en la ventana de edición, pulse en la pestaña

"Objeto" y en el recuadro "Vacía" incremente su valor. Este valor incida el porcentaje de hueco que tiene el objeto, debajo del recuadro "Vacía" hay un desplegable llamado "Forma vacía" con el que puede cambiar la forma del hueco.

#### 6.10.8 Recortar

Se puede recortar los lados de un objeto cambiando los valores de los recuadros "B" y "E" de la opción "Camino de principio a fin" en la pestaña "Objeto".

#### 6.10.9 Retorcer

Para retorcer la parte inferior o superior de un objeto modifique los valores de los recuadros "B" y "E" de la opción "Girar del comienzo al fin" en la pestaña "Objeto".

#### 6.10.10 Disminuir la parte superior o inferior

Se puede disminuir la parte inferior y superior de un objeto cambiando los valores de los recuadros "X" e "Y" de la opción "Disminuir" en la pestaña "Objeto".

#### 6.10.11 Dar flexibilidad

Para hacer un objeto flexible marque la casilla "Camino flexible" en la pestaña "Carácter" de la ventana de edición. Disponemos de varias opciones como, incrementar el

"Viento", para que el objeto ondee, incrementar la "Tensión" para hacer el objeto más rígido, también se puede aumentar su suavidad, su gravedad y la fuerza en cada uno de sus ejes.

#### 6.10.12 Unir varios objetos

Para unir varios objetos en uno solo primero hay que editar uno de ellos y seleccionar los demás pulsando en ellos con el botón izquierdo del ratón mientras mantiene pulsada la tecla "SHIFT" del teclado. Después en el menú superior de la pantalla pulsar "Herramientas" y seleccionar "Unir", o bien pulsar las teclas "CTRL" y "L" del teclado simultáneamente.

Separar objetos: Para separar objetos edite el objeto y en el menú superior de la pantalla pulse "Herramientas" y seleccione "Separar", o bien pulse las teclas "CTRL", "SHIFT" y "L" del teclado simultáneamente.

#### 6.10.13 Aplicar texturas

Para aplicar una textura a un objeto pulse abra la ventana de edición y en la pestaña "Textura" pulse sobre el cuadro "Textura", seleccione cualquier imagen de su inventario y esta aparecerá en todos los lados del objeto. Si desea aplicar una textura a una sola cara del objeto basta con arrastrarla desde su inventario, manteniendo pulsado el botón izquierdo del ratón, a la cara deseada del objeto.



#### 6.10.14 Permisos del objeto

En la pestaña "General" de la ventana de edición tenemos varias opciones para modificar los permisos, las principales son:

- Compartir con grupo: Al marcar esta opción, cualquier miembro del grupo puede usar el objeto como si fuera el dueño.
- ➤ Permitir a todos mover: Esta opción permite a cualquiera otro avatar mover el objeto.
- Permitir a todos copiar: Esta opción permite a cualquiera otro avatar coger una copia del objeto.
- Modificar: Marque esta opción para permitirle al siguiente dueño que haga cambios en el objeto, por ejemplo, cambiar el nombre, texturas o dimensiones del objeto.
- ➤ Copiar: Marque esta opción para permitirle al siguiente dueño hacer copias ilimitadas del objeto.

#### 6.11 CIMAC León (Centro de Iniciativas Medio Ambientales)

Nicaragua es conocida por su rica biodiversidad. En un esfuerzo para educar a la gente acerca de la flora y fauna de Nicaragua, el gobierno y una organización española crearon el Centro de Iniciativa Medio Ambiental (CIMAC) (Leon, municipio, 2015).

El proyecto empezó con la reforestación de aproximadamente una hectárea de la parte sur en la ciudad colonial de León. Tradicionalmente, las partes bajas del pacífico de Nicaragua no son muy conocidas por su biodiversidad. Sin embargo, CIMAC alberga distintos tipos encontrados de bosques que pueden ser en toda Nicaragua. Se construyó un camino que les permite a los visitantes caminar a través de un bosque húmedo, bosque seco, un bosque de ribera y un bosque de sabana. Cuatro tipos de vegetación pueden encontrarse en cada bosque. Acompañados por un guía, los visitantes tendrán la oportunidad de aprender más acerca de estos bosques, ya que el guía le dará una explicación las características de cada de uno.

Dentro del área puede encontrar cedro, mango, papaya y muchos otros tipos de árboles. Así mismo, hay algunos tipos de animales viviendo en el área. Puede observar pájaros como el colibrí, pájaros carpinteros, entre otros tipos de aves. El mejor momento para verlos es por la mañana. Recientemente, un venado fue adoptado por el centro CIMAC, así que también podrá observarlo merodear por el área.

<sup>6</sup> 

<sup>&</sup>lt;sup>6</sup> **Textura:** Imagen aplicada sobre una superficie en gráficos.



#### 6.11.1 Bosque de Sabana

Es una formación intermedia entre un bosque y una pradera.

### 6.11.2 Bosque Húmedo

Es el de mayor extensión en Nicaragua y representa la mayor reserva forestal del país.

#### **6.11.3 Bosque Seco**

Estos bosques resisten una época seca de 6 meses.

### 6.11.4 Bosque de Ribera

Ocupa los suelos de orillas de ríos y lagos.

#### 7. Diseño metodológico

En esta sección se enumeran los diferentes materiales empleados en este trabajo y se describen las etapas a través de las cuales se abordó la investigación.

### 7.1 Materiales utilizados

#### 7.1.1Hardware

Los materiales hardware empleados en el trabajo son:

Material	Descripción	Costo aproximado
1 1	AMD E2-1800 APU with Radeon(tm) HD Graphics	
Laptop SAMSUNG	1.70 GHz , RAM 4GB Intel(R) Core(TM) i3-2370M CPU @ 2.40 GHz , RAM 4GB	
Modem Ubee	Ubee P/N DDW2602 Model NO. U10C037 Input 12 v DC 1.0A	50 dólares

**Tabla 1** Materiales Hardware.

#### 7.1.2 Software

Software	Descripción
----------	-------------



Sistema Operativo Windows 8	Arquitectura x64 es un sistema operativo moderno y con la capacidad de soportar la instalación y configuración de los programas con fines informáticos demostrando la correcta compatibilidad.
OpenSim 0.8.0.4	Es un programa de código abierto que ofrece la opción de servidor web de mundos virtuales en 3D.
Hippo OpenSim Viewer	Es un visor de metaversos dirigida a usuarios de OpenSim. Es quien permite construir objetos e implementar todos los complementos que se ofrecen en OpenSimulator.
Vivox	Proporciona servicios gestionados de comunicaciones en forma de chat de voz integrado, mensajería instantánea (IM) y la presencia de los juegos online, mundos virtuales y otras comunidades en línea.
Dynamic DNS update Client (DUC)	Es quién permite mantener actualizada la dirección ip pública que nos está siendo entregada por nuestro proveedor de internet mediante un nombre de dominio.
Photoshop	Photoshop es el nombre popular de un programa informático de edición de imágenes. Su nombre completo es <b>Adobe Photoshop</b> y está desarrollado por la empresa Adobe Systems Incorporated. Funciona en los sistemas operativos Apple Macintosh y Windows.

Tabla 2 Materiales Software.

# 7.2 Etapas del proyecto

Para abordar y cumplir todos los objetivos, se dividió el trabajo en las siguientes etapas:

# 7.2.1 Etapa I: exploración

En esta etapa se recopiló información acerca del tema, haciendo búsquedas mediante la visita de sitios web.



#### 7.2.2 Etapa II: Instalación de OpenSim 0.8.0.4

En esta etapa se llevó a cabo la instalación y configuración de los distintos ficheros de configuración que trae el software en donde se define las características con las que contará el mundo virtual.

#### 7.2.3 Etapa III: Instalación de Hippo OpenSim Viewer

Se instaló y se puso a prueba el visor, que permite ver de manera gráfica lo que está ocurriendo en la consola del servidor de OpenSim. Para ello se realizaron una serie de subetapas que se describen a continuación:

#### 7.2.3.1 Creación del Grid.

Para la comunicación tanto local como remota se necesitaba la creación de una red privada la cual sería la clave principal para que todos los usuarios se pudieran conectar con el servidor, estando compuesta de una dirección ip más el puerto por el cual sería accedido en este caso el 9000.

