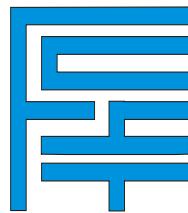


UNIVERSIDAD MAYOR DE SAN SIMÓN
FACULTAD DE CIENCIAS Y TECNOLOGÍA
CARRERA DE LIC. EN INGENIERÍA INFORMÁTICA



**APLICACIÓN INFORMÁTICA EDUCATIVA DE
ESTRUCTURAS DE DATOS Y REDES DE
COMPUTADORAS EN LA UNIVERSIDAD MAYOR DE SAN
SIMÓN, PARA EXPLORAR
TECNOLOGIAS WEB 3D.**

Proyecto de Grado, Presentado Para Optar al Diploma Académico
de Licenciatura en Ingeniería Informática.

Presentado por: ROJAS CHOQUE IVO NABIL

Tutor: Lic. Montoya Burgos Yony Richard.

COCHABAMBA – BOLIVIA

Noviembre, 2013

DEDICATORIA.

Todo el esfuerzo para realizar este proyecto de grado, fue motivado principalmente mis padres, y amigos, a quienes aprecio infinitamente, por su preocupación y permanecer al pendiente.

AGRADECIMIENTO.

La conclusión de este Trabajo final de Grado ha supuesto un gran esfuerzo, todo un proceso de Investigación y cooperación de varias personas, que me ayudaron de una y otra forma para su Desarrollo. En estas líneas únicamente podré incluir una parte de todos los agradecimientos que me gustaría dar.

Gracias a mis padres y hermanos por brindarme todo el amor, confianza y apoyo durante todos estos años difíciles de mis estudios donde he podido comprobar el cariño y la amistad de muchas personas.

A la Carrera de Lic. En Ingeniería Informática de la “Universidad Mayor de San Simón” por la formación recibida y mi docente guía, Lic. Yony Richard Montoya Burgos.

Gracias a todos mis amigos, compañeros y a todas aquellas personas que ayudaron de alguna u otra forma para que este trabajo se haga realidad. Su preocupación me ha ayudado mucho a conseguir un grado más de perseverancia y ha sido un gran apoyo.

FICHA RESUMEN.

El presente Proyecto de Grado, está dirigido principalmente a mostrar la utilidad del área de programación gráfica en tres dimensiones (3D), aplicado a un caso particular en la educación.

La idea consiste en presentar gráficamente los conceptos o temas, para ser mejor abstraídos y asimilados por los estudiantes, también otorgarles un alto grado de Interactividad, capturando de esta manera su atención.

Para el campo de aplicación, en el sector de educación, se eligieron dos materias de la Universidad Mayor de San Simón:

- Elementos de programación y Estructuras de datos.
- Redes de computadoras.

Por su alto grado de dificultad de aprendizaje, se pretende apoyar en su comprensión con esta Aplicación Web en determinados temas.

¿Porque en la web?, porque se ha convertido en el medio de acceso generalizado, independiente de sistema operativo y/o hardware.

Esto hace que los contenidos pueden ser consumidos tanto en equipos de escritorio y laptops, como también sobre plataformas móviles como Tabletas y Teléfonos Inteligentes, lo que lo hace más accesible al alcance de los usuarios, entre quienes se hace más popular estos dispositivos.

La experiencia de todo este aprendizaje hace de este proyecto escalable para ser aplicado a más contenidos en una gran variedad de carreras en la universidad.

INDICE GENERAL.

DEDICATORIA	3
AGRADECIMIENTO	4
FICHA RESUMEN.....	5
INDICE GENERAL.....	7
INDICE DE FIGURAS	9
INDICE DE CUADROS	11
CAPÍTULO 1: INTRODUCCIÓN.....	12
1. Introducción	12
1.1. Muestra de aplicación.....	12
1.2. Esperanza	12
1.3. Descripción de la Problemática.....	12
1.4. Objetivos.	13
1.4.1. Analizar Materias.	13
1.4.2. Producir Introducción.....	13
1.4.3. Seleccionar Tecnologías.	13
1.4.4. Animaciones.....	13
1.4.5. Interactividad.	13
CAPÍTULO II: MARCO REFERENCIAL	14
2. Marco Referencial.....	14
2.1. WebGL.....	14
2.1.1. Experiencia Personal con WebGL.....	14
2.1.1. Soporte de WebGL	15
2.1.1.1. Navegadores.	16
2.1.1.2. Hardware.....	17
2.2. JavaScript.....	17
2.2.1. Three.js.....	18
2.2.2. Tween.js.	19
2.2.3. Jquery.....	20
2.2.4. JqueryMobile.....	20
2.3. Educación	20
2.3.1. Ventajas uso Tic en el Ámbito Educativo.	21
2.3.2. Desventajas uso Tic en el Ámbito Educativo.....	21
2.3.1. Ventajas uso Tic para el Docente.	21
2.3.1. Desventajas uso Tic para el Docente.....	22
2.3.1. Ventajas uso Tic para los estudiantes.	22
CAPITULO 3: ÁREA DE APLICACIÓN.....	23
3. Área de Aplicación.....	23
3.1. Análisis del contenido educativo.....	23
3.1.1. Estructuras de datos.....	23
3.1.1.1. Listas.....	24
3.1.1.2. Pilas.	25
3.1.1.3. Árboles.	26
3.1.1.3.1. Operaciones con Árboles.	26
3.1.2. Redes de Computadoras.	27
3.1.2.1. Transporte de paquetes.....	28
3.1.2.1.1. Paquete de Red.	28
3.1.2.2. Establecimiento de Conexión TCP.....	29
3.1.2.3. Redirección de paquetes.....	30

3.1.2.3.1. Firewall	30
3.1.2.3.2. Proxy.....	30
CAPÍTULO 4: DESARROLLO DEL PROYECTO	32
4. Desarrollo del Proyecto.	32
4.1. Proceso basado en la Filosofía Scrum.....	32
4.2. Roles del equipo.	32
4.2.1. Propietario del producto.	32
4.2.2. Scrum Master.	32
4.2.3. Equipo Desarrolladores.	32
4.3. Usuarios del producto.	32
4.3.1. Estudiante.....	32
4.3.1.1. Rapidez de respuesta.	32
4.3.1.2. Compatibilidad.	32
4.3.1.3. Colores.....	33
4.3.1.4. Mas Gráficos menos Texto.....	33
4.3.1.5. Movimiento e Interactividad.....	33
4.3.2. Docente.	33
4.4. Product Backlog.....	34
4.5. Release Backlog.	35
4.5.1. Estimación de Tiempos.....	35
4.5.2. Tareas y selección de características deseables.....	37
4.6. Definición y Desarrollo de Sprints.	39
4.7. Sprint 1.....	41
4.7.1. Burndown Chart - Sprint 1.....	42
4.7.2. Buscar una tecnología de alta compatibilidad.	42
4.7.3. Desarrollar para el Navegador Web.	45
4.7.4. Desarrollar del lado del Servidor.	45
4.7.5. Desarrollar la vista principal.	46
4.7.6. Eventos para click en el Entorno.	46
4.7.7. Desarrollar las vistas Dev (pruebas).	46
4.7.8. Crear un objeto en el Espacio.....	48
4.7.9. Mover la Cámara en entorno al origen (0,0,0) con el Mouse	50
4.7.10. Evento que arrastra un objeto sobre el espacio (Drag).....	50
4.7.11. Implementar Responsive Design en Template.....	53
4.7.12. Desarrollo del lado del Cliente.	53
4.7.13. Crear Imágenes Representativas.....	54
4.7.14. Definir la forma y colores del tema de la aplicación.	55
4.7.15. Crear una Línea.....	56
4.7.16. Eventos con el mouse para Línea.	57
4.7.17. Vincular una línea a un Objeto.	57
4.7.18. Ordenar Código en forma de Orientado a Objetos.	59
4.8. Sprint 2.....	61
4.8.1. Burndown Chart - Sprint 2.....	61
4.8.2. Implementar Responsive Design en panel 3D.....	61
4.8.3. Optimizar Experiencia de Usuario.	62
4.8.4. Desarrollar las vistas Estructuras de Datos.	63
4.8.5. Desarrollar vista Plantillas: listas pilas árboles.	63
4.8.6. Eventos para los botones de la interfaz interactiva.	64
4.8.7. Touch del mouse para dispositivos móviles.	66
4.8.8. Vincular los 2 extremos de línea a 2 Objetos.	66

4.8.9. Motor de Animación entre Tween.js y Director.js	68
4.8.9.1. Director.js	68
4.8.9.1. Tween.js	69
4.8.10. Eventos disparadores de animaciones.....	70
4.8.11. Encadenar Animaciones una después de Otra.....	71
4.8.12. Usar Recursividad y callbacks en JavaScript.....	71
4.8.13. Crear animación para listas	72
4.8.14. Crear animación para Pilas.....	73
4.8.15. Crear animación para Arboles	¡Error! Marcador no definido.
4.9. Sprint 3	76
4.9.1. Burndown Chart - Sprint 3.....	76
4.9.2. Desarrollar las vistas Redes	76
4.9.3. Desarrollar vista Plantillas: sección de Redes	77
4.9.4. Crear un modelo 3D de una PC	78
4.9.5. Crear un modelo 3D de una Laptop	79
4.9.6. Crear un modelo 3D Router	79
4.9.7. Importar Objetos de 3DSMax mediante la API Three.js	80
4.9.8. Desarrollar animaciones para Establecimiento de C.....	81
4.9.9. Desarrollar animaciones para Transporte de P.....	¡Error! Marcador no definido.
4.9.10. Desarrollar animaciones para Redirección de P.....	¡Error! Marcador no definido.
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.....	82
5. Conclusiones y recomendaciones	82
5.1. Conclusiones.....	82
5.2. Recomendaciones	82
Bibliografía	83

INDICE DE FIGURAS.

Figura 1: significado de WebGL y Navegadores que lo soportan. Fuente: Interactive 3D Graphics Course With Three - VIDEO	14
Figura 2 Captura del Equipo del Navegador que soporta WebGL Fuente: http://get.webgl.org	17
Figura 3. Representación del papel que cumple Three.js Fuente: Interactive 3D Graphics Course With Three - VIDEO	18
Figura 4. Diferentes Animaciones con Tween.js. Fuente: Ejemplos del Paquete Tween.js	20
Figura 5: Estructura de datos, Elaboración propia	23
Figura 6: Figura de Una Pila en Estructura de Datos. Fuente: Elaboración Propia	25
Figura 7Figura de Una Árbol para Estructura de Datos. Fuente: Elaboración Propia	26
Figura 8: Redes de Computadoras, Elaboración Propia.....	27
Figura 9: Transporte de Paquetes. Fuente Creación Propia	28
Figura 10. Establecimiento de Conexión TCP - Fuente Creación Propia	29
Figura 11Redirección de Paquetes - Elaboración Propia	30
Figura 12 Servidor proxy conectando indirectamente dos ordenadores. – Fuente: http://commons.wikimedia.org/wiki/File:Schematic_Proxy_Server.svg	31
Figura 13Renderizado Interactivo Fuente: Curso En Línea 3D Udacity.....	44
Figura 14 Cámara y renderizado Fuente: Curso En Línea 3D Udacity.....	44
Figura 15 Puntos y Vectores Fuente: Curso En Línea 3D Udacity	44
Figura 16 Disposición Sistema de Coordenadas Notación Mano Derecha Fuente: Curso En Línea 3D Udacity	45

Figura 17 Codeigniter Logo.....	45
Figura 18. Vista Principal de la Aplicación- Elaboración Propia.....	46
Figura 19 link para la sección Dev(Pruebas) - Elaboración Propia.....	47
Figura 20 Portada Sección Dev (Pruebas) – Elaboración Propia	48
Figura 21 – Código de Ejemplo Básico Three.js – Elaboración Propia.....	49
Figura 22 Ejemplo de Cuadrado Girando - Three.js - Elaboración Propia	49
Figura 23secuencia de movimiento de la cámara - Elaboración Propia.....	50
Figura 24 Uso de TrackballControls.js - Elaboración Propia.....	50
Figura 25 intersección entre un haz que parte de la cámara y varios objetos - Elaboración Propia.	51
Figura 26 al presionar click - Elaboración Propia.....	51
Figura 27 Transformación del coordenadas 2D a 3D - Elaboración Propia.....	52
Figura 28. Diferentes tamaños de la misma portada comparadas lado a lado - Elaboración Propia.	53
Figura 29 Flujo de datos a Tráves del sistema - Fuete Manual Codeigniter.....	54
Figura 30 Logo JqueryMobile	54
Figura 31 Ilustraciones Vectoriales SVG - Elaboración Propia.....	55
Figura 32 Proceso de creación de las imágenes SVG para el Proyecto - Elaboración Propia.....	55
Figura 33 Colores del tema principal. - Elaboración Propia.	56
Figura 34 Crear una línea en base a 2 puntos - Elaboración Propia.....	56
Figura 35 Función que crea una línea - Elaboración Propia.	57
Figura 36 Objeto con un extremo de una línea asociado a el - Elaboración Propia.....	57
Figura 37 Lina con uno de sus extremos vinculado a un Objeto - Elaboración Propia.	58
Figura 38 Organización del código - Elaboración Propia.	60
Figura 39Funcion del Evento de Redimensión de la Ventana – Elaboración Propia.....	62
Figura 40 llamada a Evento de Redimensión de la Ventana – Elaboración Propia.	62
Figura 41Llamadas a las distintas vistas desde CodeIgniter - Elaboración Propia.	63
Figura 42 Captura de La sección de Estructuras de datos. - Elaboración Propia	64
Figura 43Interfaz Gráfica dat.gui - fuente https://code.google.com/p/dat-gui/	64
Figura 44 Esquema de los controles de la aplicación - Elaboración Propia.....	65
Figura 45 Captura que muestra los Controles de la Aplicación Web - Elaboración Propia.....	65
Figura 46 Problemas de vincular 2 objetos a través de una línea - Elaboración Propia.	66
Figura 47 Esquema de vinculación entre línea y objetos - Elaboración Propia.....	67
Figura 48 Código para Vincular 2 objetos a una línea - Elaboración Propia.....	67
Figura 49 Código para Actualizar la posición de la línea mientras se mueve un objeto - Elaboración Propia.....	68
Figura 50 Captura del resultado de vincular varios objetos y varias líneas - Elaboración Propia.	68
Figura 51 Página en GitHub del Director.JS - Fuente https://github.com	69
Figura 52 Página de Theen.js – Fuente: http://www.createjs.com/#!/TweenJS	69
Figura 53 Lugares donde al presionar con el cursor se Inician las Animaciones – Elaboración Propia.....	70
Figura 54 Secuencia de Animación - Elaboración propia	71
Figura 55 Callbacks para secuenciar 2 animaciones - Elaboración Propia	72
Figura 56Modelo 3D: PC - Elaboración Propia.	78
Figura 57Modelo 3D: Laptop - Elaboración Propia.....	79
Figura 58 Modelo 3D: Router - Elaboración Propia.....	79
Figura 59 Exportar 3DsMax a Objeto compatible Web – Elaboración Propia.....	80
Figura 60 Comando Python para convertir un Archivo OBJ a JS – Elaboración Propia	80

INDICE DE CUADROS.

Cuadro 1 Navegadores Compatibles con WebGL. Fuente: http://caniuse.com/webgl	16
Cuadro 2 Tareas para compatibilidad. - Elaboración Propia.....	35
Cuadro 3 Tareas para Experiencia Usuario. - Elaboración Propia.....	35
Cuadro 4 Tareas para contenido gráfico - Elaboración Propia.	36
Cuadro 5 Tareas para Interactividad - Elaboración Propia.	36
Cuadro 6 Tareas para Animación - Elaboración Propia.....	37
Cuadro 7 Resumen Global del Release Backlog – Elaboración Propia.....	40
Cuadro 8 Resumen de Tareas del 1er Sprint - Elaboración Propia	41
Cuadro 9 Burndown Chart del Sprint 1 - Elaboración Propia.....	42
Cuadro 10 Resumen de Tareas del 2do Sprint - Elaboración Propia	61
Cuadro 11 Burndown Chart del Sprint 2 - Elaboración Propia.....	61
Cuadro 12 Resumen de Tareas del 3er Sprint - Elaboración Propia	76
Cuadro 13 Burndown Chart del Sprint 3 - Elaboración Propia.....	76

CAPÍTULO 1: INTRODUCCIÓN.

1. Introducción.

El reto de transmitir el conocimiento abstracto ha sido un desafío constante para las técnicas de enseñanza, de manera que se logre en el menor tiempo posible, obtener la comprensión de los conceptos iniciales importantes.

Es en esta búsqueda de hallar herramientas que permitan apoyar a este deseo es que se presenta como una alternativa a la capacidad de la tecnología de crear entornos 3D que faciliten la comprensión de la teoría planteada donde el estudiante visualice, interactúe y experimente con el contenido educativo, mediante el uso cada vez más extendido de dispositivos móviles.

Para el desarrollo de este software, se promueve la utilización de tecnologías cuyas características principales son: manipular elementos tridimensionales, animaciones predefinida e interactiva, con la finalidad de que el estudiante sea capaz de abstraer los contenidos educativos de forma visual y atractiva, como una innovación a la forma tradicional de compartir conocimiento e incrementa el interés por aprender.

1.1. Muestra de aplicación.

Este trabajo es un prototipo, dirigido a los estudiantes de niveles Iniciales de la Universidad Mayor de San Simón, en las carreras de Lic. Informática, Ing. de Sistemas e Ing. Electrónica, de esta manera colaborar en las actividades prácticas en las materias que contiene:

- Estructuras de Datos como Listas, Pilas, Arboles;
- Redes de Computadoras donde se tienen dificultades al comprender transporte de información e interconexión entre dispositivos de red.

Estos conceptos abstractos y con dificultad comprensión en su fase inicial requiere que el estudiante realice prácticas dentro y fuera de aula para así poder comprender el cómo funciona, es ahí donde se desea insertar esta herramienta para acelerar este proceso importante en la formación de todo estudiante, donde el concepto principal es tocar ver e experimentar con entornos que permitan construir el conocimiento sólido en estos conceptos.

Gracias a las tecnologías más recientes de HTML5, que hacen eficiente la velocidad optimizando el transporte de información, usando un mínimo de recursos multimedia (Imágenes, audio, video...) e incrementando la generación de gráficos mediante código de programación (vectores, coordenadas, librerías optimizadas), se puede notar que se consiguen efectos visuales muy atractivos a la vista, con un tiempo de espera muy corto.

1.2. Esperanza

Se tiene la esperanza de lograr un cambio de las aplicaciones en 2D al modelo 3D, dado que pueden interactuar y facilitan el hecho de mostrar cualquier concepto de la vida real en general.

1.3. Descripción de la Problemática.

La educación es parte muy importante del desarrollo del conocimiento de las personas, en la variedad de métodos de educación a grupos de personas, se tiene el uso de la pizarra y el marcador, data Display, pizarras interactivas, lectura de libros, etc.

Estos métodos explican los **conceptos teóricos** que enriquecen el aprendizaje, pero en algunas circunstancias al momento de mostrar, describir, o representar conceptos muy complejos o abstractos, se dificulta el proceso.

Por ejemplo: teóricamente se podría explicar el funcionamiento de las leyes físicas, para empezar esto toma tiempo y tiende a ser confuso cuando el tema depende de enseñanzas previas y en experiencia personal tiende a mezclarse u olvidarse rápidamente. Existen clases

prácticas o laboratorios que permiten apoyar este proceso. Pero esto requiere una infraestructura y horario adicional.

Para dar un ejemplo concreto: para la materia de Elementos de programación y estructuras de datos existe una dificultad en los estudiantes de comprender los conceptos de Estructuras de datos, puesto que para entender estos conceptos es necesario entender la idea de manera espacial y abstracta, los docentes recurren en su explicación a gráficos representativos en el pizarrón, que en ocasiones le toma tiempo dibujar.

Por otro lado el hardware y el software de los dispositivos de consumo de información (PC, Laptop, Tablet, Smartphone), han evolucionado rápidamente en cuanto a capacidad, velocidad y rendimiento, especialmente en gráficos, esto puede ser aprovechado al máximo al desarrollar el software, además de incluir la capacidad de interactuar que ofrece la tecnología haciendo una experiencia completa para el reto de aprender los conocimiento básicos e importantes en la formación profesional.

1.4. Objetivos.

1.4.1. Analizar Materias.

Analizar información del proceso de enseñanza aprendizaje actual en las asignaturas especificadas para obtener el contenido educativo de la aplicación.

1.4.2. Producir Introducción.

Producir una introducción gráfica para implementar en las secciones de cada tema el contenido educativo necesario, colaborando con docentes especializados.

1.4.3. Seleccionar Tecnologías.

Seleccionar tecnologías, arquitecturas de software, procedimientos y estándares de desarrollo web tridimensional adecuados para el desarrollo del software.

1.4.4. Animaciones.

Desarrollar animaciones didácticas predefinidas para introducir al estudiante en los conceptos educativos.

1.4.5. Interactividad.

Desarrollar la capacidad de los objetos 3D para que respondan a acciones que los estudiantes hagan sobre ellos.

CAPÍTULO II: MARCO REFERENCIAL

2. Marco Referencial.

2.1. WebGL.



Figura 1: significado de WebGL y Navegadores que lo soportan. Fuente: Interactive 3D Graphics Course With Three - VIDEO

Según la definición de (Wikipedia, 2013) “WEBGL es una especificación estándar que está siendo desarrollada actualmente para mostrar gráficos en 3D en navegadores web. El WebGL permite mostrar gráficos en 3D acelerados por hardware (GPU¹) en páginas web, sin la necesidad de *plug-in*²s en cualquier plataforma que soporte OpenGL³ 2.0 u OpenGL ES 2.0. Técnicamente es un API para JavaScript que permite usar la implementación nativa de OpenGL ES 2.0 que será incorporada en los navegadores. WebGL es gestionado por el consorcio de tecnología sin ánimo de lucro Khronos Group.”

2.1.1. Experiencia Personal con WebGL.

WebGL prácticamente nos permite mostrar cualquier contenido 3D, con animaciones e interactividad, está siendo usado en una gran variedad de aplicaciones de empresas como Google por ejemplo GoogleMaps, que muestra el con relieve y textura 3D los edificios, casas parques directamente en el Navegador sin instalar absolutamente nada lo cual es fantástico, este soporte de empresas tan grandes hace que esta tecnología tenga continuidad en el futuro y comunidades activas que la mantenga desarrollándose.

Previamente estaba desarrollando en una tecnología llamada x3D, la cual tenía un par de inconvenientes:

- **La necesidad de un Plug-in** de terceros instalado en el Navegador del Usuario que quisiera ver el contenido creado en 3D, que posteriormente de descargar toca instalarla. Esto desafortunadamente crea una dependencia innecesaria, para el usuario pueda llegar al contenido.
- **Escasa documentación**, no tenía una comunidad activa, lo que dificulta poder programar y personalizar las cosas que uno quisiera hacer.

¹ GPU del Inglés “Graphical Processing Unit” unidad de procesamiento gráficos.

² *Plug-in* es un término informático para describir a una extensión adicional externa a un programa.

³ OpenGL del inglés “Open Graphic Library” una especificación Estándar que define una api Multilenguaje y Multiplataforma.

También pase por *Flash* con su lenguaje ActionScript⁴ que puede representar contenido 3D, pero necesita también un plug-in llamado FlashPlayer previamente instalado. Con un limitado conjunto de funciones para desarrollar en 3D las animaciones y otros inconvenientes, además Flash está siendo desplazado por HTML5, css3, y JavaScript por lo que también fue descartado.

Investigando encontré esta especificación, WebGL, que tiene ventajas sobre otras opciones para representar 3D. La ventaja que más me llamó la atención es su aceleración por Hardware: utiliza directamente el acelerador de gráficos (GPU), sin intermediarios.

Intermediarios como por ejemplo aceleración por Software que el resultado es más lento y consume más procesador y por ende Batería.

2.1.1. Soporte de WebGL.

WebGL obviamente funciona en la web, el requisito es tener un navegador web, tenemos para escoger entre un conjunto de navegadores para poder visualizar el 3D sin instalar nada adicional, como esta especificación está en desarrollo.

Tenemos por otro lado una variedad de empresas que dan soporte a los navegadores, empresas que toman decisiones, actualizan, etc. no todos soportan WebGL, lo bueno es que los principales si incluyen soporte WebGL en sus navegadores:

⁴ ActionScript es un Lenguaje de Programación basado en la notación ECMA-SCRIPT.

2.1.1.1. Navegadores.

Aquí se muestra un cuadro con los navegadores que soportan WebGL a distintos niveles:

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Blackberry Browser	Opera Mobile	Chrome for Android	Firefox for Android
26 versions back			4.0: Not supported									
25 versions back		2.0: Not supported	5.0: Not supported									
24 versions back		3.0: Not supported	6.0: Not supported									
23 versions back		3.5: Not supported	7.0: Not supported									
22 versions back		3.6: Not supported	8.0: Partial support									
21 versions back		4.0: Partial support	9.0: Partial support									
20 versions back		5.0: Partial support	10.0: Partial support									
19 versions back		6.0: Partial support	11.0: Partial support									
18 versions back		7.0: Partial support	12.0: Partial support									
17 versions back		8.0: Partial support	13.0: Partial support									
16 versions back		9.0: Partial support	14.0: Partial support									
15 versions back		10.0: Partial support	15.0: Partial support									
14 versions back		11.0: Partial support	16.0: Partial support									
13 versions back		12.0: Partial support	17.0: Partial support		9.0: Not supported							
12 versions back		13.0: Partial support	18.0: Supported		9.5-9.6: Not supported							
11 versions back		14.0: Partial support	19.0: Supported		10.0-10.1: Not supported							
10 versions back		15.0: Partial support	20.0: Supported		10.5: Not supported							
9 versions back		16.0: Partial support	21.0: Supported		10.6: Not supported							
8 versions back		17.0: Partial support	22.0: Supported		11.0: Not supported							
7 versions back		18.0: Partial support	23.0: Supported	3.1: Not supported	11.1: Not supported							
6 versions back	5.5: Not supported	19.0: Partial support	24.0: Supported	3.2: Not supported	11.5: Not supported			2.1: Not supported		10.0: Not supported		
5 versions back	6.0: Not supported (but has polyfill available)	20.0: Partial support	25.0: Supported	4.0: Not supported	11.6: Not supported	3.2: Not supported		2.2: Not supported		11.0: Not supported		
4 versions back	7.0: Not supported (but has polyfill available)	21.0: Partial support	26.0: Supported	5.0: Not supported	12.0: Partial support	4.0-4.1: Not supported		2.3: Not supported		11.1: Not supported		
3 versions back	8.0: Not supported (but has polyfill available)	22.0: Partial support	27.0: Supported	5.1: Partial support	12.1: Partial support	4.2-4.3: Not supported		3.0: Not supported		11.5: Not supported		
2 versions back	9.0: Not supported (but has polyfill available)	23.0: Partial support	28.0: Supported	6.0: Partial support	15.0: Supported	5.0-5.1: Not supported		4.0: Not supported		12.0: Partial support		
Previous version	10.0: Not supported (but has polyfill available)	24.0: Partial support	29.0: Supported	6.1: Partial support	16.0: Supported	6.0-6.1: Not supported		4.1: Not supported	7.0: Not supported	12.1: Partial support		
Current	11.0: Supported	25.0: Partial support	30.0: Supported	7.0: Partial support	17.0: Supported	7.0: Not supported	5.0-7.0: Not supported	4.2-4.3: Not supported	10.0: Supported	16.0: Supported	30.0: Not supported	25.0: Supported
Near future		26.0: Partial support	31.0: Supported		18.0: Supported			4.4: Not supported				
Farther future		27.0: Partial support	32.0: Supported									

Cuadro 1 Navegadores Compatibles con WebGL. Fuente: <http://caniuse.com/webgl>

2.1.1.2. Hardware.

En cuanto al hardware simplemente es necesario tener una GPU que soporte estándar a WebGL, ya que WebGL utiliza instrucciones del procesador de gráficos, este tiene que tener estas instrucciones disponibles en su núcleo.

En mi experiencia personal, equipos fueron fabricados desde el 2005 en adelante, ya tienen este soporte. Lo cual deja muy pocos equipos que no soportan WebGL, Equipos antiguos.

Para los Teléfonos Inteligentes, y Tabletas, el soporte es generalizado.

Para verificar la compatibilidad de WebGL en un determinado Equipo, basta con ir a esta dirección:

<http://get.webgl.org>

Y verificar si se muestra el Cubo 3D en el centro de la pantalla como se muestra en la imagen.



Figura 2 Captura del Equipo del Navegador que soporta WebGL Fuente: <http://get.webgl.org>

2.2. JavaScript.

Todo lo anterior explicado a cerca de WebGL, tiene que ver más que todo con las ventajas a cerca de la compatibilidad de esta tecnología sobre distintas plataformas físicas y de software, pero ahora es cuando nos ponemos técnicos, JavaScript es el lenguaje que uso en este proyecto para implementar sus detalles y funcionamiento.

Según su definición: (Flanagan, 2002) "JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por

ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del (DOM⁵).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases.”

Para programar el contenido que se muestra a los usuarios, no descubrí la rueda, preferí utilizar una librería de JavaScript muy interesante llamada Three.js.

Esta librería abstrae toda la complejidad del uso de la API⁶ de WebGL. Y facilita de sobremanera el desarrollo ya que se encarga de hacer las operaciones complejas, dejando al desarrollador lo más que es más esencial: programar lo que se quiere mostrar con sus eventos interacciones y animaciones.

2.2.1. Three.js.

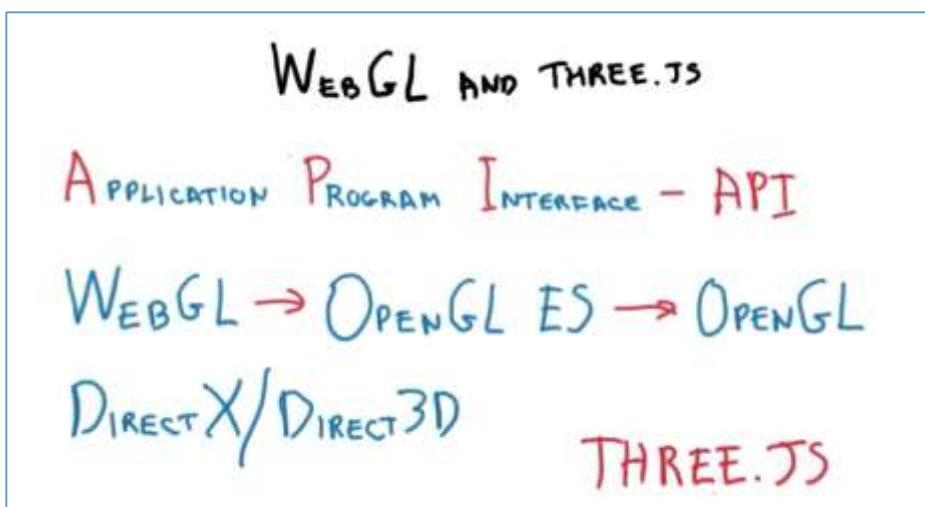


Figura 3. Representación del papel que cumple Three.js Fuente: Interactive 3D Graphics Course With Three - VIDEO

Básicamente crear una escena con un objeto girando con esta API, es tan sencillo como este código lo es:

```
<script>
  var camera, scene, renderer;
  var geometry, material, mesh;

  init();
  animate();
```

⁵ DOM del inglés Document Object Model es la estructura de un documento HTML representada jerárquicamente.