#### 7.2.3.2 Configuración de Hippo OpenSim Viewer

Se personalizó las opciones brindadas por el programa de manera de que es lo que realmente se necesitaba y lo que no. Es aquí donde se personaliza si se permite el uso de la voz ip, el uso de mensajes para la comunicación además del color, estilo y tamaño de letras, la calidad de imagen que se puede apreciar a los gráficos, entre otras cosas.

#### 7.2.3.3 Integración de Vivox al mundo virtual

Se integró la tecnología del servidor en línea llamado "Vivox", quien hace posible la comunicación mediante voz ip siempre y cuando se cuente con el acceso a internet.

# 7.2.4 Etapa IV: Construcción y diseño de objetos que conforman el mundo virtual.

Se realizó la construcción de todo el entorno con el que contaría el metaverso implementando los conocimientos anteriormente aprendidos en la etapa de exploración además de la programación de las animaciones con la que se cuentan dentro del mundo virtual a como puede ser por ejemplo el encendido y apagado de una lámpara.

#### 7.2.5 Etapa V: Realización de pruebas de conexión local y remota.

En este momento se realizó la prueba del servidor conectándose con otras máquinas que estaban sirviendo de clientes las antes mencionadas solo necesitaban tener instaladas Hippo OpenSim Viewer.

Primeramente se logró la conexión local ósea solamente se podían conectar personas que se encontraban dentro de una LAN. Luego se logró la conexión remota en donde personas que estaban desde departamentos que no fuese león se podían conectar al mundo virtual y de



esa manera gozar de una comunicación mediante la mensajería en texto o chat por voz, etapa donde se hizo posible compartir el logro de la puesta en marcha del servidor pero antes de esos se tuvo que realizar unas sub-etapas:

#### 7.2.5.1 Configuración del router que brinda la red

Esta etapa es la fundamental de la conexión ya que el router es el corazón de la comunicación es quien da la orientación de la red y quien decide quien entra quien sale y hacia donde se va a dirigir en el complejo tráfico de internet.

Primeramente había que lograr que la máquina que es servidor siempre contara con una ip estática que no cambiara siempre fuese la misma debido a que se perdía la configuración en caso de que cambiase la ip. Seguidamente había que controlar que la ip que nos ofrece el proveedor de internet tampoco cambiara, es donde se aplicó la técnica de usar un nombre de dominio para la ip publica y así no se trabajaría con una ip que en cualquier momento podía cambiar, esto se logró gracias a la página web llamada no-ip junto al resto de configuraciones que se tenían que realizar en el aparato que más adelante serán explicadas en el documento.

#### 7.2.5.2 Capacitación a los usuarios

Al momento de realizar la conexión las personas que fueron seleccionadas para la prueba no sabían el uso acerca del software por tanto se tenía que dar una capacitación mediante tutorial de texto los pasos a seguir para la correcta conexión.

#### 7.2.5.3 Liberación de puertos de comunicación

Se tuvo que liberar ciertos puertos con el cual trabaja el servidor para que aceptara la conexión solicitada desde una máquina que estaba en el exterior a la red.

#### 7.2.5.4 Evaluación de la calidad de llamada

En esta etapa se evaluó la calidad de llamada teniendo en cuenta ciertos parámetros, el tiempo que se lograba mantener activa la llamada la claridad de transferencia de voz, sino se escuchaba ruido o algún retorno al momento de entablar la comunicación. Para ello se les realizó las preguntas a las personas que participaron en la prueba de conexión en red de acuerdo a los parámetros antes mencionados.

#### 7.2.5.5 Evaluación de la calidad de los gráficos

Se les pregunto a las personas que se ofrecieron voluntariamente a la prueba de conexión a nuestro mundo virtual como era la claridad de los gráficos tomando en cuenta la claridad de los objetos, si los miraban completos o solamente se les presentaba una parte además si el avatar aparecía completo al momento de entablar la comunicación y que tan buena era la calidad al momento de desplazarse por el mundo y lo más importante si cerraba el mundo luego de cierto tiempo.



# 7.2.5.6 Validación de los resultados obtenido de la encuesta a los usuarios

Se tomó en cuenta las repuestas que se habían obtenido de cada uno de los usuarios y de esa manera saber cómo estábamos ante los gustos de los demás, luego de eso retocamos los errores y mejoramos las partes en donde se estaba débil.

#### 7.2.6 Etapa VI: Redacción del informe final

Redacción y organización del documento final, el cual contiene la explicación de cada una de las etapas de este proyecto y solución de cada uno de los objetivos planteados.

#### 8. Desarrollo

#### 8.1 Como crear un script

Primero que todo se tiene que crear un objeto. Para ello se pulsa en el botón construir en la parte baja de la pantalla (al lado del botón de "Buscar" y "Mini mapa") o se bien pulsa el botón derecho en el terreno y seleccionar la opción "Crear". Aparecerá una ventana con las formas primitivas (Prims) disponibles, si se desplaza el ratón fuera de esta ventana se ve como el cursor se transforma en una varita mágica, se pulsa con botón izquierdo del ratón sobre cualquier punto del terreno y aparecerá la forma seleccionada.

Los script se asocian a objetos que se crean. Se puede crear un script directamente dentro del objeto, pulsando en "Nuevo script" desde la pestaña "Contenido" de la ventana de edición del objeto. O directamente en el inventario para más tarde arrastrarlo al inventario del objeto en la pestaña "Contenido".

#### 8.2 Creando el primer script

Una vez creado un script, al hacer doble clic en él aparecerá una ventana llamada editor de script desde la cual se podrá modificar el script. Se ve que el script comienza con el estado default, todos los scripts deben tener definido obligatoriamente el default que será el primero que se ejecute, el texto azul corresponde a los estados, el texto rojo a las funciones Se borra el contenido del script y se escribe lo siguiente:

```
default
```



#### } // Cerramos el estado default

#### Este script tiene dentro del estado default dos eventos:

1. touch start (integer total number)

Cuando toquemos el objeto se ejecutaran las funciones del interior de este evento, en ese caso la función es llSay (0, " Me has tocado ");

Con la función IlSay () hacemos que el objeto diga algo por un canal, como vemos la función IlSay () tiene dos parámetros dentro del paréntesis separados por una coma (0, " Me has tocado ") el primero indica el canal y el segundo es el texto que queremos que diga el objeto que debe ir siempre entre comillas.

En este caso al tocar el objeto este nos dirá "Me has tocado" por el canal 0 que es del chat y por lo tanto lo escucharían todos los avatares que estuviesen a menos de 20 metros del objeto.

```
2. collision_start (integer num)
```

Las funciones del interior de este evento se ejecutaran al chocar con un objeto, en este caso al chocarnos el objeto nos dirá "Te has chocado", por el canal 0

Como se ve las funciones siempre tienen que ir dentro de las llaves que abren y cierran un eventos y terminadas en";" y los eventos siempre tienen que ir dentro de las llaves que abren y cierran los estados. Con // podemos escribir anotaciones dentro de los script y que no interfieran en ellos

#### Poner texto sobre un objeto:

```
Para poner un texto sobre un objeto se usa la función IlSetText (), por ejemplo: IlSetText ("Hola", <1.0, 0.0, 0.0>, 1);
```

#### Esta función tiene 3 parámetros separados por comas:

- 1. El primero es el texto el cual queremos poner encima del objeto que debe ir siempre entre comillas.
- 2. El segundo es un vector que nos indica el color del texto. Una forma de representar el color en la informática es el RGB (red, green, blue). La intensidad de un color se tendría que definir con un número que va del 0.0 al 1.0, por ejemplo:

```
<1.0, 0.0, 0.0> = rojo
<0.0, 1.0, 0.0> = verde
<0.0, 0.0, 0.0> = blanco
<1.0, 1.0, 0.0> = amarillo
```

3. El tercer parámetro indica la transparencia del texto siendo 0 completamente transparente y 1 completamente opaco.

```
default
```

```
{ // Abrimos default state_entry () //este es el evento state_entry () { //abrimos evento
```



state\_entry () es un evento que se ejecuta al ejecutarse el estado al que pertenece, como al ejecutarse el script lo primero que se ejecuta es el estado default, lo primero que se ejecutara serán las funciones que haya dentro del state\_entry (). En este caso llSetText ("Hola",

```
<1.0, 0.0, 0.0>, 1);
```

#### **Estados:**

Además del estado default se pueden tener más estados que se definen escribiendo "state" seguido del nombre del estado. Por ejemplo:

```
default
{ // Abrimos default
       state_entry ()
               IlSetText ("Estoy en el estado default", <1.0, 1.0, 1.0>, 1);
       touch_start (integer total_number)
               state dos; // con ente comando hacemos que cambie al
                              //estado "dos"
        }
} //cerramos default
state dos
{ // Abrimos el estado dos
       state_entry ()
               IlSetText ("Estoy en el estado dos", <1.0, 1.0, 1.0>, 1);
       touch_start (integer total_number)
               state default; // con ente comando hacemos que cambie al
                              //estado "default"
} //cerramos el estado dos
```

Al ejecutar este script se ejecutará primero el estado default, por lo tanto se ejecutara primero la función llSetText ("Estoy en el estado default", <1.0, 1.0, 1.0>, 1); por encontrarse dentro del state\_entry () del estado default. Al ejecutarse esta función aparecerá el texto "Estoy en el estado default" encima del objeto.