⁶ API del inglés Application Program Interface.

```

function init() {
    camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 1, 10000 );
    camera.position.z = 1000;

    scene = new THREE.Scene();

    geometry = new THREE.CubeGeometry( 200, 200, 200 );
    material = new THREE.MeshBasicMaterial( { color: 0xffff00, wireframe: true } );

    mesh = new THREE.Mesh( geometry, material );
    scene.add( mesh );

    renderer = new THREE.CanvasRenderer();
    renderer.setSize( window.innerWidth, window.innerHeight );

    document.body.appendChild( renderer.domElement );
}

function animate() {
    // note: three.js includes requestAnimationFrame shim
    requestAnimationFrame( animate );

    mesh.rotation.x += 0.01;
    mesh.rotation.y += 0.02;

    renderer.render( scene, camera );
}
</script>

```

Es todo!, 3D en el navegador web compatible en una amplia gama de plataformas.

Describiendo un poco el código se puede notar que hay:

- Una cámara que representa lo que el usuario verá.
- Una escena es el lugar donde incluimos los objetos que queremos Mostrar.
- Una Geometría que es el esqueleto de una figura, en este caso un Cubo.
- Un Material que es la envoltura que envolverá la Geometría.
- Una Malla Mesh que es un objeto que junta la geometría con el material.
- Incluimos la Malla en la escena para que la podamos ver.
- Tenemos un Renderer que se encarga de interpretar el código en el navegador.
- Al final hay una función que se encarga de animar, girando el cubo.

Esa es la estructura más básica de un escenario 3D que se muestra en el navegador Web utilizando Three.js.

2.2.2. Tween.js.

Tween.js es una Librería desarrollada en JavaScript que encontré en el camino, mientras desarrollaba el contenido, para animar los objetos.

Su objetivo básicamente es interpolar los cuadros entre 2 valores, en un tiempo dado, estos valores pueden ser un numero x, o un conjunto de valores como (x,y,z), en un tiempo t, lo cual es muy útil para definir el estado inicial y final de un objeto en el espacio.

Este estado puede ser una posición en el espacio, una rotación, un color o cualquier cosa que se nos pueda ocurrir.

Una característica adicional es que: se puede asignar un tipo de animación mientras dura la transición del estado inicial al final:

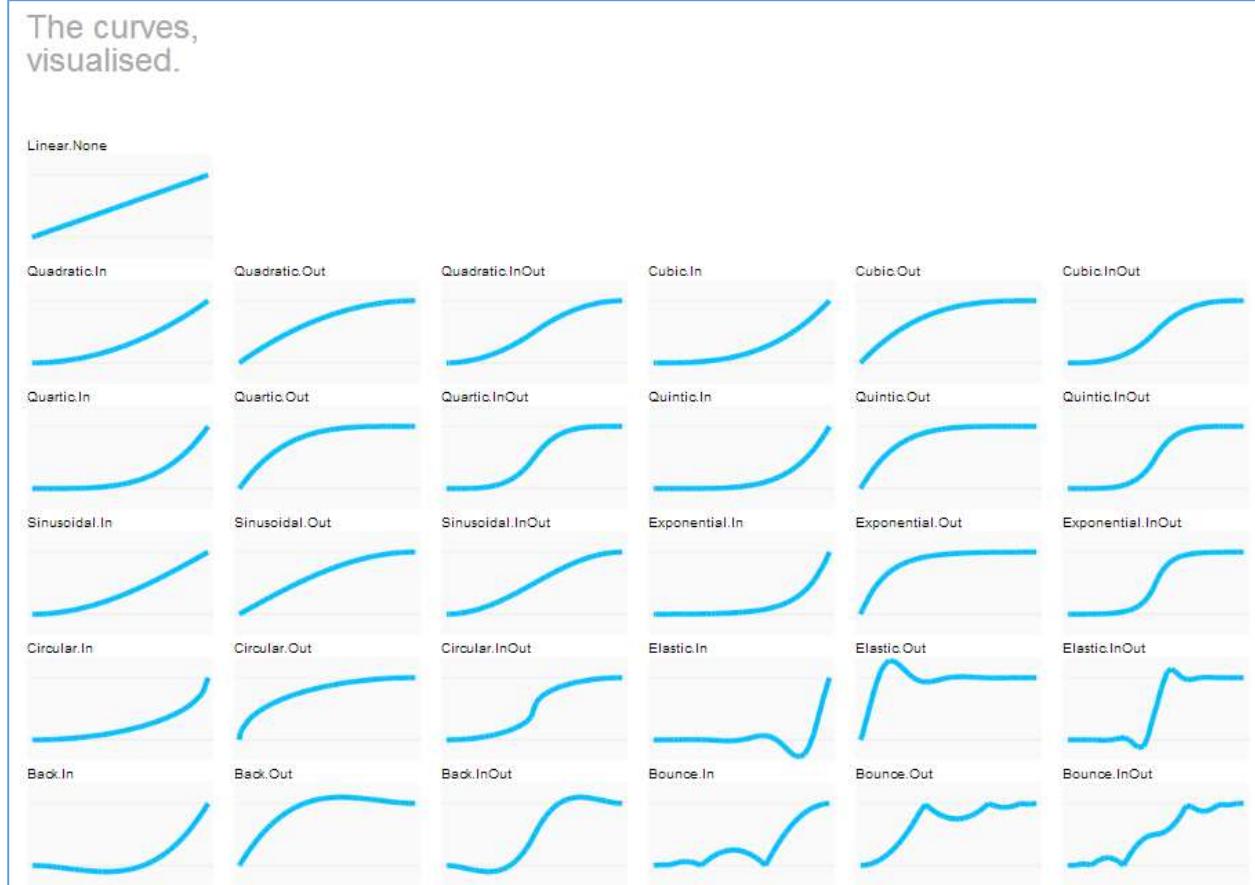


Figura 4. Diferentes Animaciones con Tween.js. Fuente: Ejemplos del Paquete Tween.js

Esta librería fue seleccionada de entre 2 opciones, la otra opción era Director.js que al final la descarté porque no tenía bastante soporte ni documentación, de hecho la última publicación de Director.js fue hace unos 3 años.

En cambio Tween.js fue desarrollado y actualizado constantemente, en septiembre fue su última actualización. Está bastante bien documentado.

La razón defina por la que seleccione Tween.js es porque permite encadenar animaciones una detrás de la otra, dinámicamente.

2.2.3. JQuery.

JQuery me permite interactuar con la interfaz y de forma muy simplificada. En el capítulo Desarrollo del Proyecto, más adelante en este libro, describo la utilidad técnica que me brinda esta librería.

2.2.4. JqueryMobile.

Es la razón por la que este proyecto se adapta para mostrarse en diferentes navegadores y plataformas.

Básicamente JqueryMobile me provee los elementos de interfaz que se adaptan al tamaño del navegador que el cliente este utilizado, y lo más importante funciona en dispositivos móviles. Todo lo que uno puede ver en el navegador se acomoda a diferentes tamaños redimensionándose y los elementos como botones y títulos siempre de una forma muy correcta.

2.3. Educación.

En el ámbito de la educación la aplicación de las tecnologías, intenta optimizar el proceso de enseñanza aprendizaje, reduciendo el tiempo de enseñanza.

El uso de tecnología en la educación es también asociado a un término más general conocido como TIC.

En un artículo (Aula Virtual UCV, 2011) se muestran las ventajas y desventajas del uso de TIC en educación Superior.

2.3.1. Ventajas uso Tic en el Ámbito Educativo.

Aprendizaje Cooperativo: Los instrumentos que proporcionan las TIC facilitan el trabajo en grupo y el cultivo de actitudes sociales, debido a que propician el intercambio de ideas y la cooperación.

Alto grado de interdisciplinariedad: Las tareas educativas realizadas con computadoras permiten obtener un alto grado de interdisciplinariedad, ya que el computador debido a su versatilidad y gran capacidad de almacenamiento, permite realizar diversos tipos de tratamiento de una información muy amplia y variada.

Alfabetización Tecnológica (digital, audiovisual): Hoy día aún existen grupos de estudiantes y profesores que se quedan rezagados ante el avance de las nuevas tecnologías, puntualmente en lo que respecta al uso del computador y la aplicación de sus respectivas herramientas. Por suerte cada vez es menor ese grupo y tienden a desaparecer. Dada las necesidades de nuestro mundo moderno, hasta para pagar las cuentas (electricidad, teléfono, etc.) se emplea el computador, de manera que la actividad académica no es la excepción. Profesor y estudiante sienten la necesidad de actualizar sus conocimientos y profundizar en temáticas de tecnología digital, formatos de audio y video, edición y montaje, entre otras.

2.3.2. Desventajas uso Tic en el Ámbito Educativo.

Dado que el aprendizaje cooperativo está sustentado en las actitudes sociales, una sociedad perezosa puede influir en el aprendizaje efectivo.

Debido al vertiginoso avance de las tecnologías, éstas tienden a quedarse descontinuadas muy pronto, lo cual obliga a actualizar frecuentemente el equipo y aprender a utilizar nuevos software.

El costo de la tecnología no es nada despreciable por lo que hay que disponer de un presupuesto generoso y frecuente que permita actualizar los equipos periódicamente. Además hay que disponer de lugares seguros para su almacenaje y, por tanto, prevención del robo de los equipos.

2.3.1. Ventajas uso Tic para el Docente.

Alto grado de interdisciplinariedad: Hoy día, el docente tiene que saber un poco de cada cosa, desde el punto de vista instrumental y operacional (conexión de equipos de audio, video, etc) manejo y actualización de software, diseño de páginas web, blog y muchas cosas más. El docente podrá interactuar con otros profesionales para refinar detalles.

Iniciativa y creatividad: Dado que el docente viene trascendiendo del ejercicio clásico de la enseñanza al modernismo, ese esfuerzo demanda mucha iniciativa y creatividad.

Aprovechamiento de recursos: Hay fenómenos que pueden ser estudiados sin necesidad de ser reproducidos en el aula. Muchas veces la proyección de un video o el uso de una buena simulación, pueden ser suficientes para el aprendizaje. Por otro lado, el uso del papel se puede reducir a su mínima expresión reemplazándolo por el formato digital. Actualmente, una enciclopedia, libros e informes, pueden ser almacenados en un CD o pen drive y de esta forma ser transferidos vía web a cualquier lugar donde la tecnología lo permita.

Aprendizaje cooperativo: El profesor aprende con sus estudiantes y los profesores con otros docentes, gracias a la cooperación y trabajo en equipo.

2.3.1. Desventajas uso Tic para el Docente.

Es necesario la capacitación continua de los docentes por lo que tiene que invertir recursos (tiempo y dinero) en ello.

Frecuentemente, el profesor se siente agobiado por su trabajo, por lo que muchas veces prefiere el método clásico evitando de esta manera compromisos que demanden tiempo y esfuerzo.

Hay situaciones muy particulares donde una animación, video o presentación, nunca podrán superar al mundo real por lo que es necesario la experimentación que solo se logra en un laboratorio o aula de clases bien equipada.

2.3.1. Ventajas uso Tic para los estudiantes.

Aprovechamiento del tiempo: El estudiante puede acceder a la información de manera casi instantánea, puede enviar sus tareas y asignaciones con solo un clic. Puede interactuar con sus compañeros y profesor desde la comodidad de su casa o de un “ciber café”, haciendo uso de salas de chat y foros de discusión. El profesor puede publicar notas, anotaciones, asignaciones y cualquier información que considere relevante, y de manera casi instantánea. En caso de no disponer de tiempo o equipo instrumental adecuado, el profesor puede mostrar el fenómeno en estudio **empleando alguna simulación disponible**.

Aprendizaje cooperativo: Aprenden con su profesor y los estudiantes pueden aprender entre ellos, gracias a la cooperación y trabajo en equipo.

Motivación e interés: Poseen destrezas innatas asociadas con las nuevas tecnologías por lo que de forma muy natural, aceptan y adoptan el uso del computador en sus actividades de aprendizaje, además prefieren la proyección de un video ante la lectura de un libro. Los estudiantes confiesan estar muy motivados porque tienen acceso a un gran volumen de información actualizada. Por otro lado, el profesor se siente comprometido con su actividad docente, por lo que se hace imperativo la actualización de su conocimiento, sobre todo cuando se contagia del entusiasmo de sus estudiantes.

Desarrollo de habilidades en la búsqueda de la información: Hasta hace apenas unas décadas, toda una tarde de consulta en la biblioteca no era suficiente para encontrar la información buscada. Hoy día basta con pocos minutos para saturarnos de información, muchas de ellas inútiles o repetidas. Es necesario desarrollar habilidades para seleccionar adecuadamente la información útil y filtrar lo inútil para quedarnos con una cantidad de información que podamos procesar.

CAPITULO 3: ÁREA DE APLICACIÓN.

3. Área de Aplicación.

La exploración de tecnologías web en 3D, aplicado a dos materias en forma de prototipo, escogidas por su dificultad para abstraer conceptos y complejidad teórica, donde se ve un cuello de botella de estudiantes en las aulas estas materias.

3.1. Análisis del contenido educativo.

Para el contenido educativo, las dificultades en la enseñanza las materias en cuestión, el contenido educativo está representado atractivamente de manera gráfica, para llamar la atención de los estudiantes. Este contenido ha sido seleccionado y se muestra un detalle a continuación.

3.1.1. Estructuras de datos.

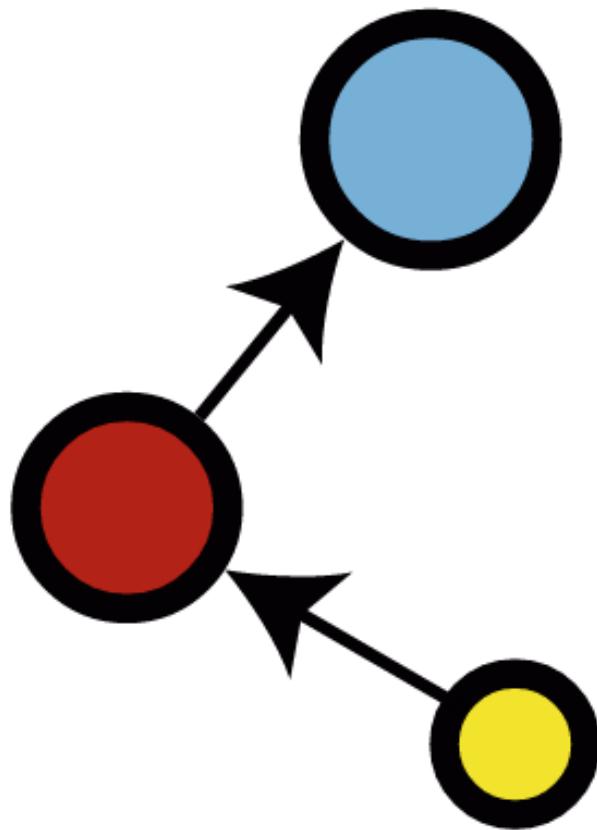


Figura 5: Estructura de datos, Elaboración propia

Según la definición de estructura de datos (Wikipedia, 2013). En programación, una estructura de datos es una forma de organizar un conjunto de datos elementales con el objetivo de facilitar su manipulación. Un dato elemental es la mínima información que se tiene en un sistema.

Una estructura de datos define la organización e interrelación de estos y un conjunto de operaciones que se pueden realizar sobre ellos. Las operaciones básicas son:

- Alta, adicionar un nuevo valor a la estructura.
- Baja, borrar un valor de la estructura.

- Búsqueda, encontrar un determinado valor en la estructura para realizar una operación con este valor, en forma secuencial o binario (siempre y cuando los datos estén ordenados).

Otras operaciones que se pueden realizar son:

- Ordenamiento, de los elementos pertenecientes a la estructura.
- Apareo, dadas dos estructuras originar una nueva ordenada y que contenga a las apareadas.

Cada estructura ofrece ventajas y desventajas en relación a la simplicidad y eficiencia para la realización de cada operación. De esta forma, la elección de la estructura de datos apropiada para cada problema depende de factores como la frecuencia y el orden en que se realiza cada operación sobre los datos.

3.1.1.1. Listas.

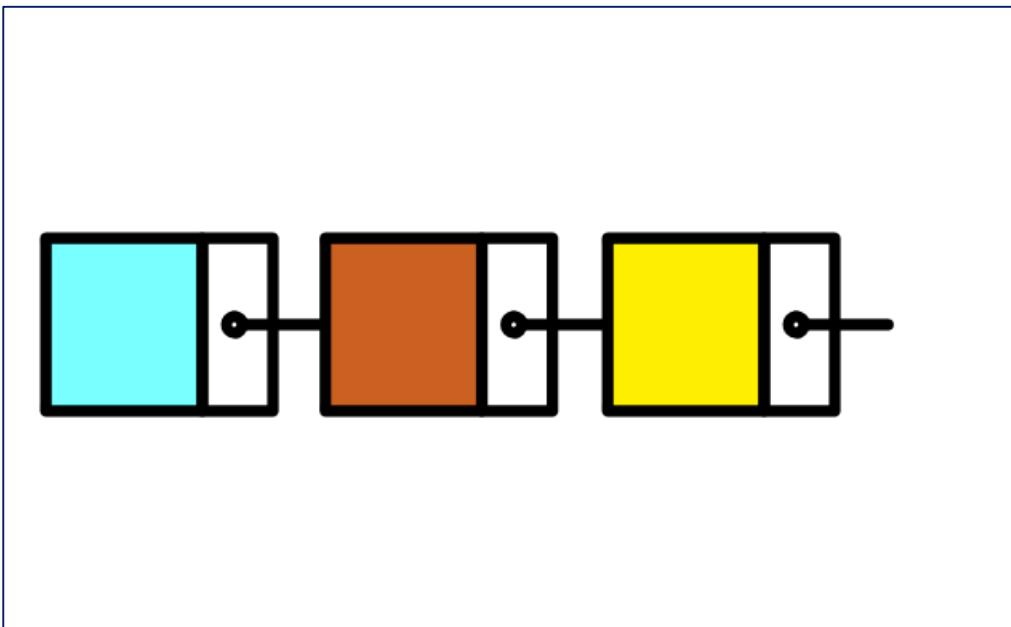


Ilustración 1 Ilustración de Listas - Elaboración Propia

En la definición de (Wikipedia, 2013), “una lista enlazada es una de las estructuras de datos fundamentales, y puede ser usada para implementar otras estructuras de datos. Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior. El principal beneficio de las listas enlazadas respecto a los vectores convencionales es que el orden de los elementos enlazados puede ser diferente al orden de almacenamiento en la memoria o el disco, permitiendo que el orden de recorrido de la lista sea diferente al de almacenamiento.

Una lista enlazada es un tipo de dato auto referenciado porque contienen un puntero o enlace (en inglés link, del mismo significado) a otro dato del mismo tipo. Las listas enlazadas permiten inserciones y eliminación de nodos en cualquier punto de la lista en tiempo constante (suponiendo que dicho punto está previamente identificado o localizado), pero no permiten un acceso aleatorio. Existen diferentes tipos de listas enlazadas: Lista Enlazadas Simples, Listas Dblemente Enlazadas, Listas Enlazadas Circulares y Listas Enlazadas Dblemente Circulares.

Las listas enlazadas pueden ser implementadas en muchos lenguajes. Lenguajes tales como Lisp y Scheme tienen estructuras de datos ya construidas, junto con operaciones para acceder a

las listas enlazadas. Lenguajes imperativos u orientados a objetos tales como C o C++ y Java, respectivamente, disponen de referencias para crear listas enlazadas.”

3.1.1.2. Pilas.

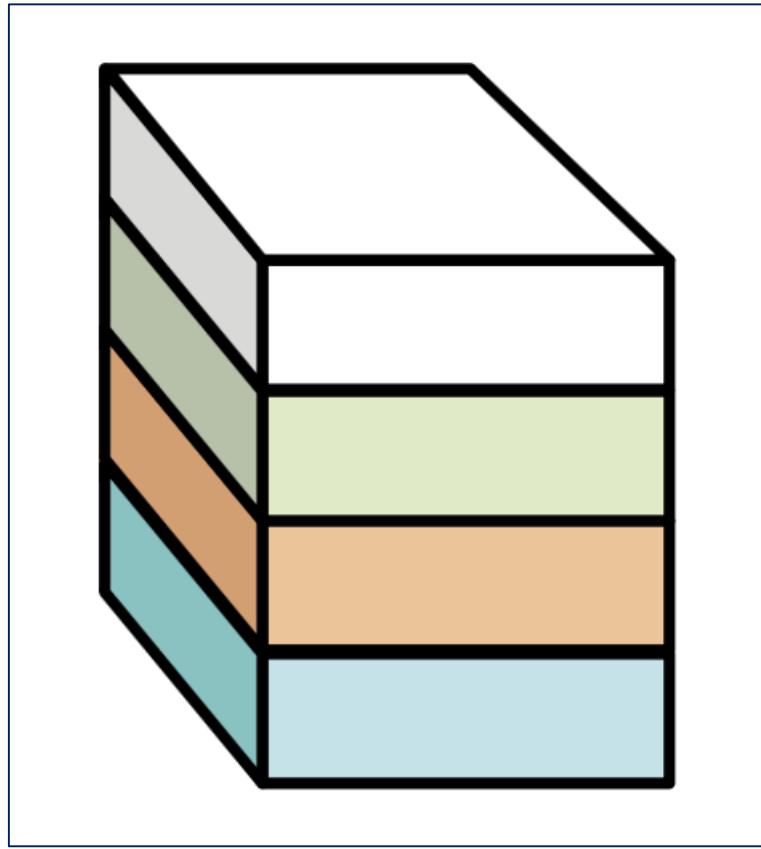


Figura 6: Figura de Una Pila en Estructura de Datos. Fuente: Elaboración Propia

Según la definición de (Wikipedia, 2013) Una pila (Stack en inglés) es una lista ordinal o estructura de datos en la que el modo de acceso a sus elementos es de tipo LIFO (del inglés Last In First Out, último en entrar, primero en salir) que permite almacenar y recuperar datos. Esta estructura se aplica en multitud de ocasiones en el área de informática debido a su simplicidad y ordenación implícita de la propia estructura.

Para el manejo de los datos se cuenta con dos operaciones básicas: apilar (push), que coloca un objeto en la pila, y su operación inversa, retirar (o desapilar, pop), que retira el último elemento apilado.

En cada momento sólo se tiene acceso a la parte superior de la pila, es decir, al último objeto apilado (denominado TOS, Top of Stack en inglés). La operación retirar permite la obtención de este elemento, que es retirado de la pila permitiendo el acceso al siguiente (apilado con anterioridad), que pasa a ser el nuevo TOS.

Por analogía con objetos cotidianos, una operación apilar equivaldría a colocar un plato sobre una pila de platos, y una operación retirar a retirarlo.

Las pilas suelen emplearse en los siguientes contextos:

- Evaluación de expresiones en notación postfija (notación polaca inversa).
- Reconocedores sintácticos de lenguajes independientes del contexto.
- Implementación de recursividad.

A modo de resumen tipo de datos, la pila es un contenedor de nodos y tiene dos operaciones básicas: push (o apilar) y pop (o desapilar). 'Push' añade un nodo a la parte superior de la pila, dejando por debajo el resto de los nodos. 'Pop' elimina y devuelve el actual nodo superior de la pila. Una metáfora que se utiliza con frecuencia es la idea de una pila de platos en una cafetería con muelle de pila. En esa serie, sólo la primera placa es visible y accesible para el usuario, todas las demás placas permanecen ocultas. Como se añaden las nuevas placas, cada nueva placa se convierte en la parte superior de la pila, escondidos debajo de cada plato, empujando a la pila de placas. A medida que la placa superior se elimina de la pila, la segunda placa se convierte en la parte superior de la pila. Dos principios importantes son ilustrados por esta metáfora: En primer lugar la última salida es un principio, la segunda es que el contenido de la pila está oculto. Sólo la placa de la parte superior es visible, por lo que para ver lo que hay en la tercera placa, el primer y segundo platos tendrán que ser retirados.

3.1.1.3. Árboles.

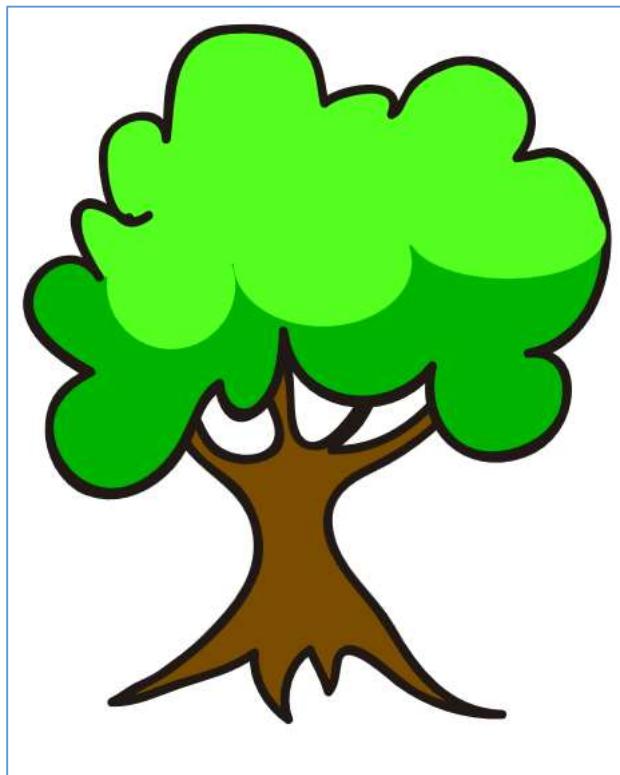


Figura 7Figura de Una Árbol para Estructura de Datos. Fuente: Elaboración Propia

Los árboles son estructuras de datos algo más compleja que las anteriores, puesto que involucra muchos conceptos, según su definición en (Wikipedia, 2013):

“En ciencias de la informática, un árbol es una estructura de datos ampliamente usada que imita la forma de un árbol (un conjunto de nodos conectados). Un nodo es la unidad sobre la que se construye el árbol y puede tener cero o más nodos hijos conectados a él. Se dice que un nodo a es padre de un nodo b si existe un enlace desde a hasta b (en ese caso, también decimos que b es hijo de a). Sólo puede haber un único nodo sin padres, que llamaremos raíz. Un nodo que no tiene hijos se conoce como hoja. Los demás nodos (tienen padre y uno o varios hijos) se les conoce como rama.”

3.1.1.3.1. Operaciones con Árboles.

Las operaciones comunes en árboles son:

- Enumerar todos los elementos.
- Buscar un elemento.
- Dado un nodo, listar los hijos (si los hay).
- Borrar un elemento.
- Eliminar un subárbol (algunas veces llamada podar).
- Añadir un subárbol (algunas veces llamada injertar).
- Encontrar la raíz de cualquier nodo.

3.1.2. Redes de Computadoras.

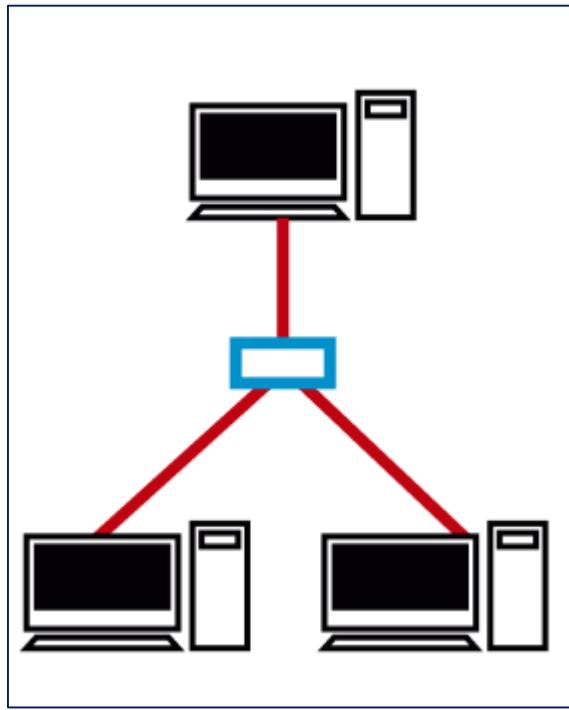


Figura 8: Redes de Computadoras, Elaboración Propia

Una red de computadoras, también llamada red de ordenadores, red de comunicaciones de datos o red informática, es un conjunto de equipos informáticos y software conectados entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos, con la finalidad de compartir información, recursos y ofrecer servicios.

Como en todo proceso de comunicación se requiere de un emisor, un mensaje, un medio y un receptor. La finalidad principal para la creación de una red de computadoras es compartir los recursos y la información en la distancia, asegurar la confiabilidad y la disponibilidad de la información, aumentar la velocidad de transmisión de los datos y reducir el costo general de estas acciones.² Un ejemplo es Internet, la cual es una gran red de millones de computadoras ubicadas en distintos puntos del planeta interconectadas básicamente para compartir información y recursos.

La estructura y el modo de funcionamiento de las redes informáticas actuales están definidos en varios estándares, siendo el más importante y extendido de todos ellos el modelo TCP/IP basado en el modelo de referencia OSI. Este último, estructura cada red en siete capas con

funciones concretas pero relacionadas entre sí; en TCP/IP se reducen a cuatro capas. Existen multitud de protocolos repartidos por cada capa, los cuales también están regidos por sus respectivos estándares.

3.1.2.1. Transporte de paquetes.

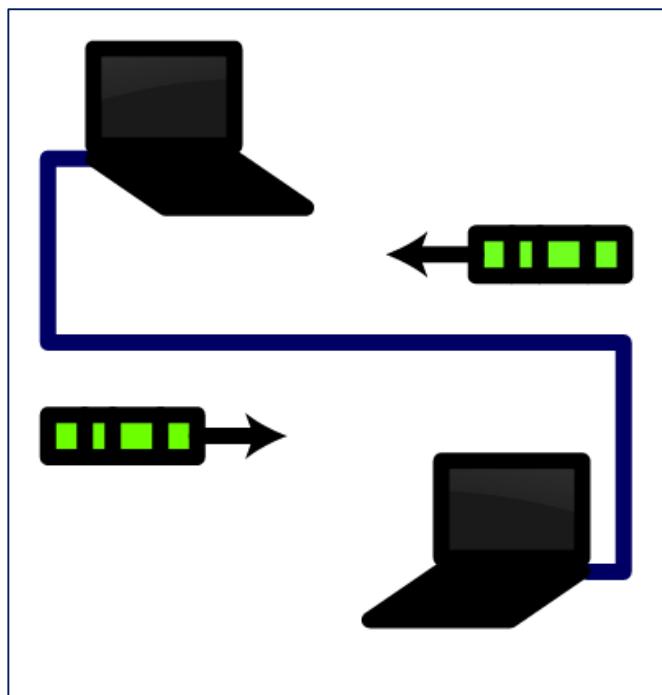


Figura 9: Transporte de Paquetes. Fuente Creación Propia

Para entender los conceptos primero vamos a ver que es un paquete de red.