Cuando toquemos el objeto se ejecutara el comando state dos; el cual hará que se ejecute el estado dos y por lo tanto encima del objeto pasara a poner "Estoy en el estado dos" porque



se ejecutará la función llSetText ("Estoy en el estado default",<1.0, 1.0, 1.0>, 1); por encontrarse dentro del state\_entry () del estado dos.

Si volvemos a tocar el objeto volverá a cambiar de estado y de texto sucesivamente.

#### 8.3 Variables

Una variable es un identificador donde se puede almacenar la información en un script. Una variable tiene un nombre, un tipo y un valor. El nombre debe empezar por una letra y las normas para nombrar una variable son similares a C o Java. Se diferencian mayúsculas de minúsculas.

Hay dos tipos de variables según la posición que ocupan, las variables locales que están definidas dentro de un estado y las variables globales que están definidas donde empieza el script. Las variables tienen que empezar por una letra y es recomendable que tengan un nombre descriptivo por si algún día quieres modificar ese script sea más fácil cambiar cosas.

#### 8.3.1Tipos de variables

Los tipos de variables dependen de lo que estas puedan almacenar:

- ✓ Integer: Sirve para almacenar números enteros, como por ejemplo: 2.
- ✓ Float: Sirve para almacenar números reales, como por ejemplo: 2,35.
- ✓ String: Almacena texto, por ejemplo: "Hola".
- ✓ key: Es un string especial usado para identificar algo propio de Vleaf, como por
- ✓ ejemplo: un avatar un objeto o una textura.
- ✓ Vector: Es una variable que está compuesta por tres números de tipo float, como por ejemplo: <1.0,1.0,1.0>
- ✓ Rotation: es una variable compuesta por 4 números de tipo float, que se usará para dar movimiento a los objetos en el eje Z.

#### 8.3.2 Declaración de variables

Para declarar una variable hay que escribir primero el tipo y luego el nombre, por ejemplo:

String miVariable

La variable miVariable es del tipo string lo que significa que en ella se puede almacenar cualquier texto.



```
string myVariable;
default
{
touch_start (integer total_number)
{
myVariable="Me has tocado"; //Almacenamos dentro de
//myVariable el valor Me has
tocado
IlSay (0, myVariable); //al poner myVariable es igual que si
//pusiésemos lo que lleva almacenada, es
//decir "Me has tocado";
}
collision_start (integer num)
{
myVariable="Me has tocado"; //Cambiamos el contenido de
//myVariable por Te has chocado
IlSay (0, myVariable);
}
}
```

Figura 2 Declaración de variables

#### 8.4 Operadores

LSL, como el resto de lenguajes, tiene unos operadores que es necesario tener en cuenta:

```
    ✓ X = Y a X le asignamos el valor de Y
    ✓ X == Y verdadero si el valor de X es igual al Y
    ✓ X != Y verdadero si X es diferente a Y
    ✓ X < Y verdadero si X es menor que Y</li>
```

 $\checkmark$  X > Y verdadero si X es mayor que Y

 $\checkmark$  X <= Y verdadero si X es menor o igual que Y

 $\checkmark$  X >= Y verdadero si X es mayor o igual que Y

 $\checkmark$  + mas(suma)

✓ - menos(resta)

✓ \* multiplicación

✓ / división

✓ ++ incremento de 1 en 1

✓ -- descrecimiento de 1 en 1

✓ % resto de división

✓ && and(y)

**✓** || or (o)

#### 8.4.1 Operador cast

integer numero2=2;

El operador cast sirve para transformar un tipo de variable en otra, por ejemplo: Con la función llSay (0, myVariable) la variable mi variable solo puede ser del tipo string si queremos que el objeto diga el valor de otra variable que no fuese string lo haríamos de la siguiente forma: llSay (0, (string) myVariable) integer numero1=3;



```
default
       touch_start (integer total_number)
               integer suma; //al declarar la variable dentro del evento solo la
                             //podemos utilizar dentro de este
               suma= numero1+ numero2;
               IlSay (0, (string) suma);
Cambiar el color la posición y la rotación a un objeto:
Cambiar el color:
Podemos cambiar el color de un objeto con la función IlSetColor (), esta función tiene dos
parámetros, el primero es un vector que indica el color y el segundo un integer que indica a
que cara del objeto se le cambia el color, por ejemplo:
       llSetColor (<1.0, 0.0, 0.0>,1);
Para cambiar de color todas las caras hay que escribir: IISetColor (<1.0, 0.0, 0.0>,
ALL SIDES);
default
{
       touch_start (integer total_number)
          vector rojo=<1.0, 0.0, 0.0>;
          IlSetColor (rojo, ALL_SIDES);
          state azul;
}
state azul
       touch_start (integer total_number)
               vector azu=<0.0, 0.0, 1.0>;
               IlSetColor (azul, ALL_SIDES);
               state default;
Cambiar la posición:
Para cambiar la posición de un objeto se usa la función llSetPos (), por ejemplo si queremos
poner un objeto en la posición x=100, y=120.5, z=50 escribiríamos llSetPos (<100, 120.5,
50>)
Si queremos por ejemplo mover un metro hacia arriba un objeto y no sabemos en qué
posición se encuentra necesitamos la función llGetPos () que nos dice en qué posición se
encuentra el objeto en ese momento
default
```



```
touch_start (integer total_number)
               IlSay (0, "Estoy en la posición"+( string)llGetPos ());
               llSetPos (llGetPos ()+<0, 0, 1>);
               IlSay (0, "Y ahora en la posición"+(string)llGetPos ());
Cambiar la rotación:
Con la función llSetRot (rotacion); podemos rotar un objeto, para ello el parámetro rotación
debe ser del tipo rotation, por ejemplo para aumentar la rotación Y de un objeto 90 grados
al cada vez que lo tocamos se haría de la siguiente forma.
default
{
   touch_start (integer total_number)
       vector xyz=<0.0, 90,0.0>; //vector que indica los grados que deseamos girar y sobre
       que eje
       vector xyz_en_radianes=xyz*DEG_TO_RAD ://Lo cambiamos a
               //radianes multiplicándolo por DEG_TO_RAD
               rotation rotacion= ||Euler2Rot(xyz_en_radianes); //convertimos el
              //vector en rotation con la función llEuler2Rot.
               IlSetRot(llGetRot()*rotacion);
}
If/else:
La forma general de esta sentencia es:
       if (expresion)
               Acciones 1
       else
              Acciones 2
Si expresión es verdadera, entonces se ejecuta las acciones 1; en caso contrario, se ejecuta
las acciones 2.
Ejemplo:
default
       touch_start (integer total_number)
               integer a=2;
               integer b=5;
```



Da lugar a un bucle en el que se ejecutan unas acciones cierto número de veces. Este número de veces se determina entre los paréntesis mediante tres términos.

- > INICIALIZACION: Se asigna el valor numérico a un "contador".
- ➤ CONDICION: Se comprueba que la variable de inicialización cumple una condición. Si la cumple se sale del bucle.
- ➤ INCREMENTO o DECREMENTO: cada vez que realiza el bucle incrementará o decrecerá según se lo indiquemos el valor de la variable de inicialización

### Un ejemplo:

El primer término del for inicializa la variable contador con inicial que va a tener en el bucle. En este caso 10. En el segundo término se comprueba si se cumple la condición para salirse del bucle si esta se cumple. Y en el tercero en este caso resta uno a contador y comienza el proceso de nuevo. Estos incrementos y decrementos también los puedes escribir de las siguientes formas: contador - - es lo mismo que contador = contador-1.contador = contador +3 incrementaría de 3 en 3.