3.1.2.1.1. Paquete de Red.

Se le llama paquete de red o paquete de datos a cada uno de los bloques en que se divide, en el nivel de Red, la información a enviar. Por debajo del nivel de red se habla de trama de red, aunque el concepto es análogo.

En todo sistema de comunicaciones resulta interesante dividir la información a enviar en bloques de un tamaño máximo conocido. Esto simplifica el control de la comunicación, las comprobaciones de errores, la gestión de los equipos de encaminamiento (routers), etc.

Un paquete de datos es una unidad fundamental de transporte de información en todas las redes de computadoras modernas. Un paquete está generalmente compuesto de tres elementos: una cabecera (header en inglés) que contiene generalmente la información necesaria para trasladar el paquete desde el emisor hasta el receptor, el área de datos (payload en inglés) que contiene los datos que se desean trasladar, y la cola (trailer en inglés), que comúnmente incluye código de detección de errores.

3.1.2.2. Establecimiento de Conexión TCP.

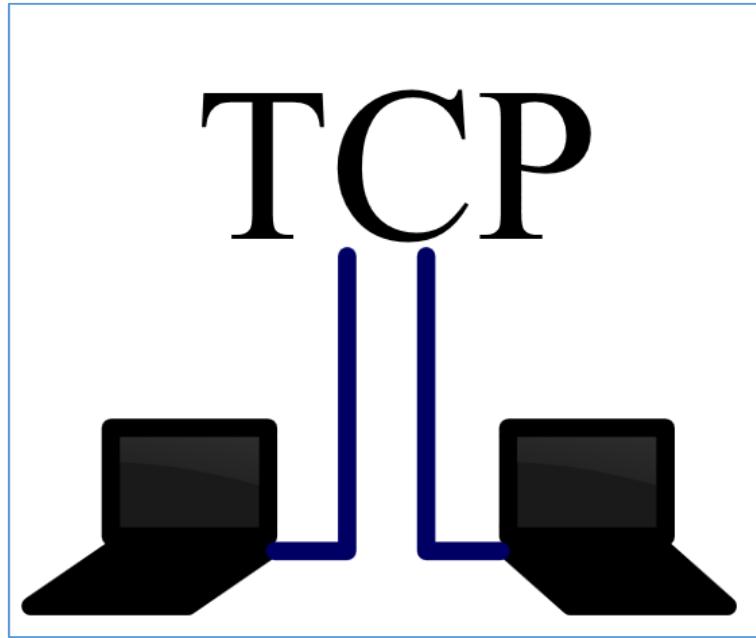


Figura 10. Establecimiento de Conexión TCP - Fuente Creación Propia

Aunque es posible que un par de entidades finales comiencen una conexión entre ellas simultáneamente, normalmente una de ellas abre un socket en un determinado puerto TCP y se queda a la escucha de nuevas conexiones. Es común referirse a esto como apertura pasiva, y determina el lado servidor de una conexión. El lado cliente de una conexión realiza una apertura activa de un puerto enviando un paquete SYN inicial al servidor como parte de la negociación en tres pasos. En el lado del servidor (este receptor también puede ser una PC o alguna estación terminal) se comprueba si el puerto está abierto, es decir, si existe algún proceso escuchando en ese puerto, pues se debe verificar que el dispositivo de destino tenga este servicio activo y esté aceptando peticiones en el número de puerto que el cliente intenta usar para la sesión. En caso de no estarlo, se envía al cliente un paquete de respuesta con el bit RST activado, lo que significa el rechazo del intento de conexión. En caso de que sí se encuentre abierto el puerto, el lado servidor respondería a la petición SYN válida con un paquete SYN/ACK. Finalmente, el cliente debería responderle al servidor con un ACK, completando así la negociación en tres pasos (SYN, SYN/ACK y ACK) y la fase de establecimiento de conexión. Es interesante notar que existe un número de secuencia generado por cada lado, ayudando de este modo a que no se puedan establecer conexiones falseadas (spoofing).

3.1.2.3. Redirección de paquetes.

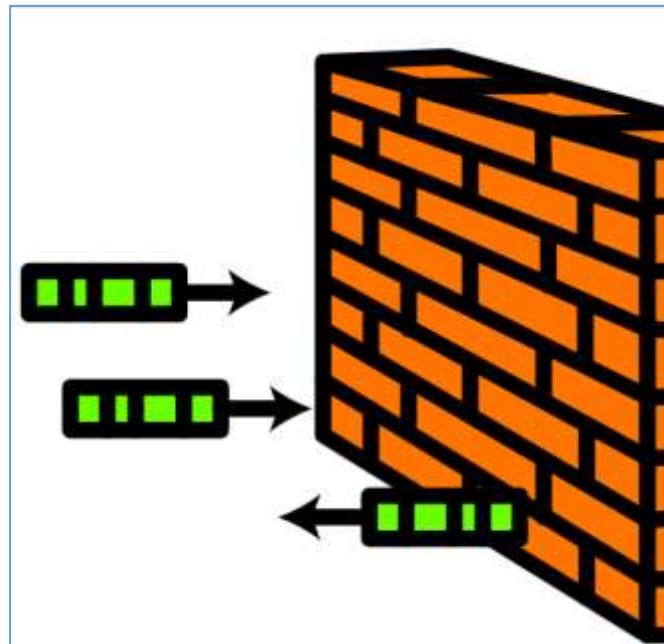


Figura 11 Redirección de Paquetes - Elaboración Propia

Para el contenido de esta sección de la aplicación web corresponde es necesario conocer conceptos el concepto de Firewall y Proxy.

3.1.2.3.1. Firewall

Según su definición (Wikipedia, 2013) “Un cortafuego (firewall en inglés) es una parte de un sistema o una red que está diseñada para bloquear el acceso no autorizado, permitiendo al mismo tiempo comunicaciones autorizadas.

Se trata de un dispositivo o conjunto de dispositivos configurados para permitir, limitar, cifrar, descifrar, el tráfico entre los diferentes ámbitos sobre la base de un conjunto de normas y otros criterios.

Los cortafuegos pueden ser implementados en hardware o software, o una combinación de ambos. Los cortafuegos se utilizan con frecuencia para evitar que los usuarios de Internet no autorizados tengan acceso a redes privadas conectadas a Internet, especialmente intranets. Todos los mensajes que entran o salgan de la intranet pasan a través del cortafuego, que examina cada mensaje y bloquea aquellos que no cumplen los criterios de seguridad especificados. También es frecuente conectar al cortafuego a una tercera red, llamada «zona desmilitarizada» o DMZ, en la que se ubican los servidores de la organización que deben permanecer accesibles desde la red exterior.

Un cortafuegos correctamente configurado añade una protección necesaria a la red, pero que en ningún caso debe considerarse suficiente. La seguridad informática abarca más ámbitos y más niveles de trabajo y protección.

3.1.2.3.2. Proxy

Un proxy, en una red informática, es un programa o dispositivo que realiza una acción en representación de otro, esto es, si una hipotética máquina A solicita un recurso a una C, lo hará mediante una petición a B; C entonces no sabrá que la petición procedió originalmente de A. Esta situación estratégica de punto intermedio suele ser aprovechada para soportar una serie de funcionalidades: proporcionar caché, control de acceso, registro del tráfico, prohibir cierto tipo de tráfico, etc.

Su finalidad más habitual es la de servidor proxy, que consiste en interceptar las conexiones de red que un cliente hace a un servidor de destino, por varios motivos posibles como seguridad, rendimiento, anonimato, etc. Esta función de servidor proxy puede ser realizada por un programa o dispositivo.

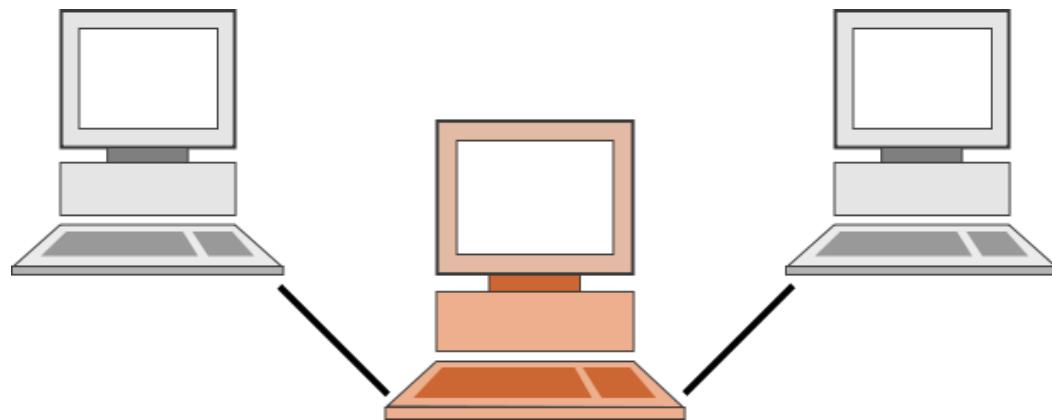


Figura 12 Servidor proxy conectando indirectamente dos ordenadores. – Fuente: Schematic_Proxy_Server.svg

CAPÍTULO 4: DESARROLLO DEL PROYECTO

Para el desarrollo del proyecto se utilizaron los componentes.

4. Desarrollo del Proyecto.

4.1. Proceso basado en la Filosofía Scrum.

4.2. Roles del equipo.

Para el caso particular del proyecto de grado, dado que no se cuenta con un equipo completo, simplemente mi persona, entonces es sencillo entender que asumiré distintos roles (lo cual no es común según la filosofía Scrum).

4.2.1. Propietario del producto.

Para este puesto, dentro de lo que es el Equipo de desarrollo del software, esta obviamente designado el tutor. **Yony R. Montoya B.**, que eventualmente, va indicando que es lo que se las características del prototipo de software, y tomando decisiones de hacia dónde va el proyecto.

4.2.2. Scrum Master.

En este cargo actúa: **Ivo Rojas**, como principal guía del proyecto, para encaminar los procesos y los tiempos de manera correcta, para llegar al objetivo con las previsiones, pruebas, retroalimentación necesaria para encaminar el proyecto y sobre todo para motivar al resto del equipo.

4.2.3. Equipo Desarrolladores.

Estos cargos implica el desarrollo del prototipo de software como tal, que por su tamaño se divide en las siguientes áreas.

- Desarrollador (Arquitecto).
- Desarrolladores (software).

4.3. Usuarios del producto.

El producto es de utilidad para personas, que desean aprender, para este fin, se contemplan los siguientes usuarios.

4.3.1. Estudiante.

El estudiante es considerado como el principal usuario para este prototipo de software, con el objetivo de tener la atención del mismo, se consideran los siguientes aspectos como prioridad del prototipo de software:

4.3.1.1. Rapidez de respuesta.

La rapidez de la respuesta de la página web es muy importante, porque es un hecho que las personas gozamos de poco tiempo en la vida, y los estudiantes aún menos, además la atención de las personas se desvía muy fácilmente cuando algo se demora más de lo normal, por ejemplo una charla muy redundante, una fila en el banco, un atasco en el tránsito, las personas se aburren muy rápido.

Por estas razones el software está altamente optimizado para dar respuesta lo más rápido posible, y aunque se depende de la web y la velocidad de conexión, sí que carga bastante rápido las librerías y los recursos necesarios para mostrar contenido 3D, sobre el navegador de los clientes web.

4.3.1.2. Compatibilidad.

Se trabajó bastante en este aspecto, porque hoy en día existe una amplia gama de dispositivos electrónicos, que muestran contenido WEB, dígase Smartphones, Tablets, Laptops, etc.

Y por otro lado existen un conjunto similarmente amplio de sistemas operativos, versiones de los mismo y navegadores web, (de Momento Google Chrome, Firefox y navegadores por defecto en Android), que en su mayoría están ampliamente extendidos entre las sociedades del mundo, son infraestructuras sencillas de acceder para mostrar el contenido desarrollado en este software.

4.3.1.3. Colores.

Los colores capturan mucho la atención del usuario, no solo aquí, sino en distintos ámbitos. Con la atención centrada en lo que se está mostrando, se consolida un recuerdo de lo que se está aprendiendo más profundo.

4.3.1.4. Mas Gráficos menos Texto.

La cantidad abrumadora de texto al aprender, hace que las personas disminuya el interés de las de leer contenidos muy extensos. Por esta razón de manera gráfica y simple se puede mejorar el interés aprender los conceptos que luego son adoptados de mejor manera.

Esto no quiere decir que leer sea malo, sino que se puede ahorrar tiempo aprendiendo conceptos generales que se quedan más tiempo en los recuerdos, ya que viene acompañados con gráficos.

4.3.1.5. Movimiento e Interactividad.

Esta es la parte fundamental del software, dado que los objetos en las diferentes secciones sirven para interactuar con los usuarios.

Esta interacción hace que el usuario no se quede estático viendo un contenido... sino que puede ver qué ocurre cuando interactúa con el contenido, eso es muy importante porque de esta manera las personas toman decisiones de cambiar ciertos aspectos dinámicamente, lo que les hace sentir que tienen un buen nivel de control.

4.3.2. Docente.

El docente está encargado de promover, y contribuir en futuras mejoras del software, puesto que es para beneficiar en el tiempo y calidad de la enseñanza.

4.4. Product Backlog.

Estas características son deseables para que este producto final, están escritas de forma general, esencialmente expresan las necesidades de un contenido que capture la atención de los usuarios.

Disponibilidad
en diferentes
plataformas.

usabilidad
intuitiva.

Mostrar
Gráficamente
el contenido

Interactividad
con los
contenidos

Animación.

Documento
Proyecto Final.

4.5. Release Backlog.

4.5.1. Estimación de Tiempos.

Disponibilidad en diferentes plataformas.

Tiempo: 9.25 Días.

Prioridad: Alta.

Asignado: Ivo Rojas.

Descripción: la aplicación web debe funcionar en la mayoría de las Pcs, laptops, tabletas, teléfonos móviles inteligentes, indistintamente del sistema operativo Windows, Mac, Linux.

N	Tarea	D	H	Prior
1	Buscar una tecnología de alta compatibilidad	3		1
2	Desarrollar para el navegador Web		4	1
3	Implementar Responsive Design en Template.		2	3
5	Desarrollo del lado del Cliente.	3		3
6	Desarrollar del lado del Servidor.	2		2
4	Implementar Responsive Design en panel 3D.		4	3

Total 8,00 10,0

Total Días 9,25

Cuadro 2 Tareas para compatibilidad. - Elaboración Propia.

Usabilidad intuitiva.

Tiempo: 11.1 Días.

Prioridad: Alta.

Asignado: Ivo Rojas.