#### 8.4.2 Evento timer

Con la función llSetTimerEvent (x) se ejecutará las acciones del interior del evento timer cada x segundos, por ejemplo:



```
timer ()
{
    float x= IIFrand(1); //Con IIFrand(1) almacenamos en "x", "y"
    float y= IIFrand(1); // y "z" un numero aleatorio de tipo float
    float z= IIFrand(1); // entre 0.0 y 1.0
    IISetColor (<x, y, z>, ALL_SIDES);
}
```

Con este script conseguimos que cada 0.5 segundos el objeto cambie a un color aleatorio.

## 8.4.3 Comunicación entre scripts

Los objetos en Vleaf se comunican entre sí en casi exactamente de la misma manera que los avatares. Todo lo dicho utiliza un canal específico, los avatares hablan normalmente en el canal cero. Si un objeto escucha en el canal cero, escuchará lo que los usuarios cerca de él digan, del mismo modo, si un objeto habla por el canal cero los avatares cerca de él le escucharán.

Con la función llListen() se le indica al objeto a través de los cuatro parámetros que posee, lo que se quiere que escuche.

Con el primer parámetro se le indica el canal por el cual va a escuchar.

Con el segundo le indicamos el nombre del objeto al cual va a escuchar, si lo dejamos en blanco escuchara a cualquier objeto.

Con el tercer se le indica la key del avatar al cual va a escuchar, si lo se deja en blanco escuchará a cualquier avatar.

Con el cuarto parámetro se le dice que solo escuche un determinado mensaje, si lo deja en blanco escuchará cualquier mensaje.

#### Por ejemplo:

Con IlListen(0, "objeto1",""); le indica que escuche por el canal 0 y solo lo que digan los objetos llamados objeto1.

Con IlListen(30, "",""); le indica que escuche todo lo que se diga por el canal 30.

Una vez que el objeto está escuchando siempre que escuche algo se ejecutará las acciones que hayan dentro del evento listen.

```
listen( integer channel, string name, key id, string message )
{
    acciones ;
}
```

En las variables channel, name, id y message se almacenara el canal, el nombre del objeto, la id del avatar y el mensaje que se ha escuchado respectivamente.

#### Interruptor:

default



```
{
       touch_start (integer i)
               IlSay(1, "Encender");
               state apagado;
}
state apagado
       touch_start (integer i)
               llSay(1, "Apagar");
               state default;
       }
}
Bombilla:
default
       state_entry ()
               !!Listen(1, "","","");
       listen (integer channel, string name, key id, string message )
               if (message=="Encender")
                      IlSetPrimitiveParams([ PRIM_FULLBRIGHT,
                             ALL_SIDES, TRUE ]); //Esta función sirve para dar
                      //"Brillo máximo".
               else
                      IlSetPrimitiveParams([PRIM_FULLBRIGHT,ALL_SIDES,
               FALSE]); //Sirve para quitar "Brillo máximo".
       }
```

Si se introduce el script bombilla en un objeto y el script interruptor en otro, al tocar el objeto con el script interruptor el objeto con el script bombilla se apaga y se enciende sucesivamente.

### Menús:

Se pueden crear menús con la función IlDialog(key avatar, string mensaje, [list botones], integer canal);



Esta función muestra un cuadro de diálogo de color azul en la esquina superior derecha de la pantalla con un mensaje y botones de opción, así como un botón de ignorar. integer canal=1000;

#### 8.4.4 Eventos más comunes

El lenguaje de programación de script es un lenguaje orientado a eventos entre ellos están los siguientes:

```
Se ejecuta durante el contacto de algún avatar con el objeto.
```

```
couch end:
```

Se ejecuta cuando algún avatar deja de tocar al objeto.

```
touch_start (integer number) {}
```

Se ejecuta cuando algún avatar comienza a tocar el objeto.

collision:

Se ejecuta cuando algún avatar colisiona con el objeto.

collision end:

Se ejecuta cuando algún avatar deja de colisionar con el objeto.

```
collision _start (integer num) {}
```

Se ejecuta cuando algún avatar comienza a colisiona con el objeto.

```
state_entry () {}
```

Se ejecuta al comienzo del estado.

State exit:

Se ejecuta cuando se deja el estado actual.

land collision:

```
land_collision (vector position)
```

Se ejecuta cuando el objeto está colisionando físicamente con el terreno.

```
Land collision end:
```



Se ejecuta cuando el objeto deja de colisionar físicamente con el terreno. Land collision start:

Se ejecuta cuando el objeto comienza a colisionar físicamente con el terreno. Listen:

listen( integer channel, string name, key id, string message )

Escucha un determinado canal y hace algo cuando llega un mensaje. Se ejecuta cuando se deja el estado actual.

#### 8.4.5 Funciones más comunes

```
IlSay(0, "");
IlSetStatus(1, TRUE);
IlSetColor(<>, ALL_SIDES);
IlListen(integer n, string m, key, string mensaje);
IlSetText("", <>, 1);
IlSetPos();
IlGetPos();
```

#### 8.5 El Chat

Hay dos formas de comunicarse con los demás miembros que se encuentran dentro del mundo virtual, de manera general (que todos visualicen nuestro mensaje) y de manera privada (dirigido específicamente a otro usuario).

- 1. Para transmitir un mensaje general, damos click en la pestaña "chat" que se encuentra en la parte inferior.
- 2. Se escribe un mensaje y luego click en "say" o "shout" y el mensaje aparece visible para todos los que estén cerca de nuestro avatar.
- 3. Si se quiere enviar un mensaje privado, click en View > Local Chat.

#### 8.6 Encuesta

#### 8.6.1 Descripción de los pasos para realizar la encuesta

El día miércoles 19 de agosto del año 2015 a las 9 am, se fue a las instalaciones del CIMAC León con el objetivo de realizar una encuesta la cual se había elaborado anteriormente para conocer las diferentes opiniones con respecto al trabajo de Tesis, el cual consiste en el diseño en 3D de dicha instalación utilizando las tecnologías del Servidor web OpenSim. En el momento en el cual se llegó, se procedió a organizar los materiales que se utilizarían entre los cuales están:



- ➤ Computador Toshiba con procesador AMD E2-1800 APU with Radeon(tm) HD Graphics 1.70 GHZ, 4 GB de RAM, Sistema Operativo de 64Bits Windows 8.
- ➤ Computador SAMSUNG con procesador Intel(R) Core(tm) i3-2370M 2.40 GHZ, 4GB de RAM, Sistema Operativo de 64Bits Windows 7 Ultimate.
- ➤ Modem Cisco
- ➤ 2 cables de red punto a punto.

A continuación se comenzó encuestando a la primera persona quien fue uno de los trabajadores del CIMAC, antes de realizarle las preguntas se dio unas breves instrucciones de como manipular el mundo virtual, luego él personalmente exploro las instalaciones diseñadas del CIMAC y así fue sucesivamente con las demás personas encuestadas de las cuales obtuvimos opiniones positivas.

### 8.6.3 Análisis de Resultados

En total realizamos 20 encuestas y a continuación mostramos los resultados obtenidos

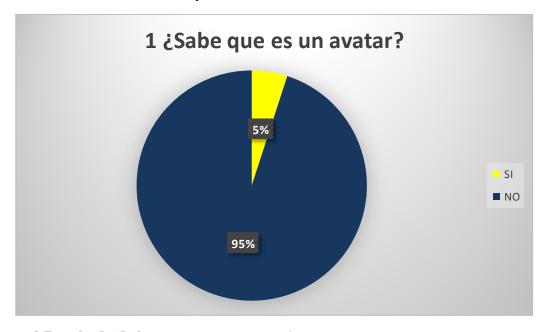


Figura 3 Resultado de la encuesta, pregunta 1



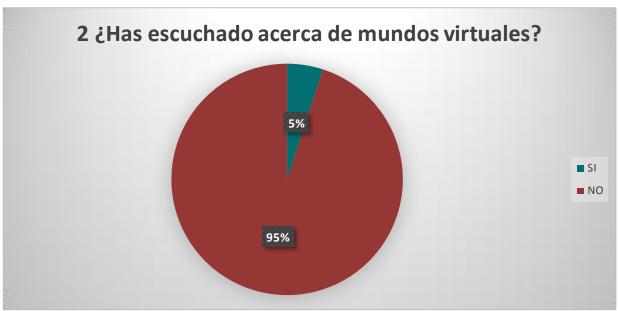


Figura 4 Resultado de la encuesta, pregunta 2

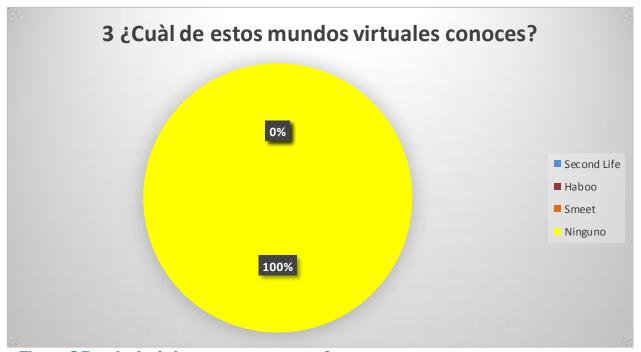


Figura 5 Resultado de la encuesta, pregunta 3





Figura 6 Resultado de la encuesta, pregunta 4

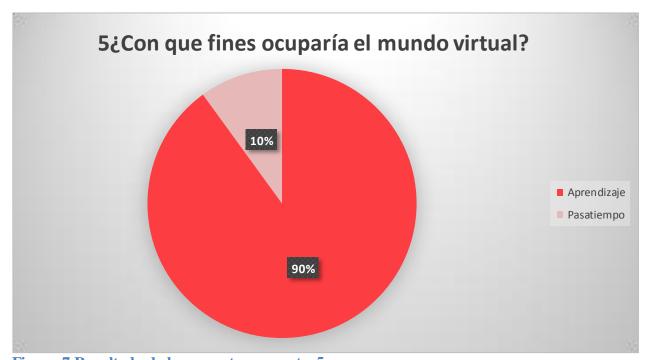


Figura 7 Resultado de la encuesta, pregunta 5





Figura 8 Resultado de la encuesta, pregunta 6



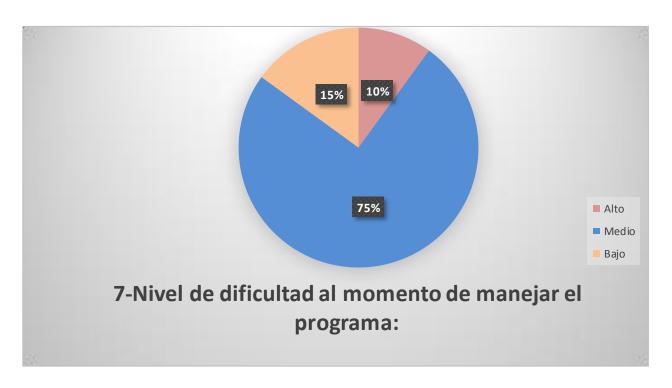


Figura 9 Resultado de la encuesta, pregunta 7

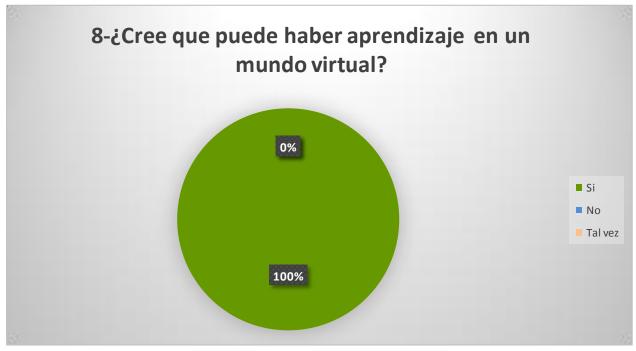


Figura 10 Resultado de la encuesta, pregunta 8





Figura 11 Resultado de la encuesta, pregunta 9



#### 9. Conclusiones

En el presente trabajo, el principal objetivo planteado fue cumplido a cabalidad ya que se logró modelar en objetos 3D cada una de las diferentes especies de árboles del Bosque de Sabana, gracias a las herramientas que nos brinda OpenSim, entre ellas una base de datos en la cual se almacenan automáticamente todos los objetos que son construidos en el entorno virtual logrando de esta manera tener un inventario virtual de las plantas no solo digitalizado sino que construido de una forma muy similar al mundo real.

Se realizó una encuesta por medio de la cual se logró concluir con partes muy importantes del trabajo, una de ellas era la de verificar que los usuarios adquirieran conocimientos sobre las plantas del Bosque de Sabana y esto se cumplió debido a que las personas encuestadas recorrieron las instalaciones diseñadas virtualmente e indagaron sobre información de las plantas que se encuentran ahí.

En total fueron 20 personas a las que se encuestaron, entre ellas estudiantes de secundaria los cuales mostraron mucho interés ya que podían obtener información en las áreas de biología y ciencias naturales de las plantas pertenecientes al Bosque de Sabana por lo que OpenSim posee las herramienta de VozIp y Chat o mensajes en tiempo real y por medio de estas se pueden realizar consultas a la persona encargada del manejo del entorno virtual ya que esta podrá resolver las interrogantes que se les presenten en el trascurso de su recorrido.

Mediante el desarrollo de esta investigación acerca de mundos virtuales se adquirieron conocimientos relevantes de la importancia que tienen en la actualidad, ya que por medio del entorno que se diseñó se logró sensibilizar a los usuarios acerca del cuido que necesita la flora en Nicaragua, debido a que estos son como un puente para la interacción entre ellos y estos a su vez aportan sus diferentes opiniones logrando de esta manera concientizarse entre sí.

Durante el diseño Virtual del CIMAC, no solo se logró adquirir conocimiento sobre entornos virtuales sino que también sobre el diseño y modelado de objetos 3D por lo que cada objeto diseñado en el entorno virtual posee ciertas características que se tienen que tener en cuenta en el momento de diseñarlos además cada objeto brinda una opción con la cual pueden tener animaciones y es, que permite que se puedan crear scripts y enlazarlos a ellos con lo que hace posible que haya mayor interacción en el sitio virtual.

Esta experiencia de investigación dentro de un mundo virtual, diseñada para responder a las interrogantes propuestas en la investigación, permite saber que el diseño interactivo aporta al mejoramiento de las competencias en los estudiantes del área tecnológica y a dar nuevas soluciones a los problemas que enfrenta la sociedad del conocimiento.



#### 10. Recomendaciones

Dentro de nuestro proyecto de tesis siempre se desea que haya una mejora continua del mismo; por lo tanto se recomienda a las personas que tengan interés en el proyecto del diseño virtual del CIMAC concientizar sobre las ventajas que este presenta para que poco a poco se vaya implementando en la educación futura utilizando estos medio virtuales con la finalidad de que la educación sea más accesible a sus usuarios.

Esperamos que en el futuro hayan personas interesadas en el desarrollo de mundos virtuales, si este es el caso también recomendamos que tengan un interés muy específico en la investigación ya que el desarrollo de estas aplicaciones requieren de un trabajo interdisciplinario, puesto que es necesario contar con un equipo que tenga conocimientos en 3 áreas: Programación, Diseño y Redes de Computadoras de esta manera podrán reunir la información necesaria para especializarse en esta campo de la tecnología, ya que estamos seguros que dentro de algunos años se desarrollara más.