N	Tarea	D	H	Prior
8	Crear Imágenes Representativas.		7	3
9	definir la forma y colores del tema de la aplicación		5	3
10	Desarrollar la vista Principal.	1		1
13	Desarrollar las vistas Dev (pruebas)		5	2
7	Optimizar Experiencia de Usuario.	2		2
11	Desarrollar las vistas Estructuras de Datos		3	1
14	Desarrollar vista Plantillas: listas pilas arboles	3		1
12	Desarrollar las vistas Redes		5	1
15	Desarrollar vista Plantillas: sección de Redes	2		1

Total 8,0 25,0

Total Días 11,1

Cuadro 3 Tareas para Experiencia Usuario. - Elaboración Propia.

Mostrar Gráficamente el contenido

Tiempo: 2,7 Días.
 Prioridad: Alta.
 Asignado: Ivo Rojas.

N	Tarea	D	H	Prioridad
16	Crear un objeto en el Espacio.		3	2
21	Crear una Línea.		3	3
17	Crear un modelo 3D de una PC		3	3
18	Crear un modelo 3D de una Laptop		3	3
19	Crear un modelo 3D Router		2	3
20	Importar Objetos de 3DS Max ⁷ mediante la API Three.js	1		2

Total 1,0 14,0

Total Días 2,7

Cuadro 4 Tareas para contenido gráfico - Elaboración Propia.

Interactividad con los contenidos

Tiempo: 17.5 Días.
 Prioridad: Alta.
 Asignado: Ivo Rojas.

N	Tarea	D	H	Prioridad
22	Eventos con el mouse para Línea.	1		3
23	eventos para click en el Entorno	2		3
26	Mover la Cámara en entorno al origen con Mouse(0,0,0)	1		2
27	Evento que arrastra un objeto sobre el espacio (Drag)	5		2
28	Vincular una línea a un Objeto	2		3
30	Ordenar Código en forma de Pseudo Objetos.	2		3
24	Eventos para los botones de la interfaz interactiva		6	2
25	Touch del mouse para dispositivos móviles		6	3
29	Vincular los 2 extremos de línea a 2 Objetos	3		2

Total 16,0 12,0

Total Días 17,5

Cuadro 5 Tareas para Interactividad - Elaboración Propia.

⁷ 3DSMax – 3D Studio Max: Software de modelado y renderización en 3D | Autodesk

Animación.

Tiempo: 20,5 Días.

Prioridad: Alta.

Asignado: Ivo Rojas.

N	Tarea	D	H	Prior
31	eventos disparadores de animaciones		6	3
32	crear animación para listas	2		1
33	crear animación para Pilas	2		2
34	Crear animación para Arboles	3		3
38	Motor de Animación entre Tween.js y Director.js	1		1
39	Encadenar Animaciones una después de Otra		6	2
40	Usar Recursividad y callbacks en JavaScript	2		3
35	Desarrollar animaciones para Transporte de P.	3		2
36	Desarrollar animaciones para Redirección de P.	3		3
37	Desarrollar animaciones para Establecimiento de C.	3		1

Total 19,0 12,0

Total Días

20,5

Cuadro 6 Tareas para Animación - Elaboración Propia.

Documento Proyecto Final.

Tiempo: 14 Días.

Prioridad: Alta.

Asignado: Ivo Rojas.

4.5.2. Tareas y selección de características deseables.

A continuación se tienen las tareas para desarrollar la aplicación web, además se muestran las características deseables de color amarillo y las que se descartaron de color Blanco.

Buscar
Tecnologías de
alta
compatibilidad

Desarrollar
para el
navegador

Desarrollo del
lado del
cliente.

Desarrollo del
lado del
Servidor.

Implementar
Responsive
Design en
Template.

Implementar
Responsive
Design en
panel 3D.

Optimizar
Experiencia de
usuario.

Crear
Imágenes
Representativa
s.

Definir la forma
y colores del
tema de la
aplicación

Desarrollar la
vista Principal.

Desarrollar las
vistas
Estructuras de
Datos

Desarrollar las
vistas Redes

Desarrollar vista
Plantilla para
listas pilas
árboles etc

Crear una esfera
en el Espacio.

Crear un modelo
3D de una PC

Crear un modelo
3D de una
Laptop

Crear un modelo
3D de un Router

Crear una Línea

Eventos con el
mouse.

Eventos para
click



4.6. Definición y Desarrollo de Sprints.

De manera global se tienen todas tareas organizadas con la información del Grupo, la tarea en sí, el sprint en el que se desarrollaran la cantidad de días, la cantidad de horas que se estima que demorar el desarrollo, y la prioridad que se le asignó.

Esta ordenado por orden de Prioridad y de Sprint.

N	Grupo	Tarea	Sprint	D	H	Prior
1	Compatibilidad	Buscar una tecnología de alta compatibilidad	1	3		1
2	Compatibilidad	Desarrollar para el navegador Web	1		4	1
10	UX	Desarrollar la vista Principal.	1	1		1
6	Compatibilidad	Desarrollar del lado del Servidor.	1	2		2
26	Interactividad	Mover la Cámara en entorno al origen con Mouse(0,0,0)	1	1		2

27	Interactividad	Evento que arrastra un objeto sobre el espacio (Drag)	1	5		2
16	Objetos 3D	Crear un objeto en el Espacio.	1		3	2
13	UX	Desarrollar las vistas Dev (pruebas)	1		5	2
3	Compatibilidad	Implementar Responsive Design en Template.	1		2	3
5	Compatibilidad	Desarrollo del lado del Cliente.	1	3		3
22	Interactividad	Eventos con el mouse para Línea.	1	1		3
23	Interactividad	eventos para click en el Entorno	1	2		3
28	Interactividad	Vincular una línea a un Objeto	1	2		3
30	Interactividad	Ordenar Código en forma de Pseudo Objetos.	1	2		3
21	Objetos 3D	Crear una Línea.	1		3	3
8	UX ⁸	Crear Imágenes Representativas.	1		7	3
9	UX	definir la forma y colores del tema de la aplicación	1		5	3
32	Animación	crear animación para listas	2	2		1
38	Animación	Motor de Animación entre Tween.js y Director.js	2	1		1
11	UX	Desarrollar las vistas Estructuras de Datos	2		3	1
14	UX	Desarrollar vista Plantillas: listas pilas arboles	2	3		1
33	Animación	crear animación para Pilas	2	2		2
39	Animación	Encadenar Animaciones una después de Otra	2		6	2
24	Interactividad	Eventos para los botones de la interfaz interactiva	2		6	2
29	Interactividad	Vincular los 2 extremos de línea a 2 Objetos	2	3		2
7	UX	Optimizar Experiencia de Usuario.	2	2		2
31	Animación	eventos disparadores de animaciones	2		6	3
34	Animación	Crear animación para Arboles	2	3		3
40	Animación	Usar Recursividad y callbacks en JavaScript	2	2		3
4	Compatibilidad	Implementar Responsive Design en panel 3D.	2		4	3
25	Interactividad	Touch del mouse para dispositivos móviles	2		6	3
37	Animación	Desarrollar animaciones para Establecimiento de C.	3	3		1
12	UX	Desarrollar las vistas Redes	3		5	1
15	UX	Desarrollar vista Plantillas: secciones de Redes	3	2		1
35	Animación	Desarrollar animaciones para Transporte de P.	3	3		2
20	Objetos 3D	Importar Objetos de 3Ds Max mediante la API Three.js	3	1		2
36	Animación	Desarrollar animaciones para Redirección de P.	3	3		3
17	Objetos 3D	Crear un modelo 3D de una PC	3		3	3
18	Objetos 3D	Crear un modelo 3D de una Laptop	3		3	3
19	Objetos 3D	Crear un modelo 3D Router	3		2	3

Total 52,0 73,0
Total Días 61,1

Cuadro 7 Resumen Global del Release Backlog – Elaboración Propia.

⁸ UX User Experience – Experiencia de usuario

Para desarrollar es proyecto, se divide las tareas del **Release Backlog** en 3 Sprints.

El primer sprint es una etapa de desarrollo, donde básicamente ingreso en un proceso de aprendizaje de las funcionalidades que brinda el API para aprender a desarrollar en el navegador 3D.

Para esto se hace un sub-conjunto de tareas que requieren de menor conocimiento y que son más importantes para tener un resultado prematuro, para mostrar como avance en el desarrollo del proyecto. Más adelante explico su desarrollo en detalle.

El segundo sprint tiene un conjunto de tareas más avanzadas. Que reutilizan varios de los conceptos aprendidos del primer sprint, combinando diferentes secciones de código y abstrayendo en forma ordenada en funciones para hacer del código más manejable y reutilizable.

Con todo esto se logra desarrolla principalmente la sección del Estructuras de Datos: listas Pilas y Arboles.

También se logra hacer las animaciones gracias a la inversión de un buen tiempo investigando en este tema porque es muy complejo, pero gracias a esto se logra cumplir uno de los objetivos del proyecto.

El Tercer Sprint está dedicado al desarrollo de la sección de Redes, que contiene los temas de Redirección de Paquetes, establecimiento de Conexión y Transporte de Paquetes de red.

Que en su conjunto dan por concluido el desarrollo del proyecto.

4.7. Sprint 1.

Para este sprint se seleccionaron las siguientes tareas:

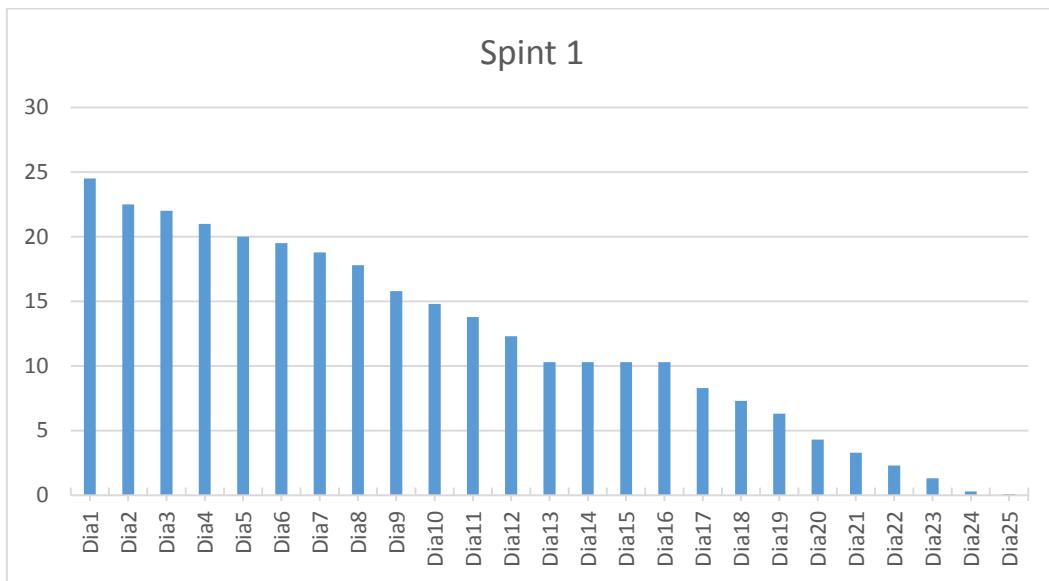
N	Grupo	Tarea	Sprint	D	H	Prior
1	Compatibilidad	Buscar una tecnología de alta compatibilidad	1	3		1
2	Compatibilidad	Desarrollar para el navegador Web	1		4	1
10	UX	Desarrollar la vista Principal.	1	1		1
6	Compatibilidad	Desarrollar del lado del Servidor.	1	2		2
13	UX	Desarrollar las vistas Dev (pruebas)	1		5	2
16	Objetos 3D	Crear un objeto en el Espacio.	1		3	2
26	Interactividad	Mover la Cámara en entorno al origen con Mouse(0,0,0)	1	1		2
27	Interactividad	Evento que arrastra un objeto sobre el espacio (Drag)	1	5		2
3	Compatibilidad	Implementar Responsive Design en Template.	1		2	3
5	Compatibilidad	Desarrollo del lado del Cliente.	1	3		3
8	UX	Crear Imágenes Representativas.	1		7	3
9	UX	definir la forma y colores del tema de la aplicación	1		5	3
21	Objetos 3D	Crear una Línea.	1		3	3
22	Interactividad	Eventos con el mouse para Línea.	1	1		3
23	Interactividad	eventos para click en el Entorno	1	2		3
28	Interactividad	Vincular una línea a un Objeto	1	2		3
30	Interactividad	Ordenar Código en forma de Pseudo Objetos.	1	2		3

Total	22,00	29,00
Total Días	25,63	

Cuadro 8 Resumen de Tareas del 1er Sprint - Elaboración Propia

4.7.1. Burndown Chart - Sprint 1.

Este es un gráfico que representa el progreso día a día del 1er Sprint.



Cuadro 9 Burndown Chart del Sprint 1 - Elaboración Propia

A continuación se describe el avance a detalle de intervalos de 5 días del avance de este sprint. Concluido a tiempo.

4.7.2. Buscar una tecnología de alta compatibilidad.

En parte del desarrollo que tiene 1er nivel de prioridad, se busca una la solución más adecuada para mostrar contenido 3D a las personas que quieran acceder a este contenido.

Ya que los usuarios de esta aplicación no tengan dificultades cuando accedan a este contenido y utilicen de manera directa, lo más rápido posible, se considera que es necesario que cumpla con los requisitos:

- Alta Compatibilidad en diferentes plataformas tanto de SO ⁹ como de Navegadores.
- Mínimos requisitos de software adicionales o Plug-Ins.
- Mínima configuración en el dispositivo del usuario.
- Rapidez de ejecución.

Básicamente Clic y usar.

Investigando existen algunas soluciones en internet... cada una con sus ventajas y desventajas.

En un Principio opte por una tecnología que se llama **x3D**, es muy interesante porque provee ciertas ventajas como importación de modelos 3D desde editores de 3D como 3DSMax y Archicad¹⁰, y crea escenas automáticamente donde uno puede interactuar navegando en torno a estos escenarios.

La desventaja de esta tecnología, es que no tiene bastante documentación para crear interacciones propias, y eventos por lo cual lo descarte.

⁹ SO Sistema Operativo

¹⁰ Archicad – Programa informático para arquitectura, Diseño, 3D.

En una segunda investigación encontré **Three.JS** que es una biblioteca desarrollada en JavaScript que brinda más funciones que la anterior sobre 2 tecnologías: WebGL y Canvas¹¹, lo cual la hace altamente compatible con diferentes.

Bueno pues toca aprender el uso de esta librería, navegando por internet uno encuentra ejemplos hechos y la comunidad constantemente está actualizando soluciones particulares a las necesidades de los desarrolladores 3D alrededor del mundo, pero nada mejor que un curso para entenderlo correctamente.

Por lo que me inscribí a un curso en línea en Udacity:

Interactive 3D Graphics Course With Three.js & WebGL - Udacity



<https://www.udacity.com/course/cs291>

Este curso es dictado por:

Eric Haines



Una Descripción de su biografía se puede encontrar en

(Udacity, 2013) Eric Haines is a Senior Principal Engineer at Autodesk, Inc., working on a next-generation interactive rendering system for computeraided design applications. He is a coauthor of the book “Real-Time Rendering”, a founder and editor of the Journal of Computer Graphics Techniques, and maintainer of the Graphics Gems code repository, among other activities. He received an MS from the Program of Computer Graphics at Cornell in 1985.

De este curso aprendí una mayoría de los conceptos necesarios para comenzar a desarrollar en con esta tecnología.

En un principio introduce en los conceptos de renderizado Interactivo:

¹¹ Canvas – es un lienzo para dibujar sobre HTML

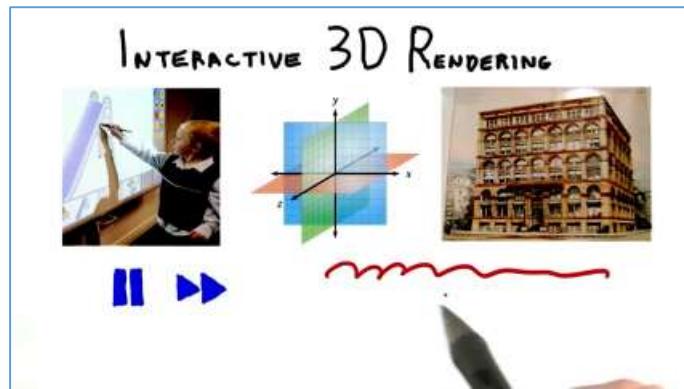


Figura 13 Renderizado Interactivo Fuente: Curso En Línea 3D Udacity

Otro concepto importante es la manera en que se renderiza un cuadro a partir de la posición de una cámara y la necesidad de definir un campo de visión para optimizar el uso del procesador del computador como se muestra en la siguiente Figura:

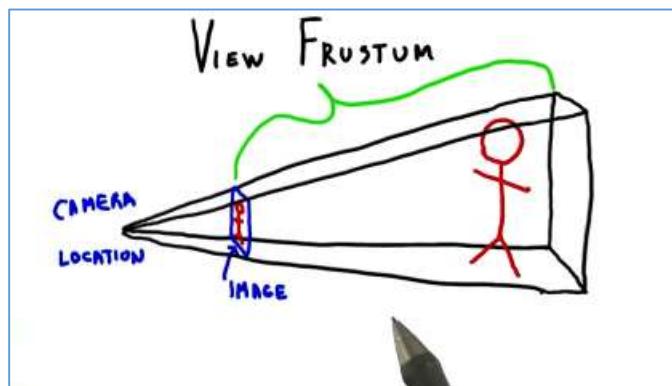


Figura 14 Cámara y renderizado Fuente: Curso En Línea 3D Udacity

El uso de Puntos y vectores para determinar la dirección y posición de cualquier objeto en el espacio.

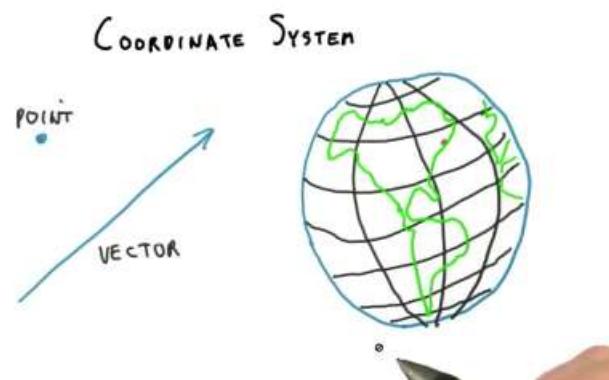


Figura 15 Puntos y Vectores Fuente: Curso En Línea 3D Udacity

Importante el uso de la notación Mando Derecha para comprender el sistema de coordenadas espacial.

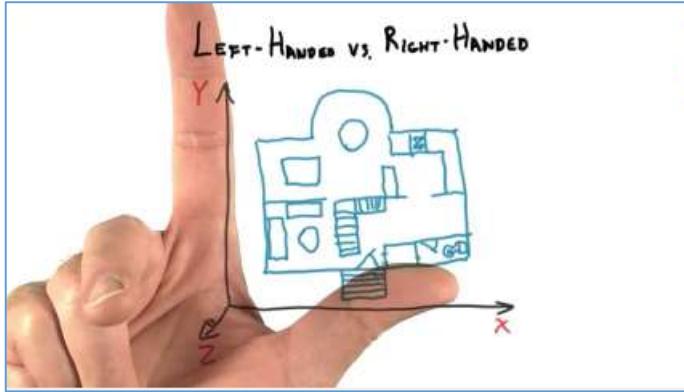


Figura 16 Disposición Sistema de Coordenadas Notación Mano Derecha Fuente: Curso En Línea 3D Udacity

Estos y otros conceptos generales son muy útiles para comprender lo que uno hace al desarrollar en 3D sobre el navegador.

4.7.3. Desarrollar para el Navegador Web.

Desarrollar para la web tiene la ventaja de que el contenido puede ser consumido por los usuarios casi en cualquier lugar y momento.

Las tecnologías necesarias para hacer realidad esto son

- PHP, del lado del servidor, porque es un lenguaje común entre los servidores.
- JavaScript, del lado del cliente.
- HTML5, como estándar para usar las nuevas características en los navegadores que más se acomodan a este estándar.
- Css3, para maquetar el contenido esto ayuda mucho en la Experiencia del Usuario.

Seleccionadas estas tecnologías para desarrollar el proyecto.

4.7.4. Desarrollar del lado del Servidor.

Para desarrollar el software se implementa en un servidor, para el caso se usa como lenguaje base PHP como se mencionó anteriormente, para hacer más estándar el proyecto, si se da continuidad a este proyecto pueda ser entendido por otros desarrolladores, se usa un Framework Codeigniter.



Figura 17 Codeigniter Logo

Este entorno de trabajo proporciona la infraestructura adecuada para desarrollar el proyecto porque es ligero, maneja URLs ¹² amigables, sigue el patrón MVC¹³ y tiene una curva de aprendizaje muy corta.

En el desarrollo del proyecto contribuye en la parte que va por detrás que no se ve para que la aplicación esté ordenada y organizada cuando el usuario visita la dirección de la aplicación web.

4.7.5. Desarrollar la vista principal.

La vista principal es el código html que forma parte del modelo vista controlador, donde se muestra la cara de la presentación.

Para esto se desarrolla para la parte del cliente la siguiente Interfaz.

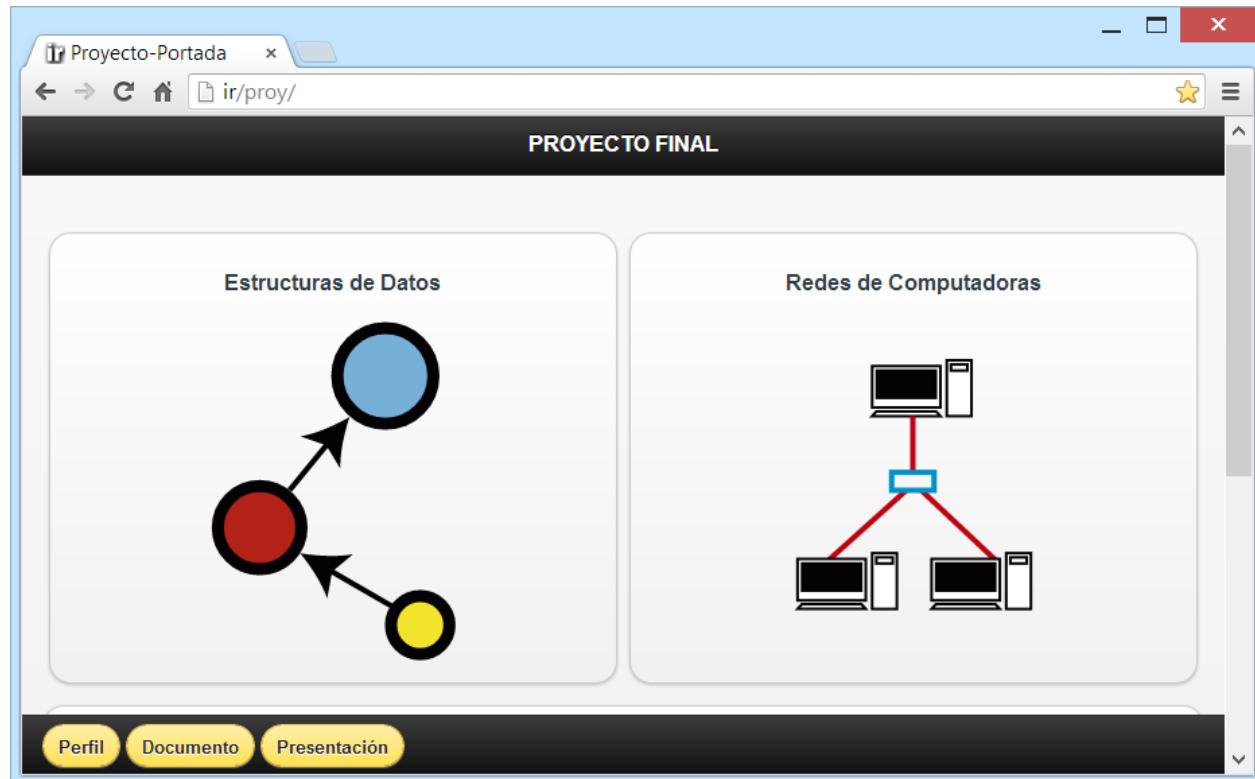


Figura 18. Vista Principal de la Aplicación- Elaboración Propia

4.7.6. Eventos para click en el Entorno.

Pretendiendo hacer interactivo el contenido, de las diferentes secciones de la aplicación se añade la funcionalidad de poder visualizar el Entorno distintos ángulos girando.

La capacidad de girar el entorno está detallada en **Mover la Cámara en entorno al origen (0,0,0) con el Mouse**.

4.7.7. Desarrollar las vistas Dev (pruebas).

Esta es la sección donde se centra el desarrollo en este sprint. El botón para acceder a este contenido es la siguiente imagen.

¹² URLs son las direcciones que enlazan a una página

¹³ MVC Model View Controller

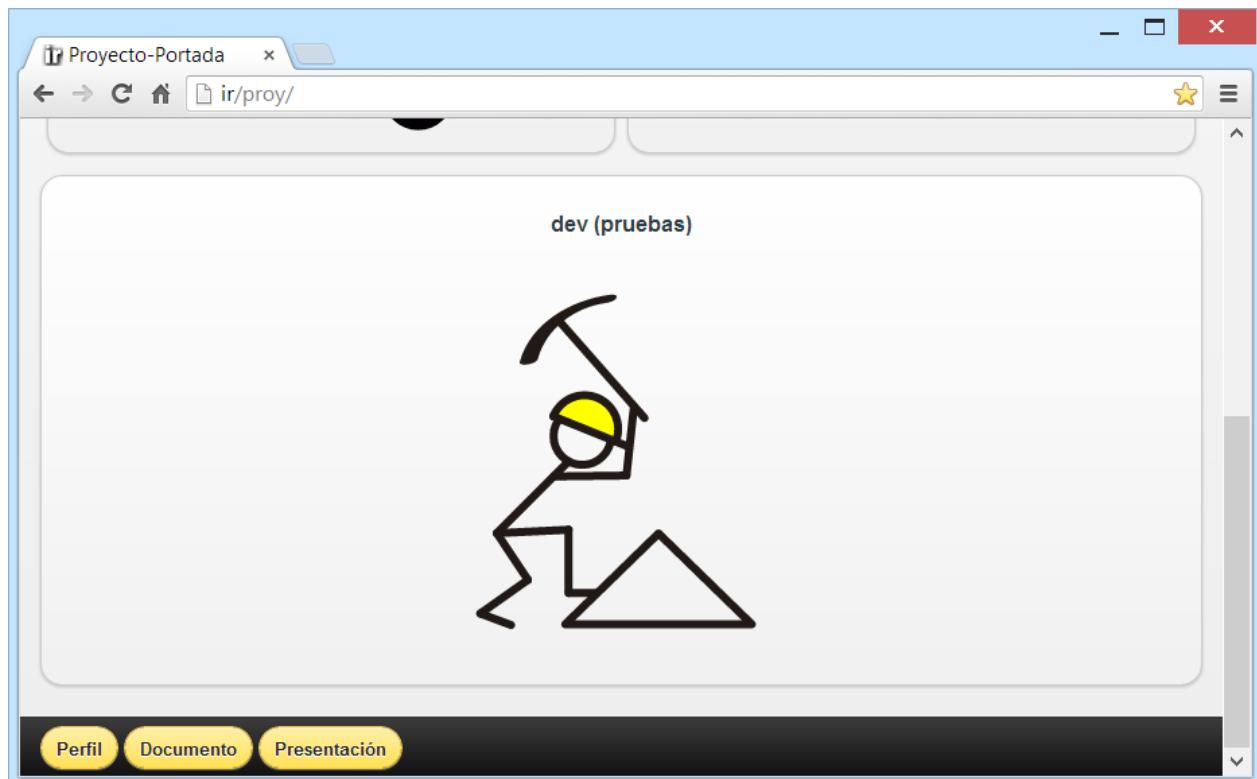


Figura 19 link para la sección Dev(Pruebas) - Elaboración Propia

A demás esta sección muestra el siguiente que son las sub-secciones de Dev(Pruebas) todo desarrollado en la parte del servidor con Codeigniter, y los elementos de interfaz son desarrollados con JQueryMobile, del se tiene una explicación más adelante.

Estas es la captura de la subsección Dev Pruebas.

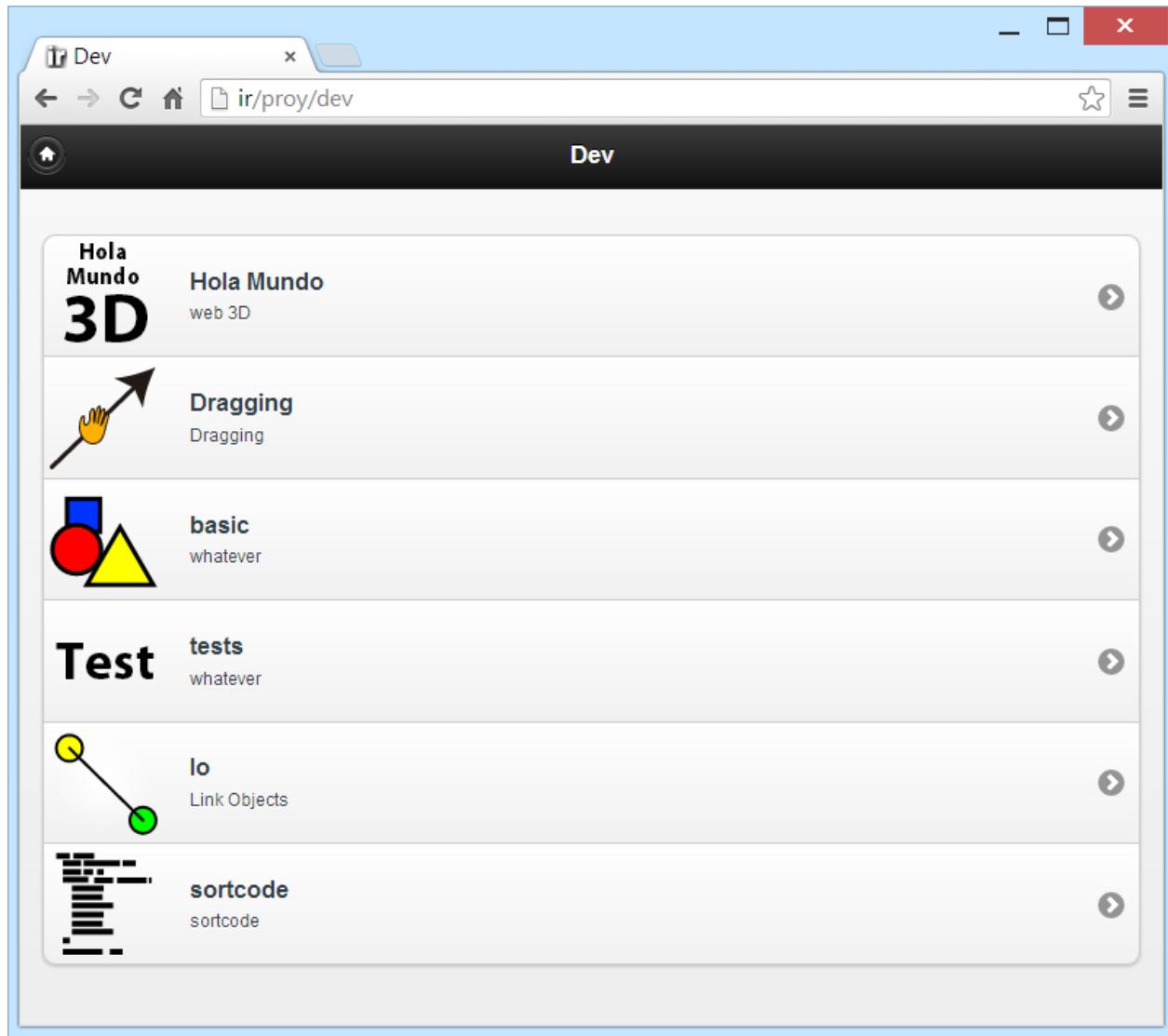


Figura 20 Portada Sección Dev (Pruebas) – Elaboración Propia

4.7.8. Crear un objeto en el Espacio.

Esto es para el ejemplo más básico desarrollando para el navegador un objeto 3D.