#### 11. Bibliografía

- Guía de iniciación Linden Scripting. (06 de 05 de 2015). Obtenido de http://aida.ii.uam.es/virtuam/wp-content/uploads/Descargas/Tutoriales/guia\_sobre\_linden\_scripting\_language.pdf
- Leon, municipio. (10 de 05 de 2015). Obtenido de http://www.leonmunicipio.com/cimac.html
- OpenSimulator. (02 de 05 de 2015). Obtenido de http://opensimulator.org/wiki/History
- *OpenSimulator*. (06 de 05 de 2015). Obtenido de http://opensimulator.org/wiki/Inventory\_Archives
- Soft 112. (02 de 05 de 2015). Obtenido de http://hippo-opensim-viewer.soft112.com/
- Vivox. (04 de 05 de 2015). Obtenido de http://support.vivox.com/opensim/, http://serviciosgate.upm.es/laboratoriosvirtuales/es/wiki/voz\_opensim
- ALEGSA. (s.f.). *Diccionario de Informatica y Tecnologia*. Recuperado el 25 de 04 de 2015, de http://www.alegsa.com.ar/Dic/script.php
- Arráiz, E. (s.f.). Recuperado el 10 de 05 de 2015, de http://ldc.usb.ve/~mcuriel/Cursos/sop3/gridEmely.pdf
- EtnasSoft. (s.f.). *OpenLibra*. Recuperado el 24 de 04 de 2015, de http://www.etnassoft.com/biblioteca/simulacion-conceptos-y-programacion/
- Fotonostra. (s.f.). *fotonostra*. Recuperado el 21 de abril de 2015, de http://www.fotonostra.com/digital/chats.htm
- Molina, M. P. (s.f.). *Bases de Datos*. Recuperado el 22 de abril de 2015, de http://www.mariapinto.es/e-coms/bases\_datos.htm#bd1
- OpenSimulator. (02 de 05 de 2015). *OpenSimulator*. Obtenido de http://opensimulator.org/wiki/Main\_Page
- Pillou, J. (s.f.). *CCM*. Recuperado el 20 de 04 de 2015, de http://es.ccm.net/contents/252-el-concepto-de-red
- Quevedo, M. A. (s.f.). *uServers*. Recuperado el 20 de 04 de 2015, de http://web.uservers.net/ayuda/soluciones/dominios/que-es-una-direccion-ip\_NTk.html
- Significados. (s.f.). *Significados*. Recuperado el 26 de 04 de 2015, de http://www.significados.com/avatar/
- TANENBAUM, A. S. (2003). Redes de Computadoras. Mexico: PEARSON EDUCACION.
- Victor Lara Arana, E. M. (domingo 4 de diciembre de 2011). *Mundos Virtuales*. Recuperado el 21 de abril de 2015, de http://vik91110-mundosvirtuales.blogspot.com/
- VozIp, T. (s.f.). *VozIp*. Recuperado el 22 de abril de 2015, de http://www.telefoniavozip.com/voip/que-es-la-telefonia-ip.htm



Web, D. (24 de diciembre de 2014). *Desarrolladores Web*. Recuperado el 21 de abril de 2015, de http://www.desarrolloweb.com/articulos/dns-dinamico.html



#### 12. Anexos



Figura 12 Logo

# 12.1 Configuración para permitir el acceso remoto al mundo virtual

Primeramente se tiene que registrar en la página de no ip la cual brinda un servicio de DNS dinámico o DDNS https://www.noip.com/sign-up

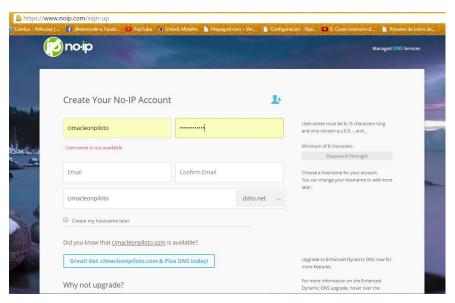


Figura 13 Configuración cuenta no-ip



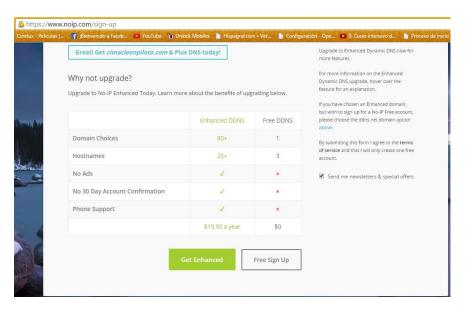


Figura 13. 1 Configuración cuenta no-ip

En este punto existen dos maneras de hacer una nueva cuenta la primera es de un forma pagada, esta es sin vencimiento y posee una versión más completa pero en este caso se crea una cuenta libre de paga que es válida por 1 año, luego de haber creado la cuenta envían un mensaje de confirmación al correo electrónico que se ingresó al momento de crear la cuenta con la dirección URL a la que se registrará, Una vez creada la cuenta se procede a ingresar con el usuario y la contraseña.

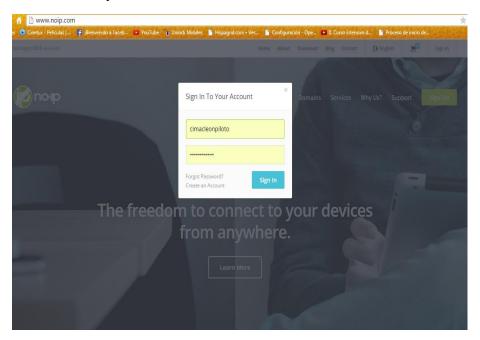


Figura 13. 2 Configuración cuenta no-ip



### 12.2 Acceso



Figura 14 Configuración para crear DDNS

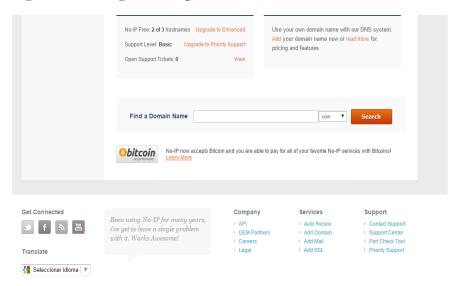


Figura 14. 1 Configuración para crear DDNS

Una vez que se ha ingresado aparecerá un icono de agregar un nuevo Host damos click ahí y así se podrá crear un host.



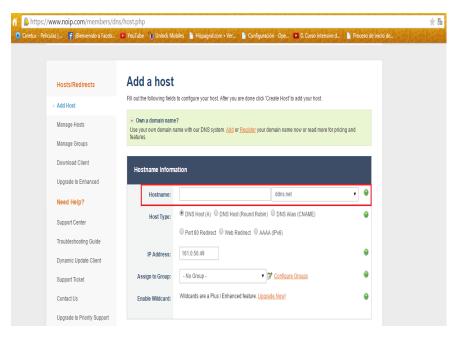


Figura 14. 2 Configuración para crear DDNS

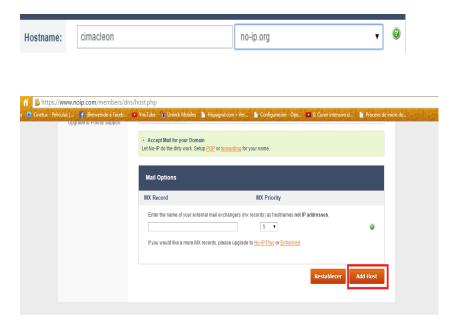


Figura 14. 3 Configuración para crear DDNS

Una vez rellenado los datos necesarios para crear tu nuevo host se da click en add host, luego una vez ya creado se procede a realizar las siguientes configuraciones en el router para hacer uso de no-ip



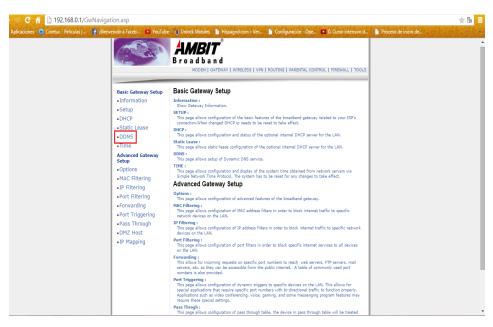


Figura 15 Configuración del router para crear DDNS

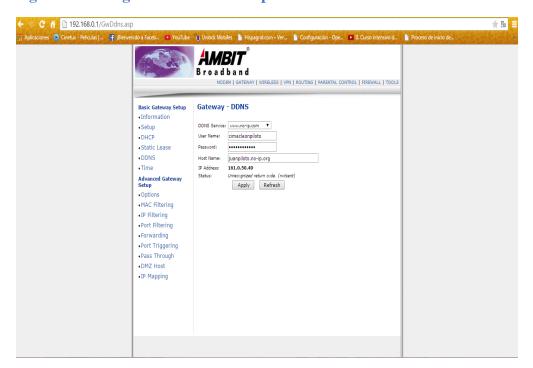


Figura 15. 1 Configuración del router para crear DDNS

Estos campos se llenan con el usuario y la contraseña que se tiene en la página de no ip

El host name se rellena con el nombre de host que se ha creado en la cuenta y cómo se puede observar en el ipaddress aparece una dirección ip, esa es la dirección ip publica que posee el router actualmente, esta se estará actualizando mensualmente pero el servidor no



se verá afectado por ello por lo que al instalarlo se utilizará el nombre del host el cual dará la ip publica que posea actualmente el router.

Luego en Static Lease se procede a poner una ip fija a la maquina en la que está instalada el servidor debido a que si esta se ve modificada por cualquier motivo entre ellos que el router brinda ip por DHCP se presentarían algunos problemas al momento de redirigir el lugar en donde se quiere que los usuarios puedan acceder de manera remota. Para poder realizar esto lo que se hace es poner la Mac de las maquina en la que se quiere la ip fija y la ip que se desea que tenga, estas configuraciones son opcional.

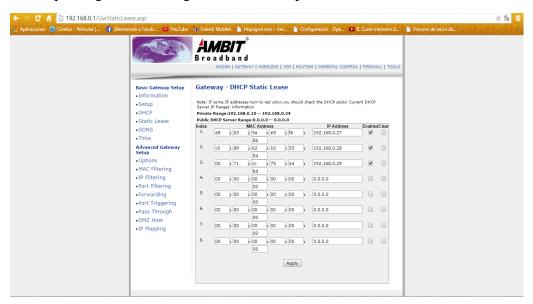
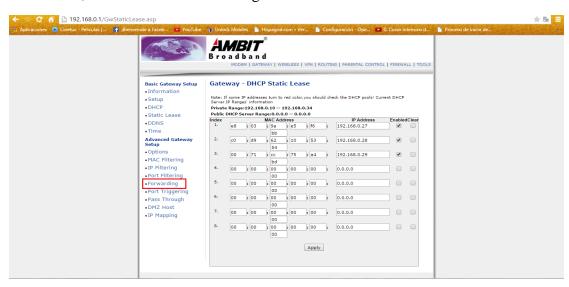


Figura 16 Configuración del router para obtener direcciones ip fijas

Una vez realizado esto se procede realizar las configuraciones para habilitar los puertos que se necesitan, esto sería en el forwarding.





### Figura 16. 1 Configuración del router para obtener direcciones Ip Fijas

En el forwarding se habilitan los puerto que son necesarios para el correcto funcionamiento del Servidor Web 3D OpenSim

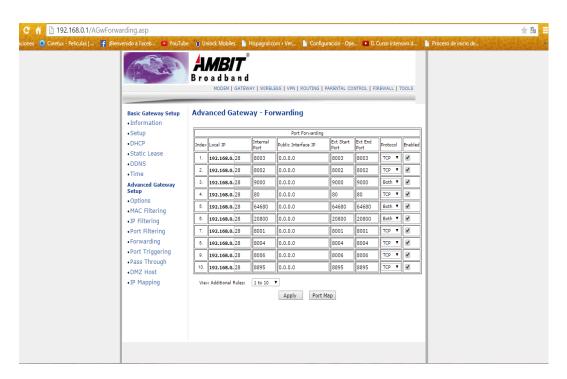


Figura 17 Configuración del router para habilitar los puertos de acceso a OpenSim

Puerto 8003 TCP: Servicios de Activos - Regiones y otros servicios de la red hablan a esta.

Puerto 8002 TCP: Usuario, Iniciar sesión, Servicios de servidor. Clientes, regiones, y otros servicios de la red hablan a esta.

Puerto 9000 TCP/UDP: Puerto predeterminado para el primer simulador. Cuadrícula, regiones y clientes hablan con ellos.

Puerto 80 TCP: puerto HTTP del servidor web.

Puerto 8001 TCP: Grid Servidor - Regiones y otros servicios de la red hablan a esta.

Puerto 8004 TCP: Inventario de Servicios - Regiones y otros servicios de la red hablan a esta.

Puerto 8006 TCP: Enviando mensajes al servidor.

Puerto 8895 TCP: Se utiliza en las primeras versiones de las comunicaciones de región a región.



#### 12.3 Instalación de OpenSimulator

Figura 18 Instalación de OpenSimulator

Primeramente al momento de ejecutar el instalador de OpenSimulator pide el nombre de la región que asigna OpenSim, luego UUID que es el identificador único que tendrá la región este se deja por defecto para que el asigne ese identificador pero se podría modificar con el que se quiera solamente debe de tener el mismo formato y no ser igual a ningún otro posteriormente solicita la localización de la región la cual se deja por defecto, A continuación solicita la dirección Ip Interna en la cual se deja la Ip publica 0.0.0.0 que trae por defecto, seguidamente solicita el puerto interno de OpenSim en este caso es el puerto 9000 que trae por defecto, pregunta si se desea un puerto alternativo por el cual se pueda acceder a OpenSim y esta respuesta se deja la que tiene por defecto que es falsa y por ultimo solicita el nombre del Host externo por el cual los usuarios puedan acceder de manera remota, este ha sido creado anteriormente con la ayuda de la página de no-ip que es juanpiloto.no-ip.org.



```
D:\Mayer\Quinto Año Segundo Semestre\Seminario\SemiMono\Monografia\opensim-0.8.0.1\ope...

13:46:26 - [GRID INFO SERUICE]: Grid info service initialized with 4 keys
13:46:26 - [LOCAL GRID USER SERUICE CONNECTOR]: Enabled local grid user for region CIMAC
13:46:26 - [LOCAL USER ACCOUNT SERUICE CONNECTOR]: Enabled local user accounts for region CIMAC
13:46:26 - [LOCAL ASSET SERUICES CONNECTOR]: Enabled connector for region CIMAC
13:46:26 - [LOCAL ASSET SERUICES CONNECTOR]: Enabled asset caching for region CIMAC
13:46:26 - [NEIGHBOUR CONNECTOR]: Local neighbour connector enabled for region CIMAC
13:46:27 - [WATCHDOG]: Started tracking thread InventoryWorkerThread0, ID 38
13:46:27 - [WATCHDOG]: Started tracking thread InventoryWorkerThread1, ID 39
13:46:27 - [AUTHORIZATION CONNECTOR]: Enabled local authorization for region CIMAC
13:46:27 - [MODULE COMMANDS]: Script engine found, module active
13:46:27 - [MODULE COMMANDS]: Script engine found, module active
13:46:27 - [IEMP ATIACHS]: Registered script functions
Estate cimac has no owner set.
Estate owner last name [User]: Leon
Password:
Estate owner last name [User]: Leon
Password:
Estate ID [a3d5ab8b-6318-4ed9-a37b-e9ea2159aa90]:
13:46:46 - [AUTHENTICATION DB]: Set password for principalID a3d5ab8b-6318-4ed9-
```

Figura 18 1 Configuración de datos de usuario

```
D:\Mayer\Quinto Año Segundo Semestre\Seminario\SemiMono\Monografia\opensim-0.8.0.1\ope...

13:46:52 - IWATCHDOG1: Started tracking thread Maintenance (CIMAC), ID 47
Currently selected region is CIMAC
PLEASE WAII FOR LOGINS TO BE EMBLED ON REGIONS ONCE SCRIPTS HAVE STARTED. Non-script portion of startup took 7m 7s.
INITIALIZATION COMPLETE FOR CIMAC - LOGINS ENABLED
13:47:10 - ISCEME COMMUNICATION SERVICE1: Informing 0 neighbours that region CIMAC
is up
13:47:10 - IMAP IMAGE SERVICE MODULE1: Upload maptile for CIMAC
13:47:10 - ITEXTURED MAPTILE RENDERER1: Generating Maptile Step 1: Terrain
13:47:10 - ITEXTURED MAPTILE RENDERER1: Fetched texture b8d3965a-ad78-bf43-699b-bff8eca6c975, found: True
13:47:10 - ITEXTURED MAPTILE RENDERER1: Fetched texture abb783e6-3e93-26c0-248a-
247666855da3, found: True
13:47:10 - ITEXTURED MAPTILE RENDERER1: Fetched texture 179cdabd-398a-9b6b-1391-
4dc333ba321f, found: True
13:47:10 - ITEXTURED MAPTILE RENDERER1: Fetched texture beb169c7-11ea-fff2-efe5-
0f24dc881df2, found: True
13:47:10 - ITEXTURED MAPTILE RENDERER1: Generating Maptile Step 1: Done in 437
ms
13:47:10 - IMAPTILE1: Generating Maptile Step 2: Object Volume Profile
13:47:10 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
13:47:14 - IREGION DB SQLLITE1: Storing terrain revision r 11
13:47:21 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
13:47:14 - IREGION DB SQLLITE1: Storing terrain revision r 11
13:47:21 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
13:47:14 - IREGION DB SQLLITE1: Storing terrain revision r 11
13:47:21 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
13:47:14 - IREGION DB SQLLITE1: Storing terrain revision r 11
13:47:21 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
13:47:14 - IREGION DB SQLLITE1: Storing terrain revision r 11
13:47:21 - IMAPTILE1: Generating Maptile Step 2: Done in 0 ms
```

Figura 18 2 Finalización de la instalación



## 12.4 Pasos para crear un Grid

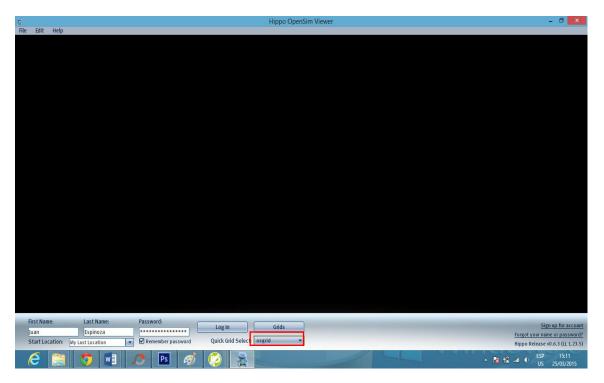


Figura 19 Creación del Grid

Una vez se haya abierto el programa Hippo OpenSim Viewer en la ventana principal se da clic sobre el botón "Grid".





# Figura 19 1 Creación del Grid

Se abrirá otra ventana en la que se dará clic sobre el botón "add".

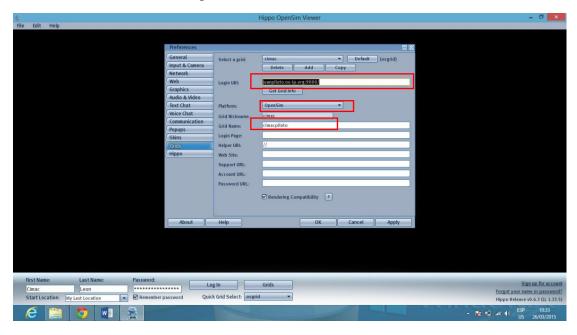


Figura 19 2 Creación del Grid

Rellenar las cajas de textos con los siguientes datos por ejemplo:

Login uri: juanpiloto.no-ip.org:9000

Plataform: OpenSim Grid NickName: cimac Grid Name: cimacpiloto

Una vez terminado el relleno de los campos, clic en "OK".



Figura 20 Acceso al mundo virtual



Una vez que el Grid se creó se agregará en Quick Grid Select el nombre del Grid que se ha creado.

Luego pasamos a loguearnos con un respectivo nombre de usuario y contraseña.

# 12.5 Personalización del Avatar

1. Ejecutar Hippo Viewer



Figura 20 1 Acceso al mundo virtual

2. Haz clic derecho sobre tu Avatar y selecciona el menú Apparence (Apariencia).

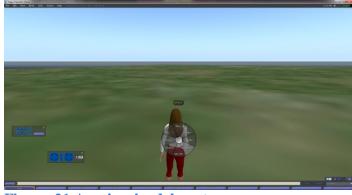


Figura 21 Apariencia del avatar

3. Modifica todos los parámetros.





Figura 21 1 Apariencia del avatar

\*Para cambiar tamaño y forma del cuerpo.



Figura 21 2 Apariencia del avatar

\*Para cambiar el tamaño de la cabeza.



Figura 21 3 Apariencia del avatar

<sup>\*</sup>Para cambiar el tamaño y forma de los ojos.





Figura 21 4 Apariencia del avatar

\*Para cambiar la forma y tamaño de las orejas.



Figura 21 5 Apariencia del avatar

\*Para cambiar el tamaño y la forma de la nariz.



Figura 21 6 Apariencia del avatar



\*Para cambiar el tamaño y la forma de la boca.



Figura 21 7 Apariencia del avatar

\*Para cambiar el tamaño y la forma de la barbilla.



Figura 21 8 Apariencia del avatar

\*Para cambiar la forma del torso.



Figura 21 9 Apariencia del avatar



Para cambiar la forma y el tamaño de las piernas



Figura 21 10 Apariencia del avatar

\*Para cambiar el estilo y el tamaño del cabello con la opción **Hair** - **Style** ahí se elige el estilo de cabello del avatar.



Figura 21 11 Apariencia del avatar

\*Para cambiar el color del cabello se va a la opción **Hair** - **Color** se elige el color de cabello del avatar.





Figura 21 12 Apariencia del avatar

\*Para cambiar la a forma y estilo de la ceja se utiliza la opción **Hair- Eyebrows** ahí se elige la forma de la ceja con la que avatar se presentara.



Figura 21 13 Apariencia del avatar

# 12.6 Subir Texturas

1. Click en File > Upload > image (free)...





Figura 22 Texturas

2. Aparecerá un cuadro de dialogo, buscar la ubicación de la textura (imagen en formato jpg o png) y abrir.

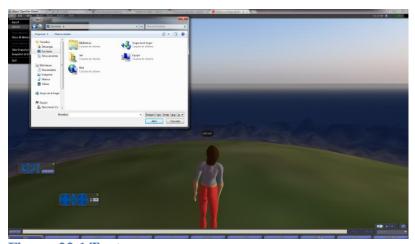


Figura 22 1 Texturas

3. aparece otro cuadro de dialogo en donde se puede modificar el nombre de la textura y descripción. Dar click en Unpload (free).





Figura 22 2 Texturas

4. Y listo, ya se tiene la textura cargada. Para confirmar buscar la textura dando click en inventario.

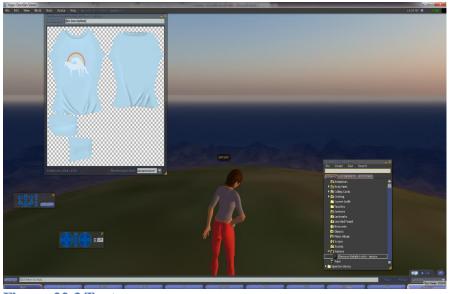


Figura 22 3 Texturas

# 12.7 Configuración de chat de voz

Una herramienta imprescindible en el uso de entornos virtuales es el chat de voz, es decir, poder hablar en tiempo real con otros usuarios en general o con algún grupo en particular. Esta característica por defecto no viene configurada en OpenSim, por lo que tendremos que hacerlo nosotros.

Para esto utilizaremos un servicio de voz que se puede integrar a OpenSim llamado Vivox. Lo que tenemos que hacer es solicitar el servicio completando el siguiente formulario:



# http://support.vivox.com/opensim/

Luego de esto llegará por mail los detalles de configuración, que a grandes rasgos, consiste en copiar/pegar un código al final del archivo OpenSim.ini.

Este es un código de ejemplo para configurar Vivox que se inserta en el Archivo alojado en la capeta: opensim-0.8 > bin > OpenSim.ini)

### [VivoxVoice]

enabled = true

; vivox voice server

vivox\_server = www.osp.vivox.com

; vivox SIP URI

vivox\_sip\_uri = osp.vivox.com

; vivox admin user name

vivox\_admin\_user = mi-user-admin

; vivox admin password

vivox\_admin\_password = mi-pass

Luego de esto, al reiniciar el servidor de OpenSim se iniciará el servicio de voz.

Iniciar OpenSimulator.

Para hablar, haz clic en el botón "Talk" (Hablar)



Figura 23 Configuración VozIp

Si la voz está deshabilitado en tu mundo, pero que desea activar, entonces usted necesita asegurarse de las casillas de verificación:

1. Habilitar voz en él Estate.



# En el menú, seleccione: World > Region/Estate



Figura 23 1 Configuración VozIp



### 12.8 Formato de la encuesta

Reacción de la población ante la creación de un mundo virtual en 3D para el centro de iniciativas medio ambientales de la ciudad de León.

Estamos realizando esta encuesta con el objetivo de conocer su pensar acerca de nuestro trabajo monográfico que lleva como tema creación de entornos virtuales en 3D, esperamos nos pueda ayudar brindándonos su valiosa información.

Sexo: OF	<b>8</b> M	Edad:
1-¿Sabe que es un avatar?		
Sí O I	No O	
2-¿Has escuchado acerca de mundos virtuales?		
Sí O 1	No O	
3-¿Cuál de estos mundos virtuales conoce:		
Second Life	O Haboo O Smeet O	Ninguno O
4-¿Cómo te parece nuestra idea del mundo virtual?		
Buena O	regular o mala o	muy buena 🔾
5-¿Con que fines ocuparía el mundo virtual?		
Aprendizaje	o pasatiempo o	
6-¿Conoce de personas que trabajen con mundos virtuales?		
Sí O I	No O	
7-Nivel de dificultad al momento de manejar el programa:		
Alto O	medio <b>O</b> bajo	0
8-¿Cree que puede haber aprendizaje en un mundo virtual?		
Si O I	No Tal vez	
9-¿Formaría parte del mundo virtual CIMAC León mediante un usuario?		
Si O 1	No O Tal vez O	