Hago un listado a continuación los elementos que se necesitan.

- Cámara
- Escena
- Malla (Compuesta de una Geometría envuelta por un Material)
- Renderizado (WebGL).
- Una función para animar.

El código es el siguiente.

```

23 <script>
24   var camera, scene, renderer;
25   var geometry, material, mesh;
26   init();
27   animate();
28   function init() {
29     camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 1, 1000 );
30     camera.position.z = 1000;
31     scene = new THREE.Scene();
32     geometry = new THREE.CubeGeometry( 200, 200, 200 );
33     material = new THREE.MeshBasicMaterial( { color: 0xffff00, wireframe: true } );
34     mesh = new THREE.Mesh( geometry, material );
35     scene.add( mesh );
36     renderer = new THREE.CanvasRenderer();
37     renderer.setSize( window.innerWidth, window.innerHeight );
38     document.body.appendChild( renderer.domElement );
39   }
40   function animate() {
41     // note: three.js includes requestAnimationFrame shim
42     requestAnimationFrame( animate );
43     mesh.rotation.x += 0.01;
44     mesh.rotation.y += 0.02;
45     renderer.render( scene, camera );
46   }
47 </script>

```

Figura 21 – Código de Ejemplo Básico Three.js – Elaboración Propia

Y el resultado es:

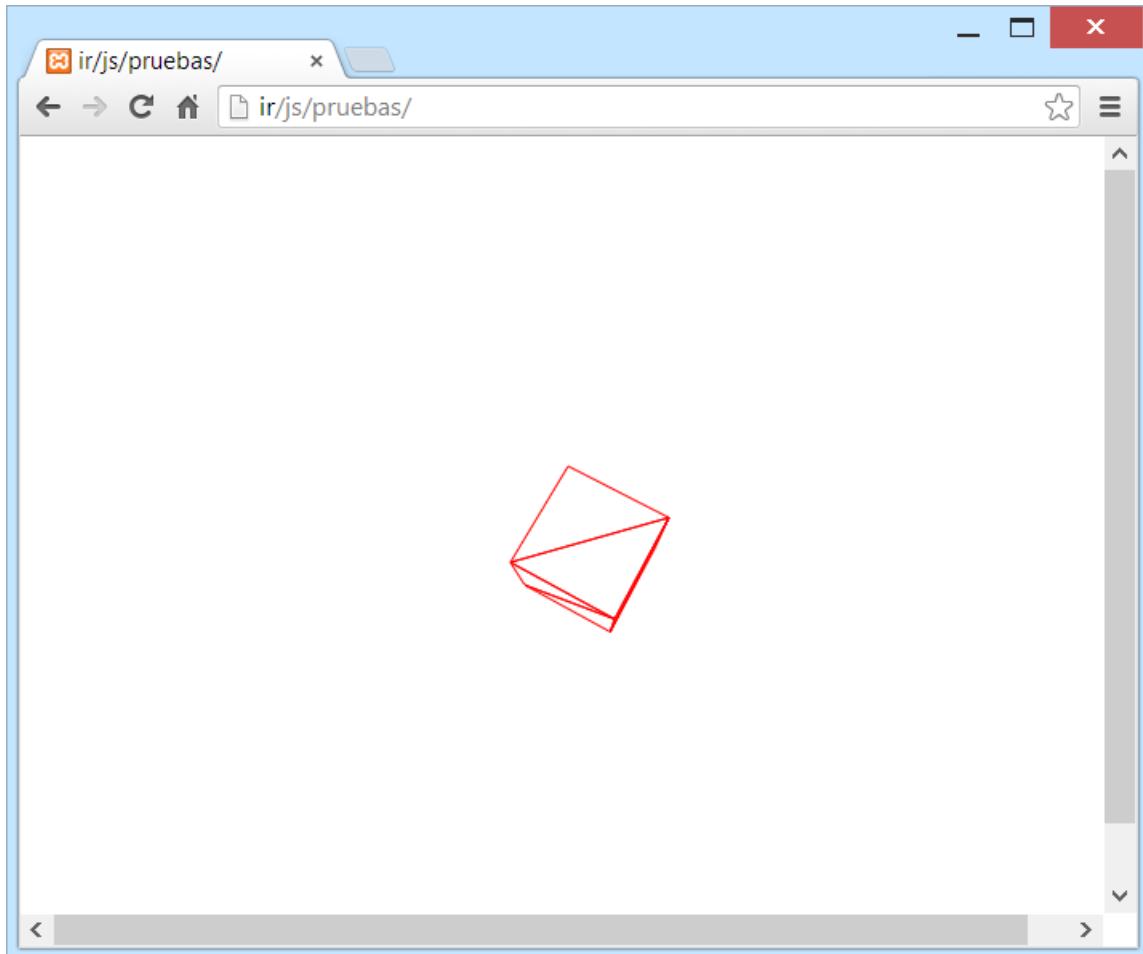


Figura 22 Ejemplo de Cuadrado Girando - Three.js - Elaboración Propia

4.7.9. Mover la Cámara en entorno al origen (0,0,0) con el Mouse

Como se ve en esta secuencia de imágenes le lograr que la escena se mueva es en realidad un efecto causado por mover la posición de la cámara.

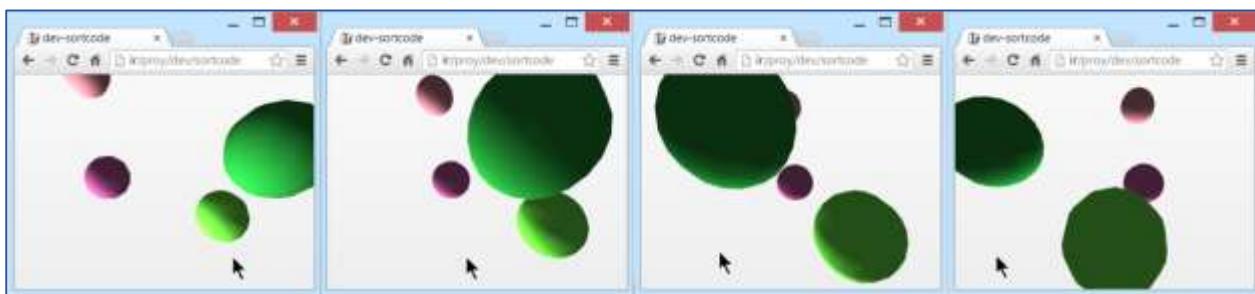


Figura 23 secuencia de movimiento de la cámara - Elaboración Propia

Esto se hacía con fórmulas complejas de Euler hasta que desarrollaron un pedazo de código JavaScript que se encarga de hacer esta operación muy sencilla e incluso es configurable en ciertos parámetros.

El script encargado de esta funcionalidad se llama TrackballControls.js y su uso se reduce a:

```
93  var cargarControles = function(area){  
94    controls = new THREE.TrackballControls( camera, renderer.domElement );  
95  
96    controls.rotateSpeed = 1.5;  
97    controls.zoomSpeed = 1.2;  
98    controls.panSpeed = 0.8;  
99    controls.noRotate = false;  
100   controls.noZoom = false;  
101   controls.noPan = false;  
102   controls.staticMoving = true;  
103   controls.dynamicDampingFactor = 0.3;  
104 }  
105
```

Figura 24 Uso de TrackballControls.js - Elaboración Propia.

4.7.10. Evento que arrastra un objeto sobre el espacio (Drag).

Esta es una tarea un poco complicada, no está implementada como una función nativa entre las funciones de three.js, pero gracias al curso en línea pude comprender una función que se encarga de este proceso y modificarla para que se adapte a lo que necesito.

Primero se necesita organizar la sucesión de eventos que implica esto:

- ¿Cuándo comienzo a mover un objeto por el espacio?
- ¿Cuándo se está moviendo un objeto que ocurre?
- ¿Cuánto termino de mover un objeto?

Para empezar esto encaja en una sucesión de eventos, para responder a la primera interrogante se mueve un objeto cuando:

Se **presiona click** sin soltar sobre el objeto que se desea mover para verificar esto hay 3 situaciones:

Se presionó **sobre el espacio vacío**. En este caso no se realiza ninguna acción de mover objetos, pero si se mueve la escena (en realidad la cámara en torno al origen 0,0,0 explicado en el subtítulo anterior)

Se presionó **sobre un objeto único**. El objeto en cuestión es marcado como **SELECTED** y hasta que se suelte el click adquiere la posición cambiante del mouse.

Se presionó **sobre varios objetos** (conjunto de objetos uno sobre otro). En este Caso se traza un as de línea invisible entre la cámara y la posición del click, este haz recorre el espacio desde la cámara en dirección hacia el click y se selecciona el que esté más cerca de la cámara luego se sigue el proceso cuando se presionó sobre un objeto

Este es un gráfico de lo que se hace en realidad:

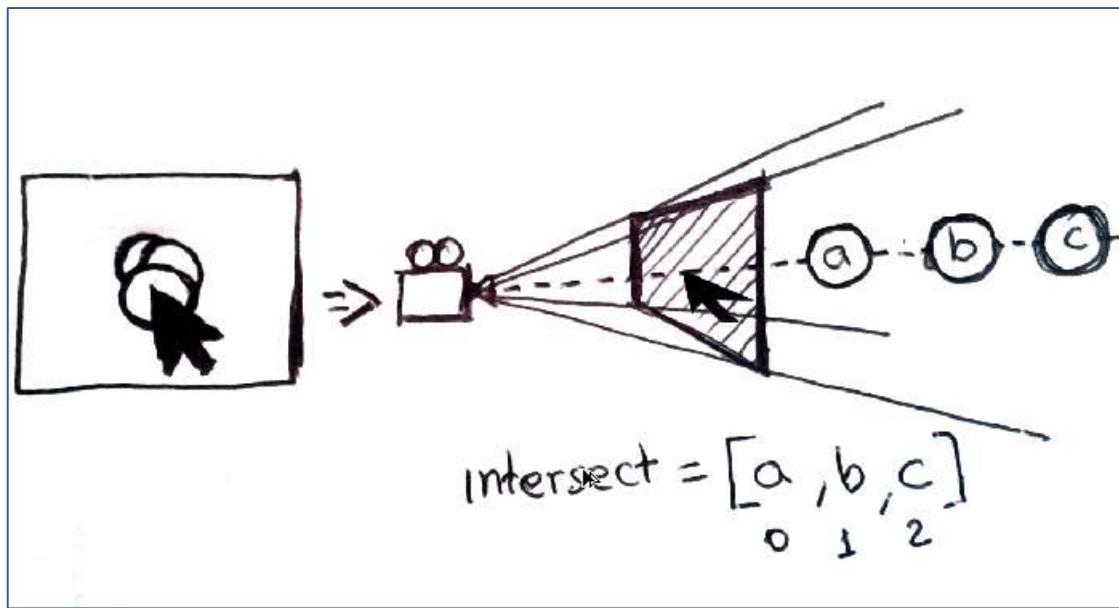


Figura 25 intersección entre un haz que parte de la cámara y varios objetos - Elaboración Propia.

El código que implementa esta lógica para guardar en **SELECTED** el objeto que está más cerca de la cámara.

```

192  var onDocumentMouseDown = Function(event) {
193    mouseDown = true;
194
195    event.preventDefault();
196    var vector = new THREE.Vector3( mouse.x, mouse.y, 0.5 );
197    projector.unprojectVector( vector, camera );
198    var raycaster = new THREE.Raycaster( camera.position, vector.sub( camera.position ).normalize() );
199    var intersects = raycaster.intersectObjects( objects );
200    if ( intersects.length > 0 ) {
201      controls.enabled = false;
202      SELECTED = intersects[ 0 ].object;
203      var intersects = raycaster.intersectObject( plane );
204      offset.copy( intersects[ 0 ].point ).sub( plane.position );
205      container.style.cursor = 'move';
206    }
207  }
  
```

Figura 26 al presionar click - Elaboración Propia.

Una vez que obtenemos el objeto que presionamos con click en SELECTED, ahora **cada vez que se mueva el cursor**, actualizamos la posición del objeto sea igual a la posición del cursor mientras está presionado.

¿Cómo hacemos esto? Bueno en el evento encargado de escuchar cuando el mouse esta se está moviendo se llama onDocumentMouseMove y se encarga de entre otras cosas mover el objeto SELECTED mientras se está presionado y mueve el cursor:

```

209  var onDocumentMouseMove = function() {
210    mouseDown? mouseMove = true : mouseMove = false;
211    event.preventDefault();
212    camera.updateProjectionMatrix();
213    npoint = new THREE.Vector3(camera.position.x,camera.position.y,camera.position.z);
214    mouse.x = ( event.clientX / window.innerWidth ) * 2 - 1;
215    mouse.y = - ( event.clientY / window.innerHeight ) * 2 + 1;
216    var vector = new THREE.Vector3( mouse.x, mouse.y, 0.5 );
217    projector.unprojectVector( vector, camera );
218    var raycaster = new THREE.Raycaster( camera.position, vector.sub( camera.position ).normalize() );
219    if ( SELECTED ) {
220      var intersects = raycaster.intersectObject( plane );
221      SELECTED.position.copy( intersects[ 0 ].point.sub( offset ) );
222
223      if (SELECTED.lineas) { ... }
224    }
225    return;
226  }
227
228  var intersects = raycaster.intersectObjects( objects );
229  if ( intersects.length > 0 ) {
230    if ( INTERSECTED != intersects[ 0 ].object ) {
231      if ( INTERSECTED ) INTERSECTED.material.color.setHex( INTERSECTED.currentHex );
232      INTERSECTED = intersects[ 0 ].object;
233      INTERSECTED.currentHex = INTERSECTED.material.color.getHex();
234      plane.position.copy( INTERSECTED.position );
235      plane.lookAt( npoint.add(INTERSECTED.position) );
236    }
237    container.style.cursor = 'move';
238  } else {
239  }
240
241  plane2.lookAt(npoint); //hacemos el el plano Mire de fre
242
243  if (aura1) { ... }
244
245  };
246
247
248
249
250
251
252
253
254
255
256
257
  }

```

Es importante saber que es necesaria una conversión de las coordenadas del monito a las coordenadas del espacio 3D, esto se hace mediante la función **Projector** de Three.js una función muy útil que además hace la operación inversa desconversión, a continuación un esquema de que es lo que se hace:

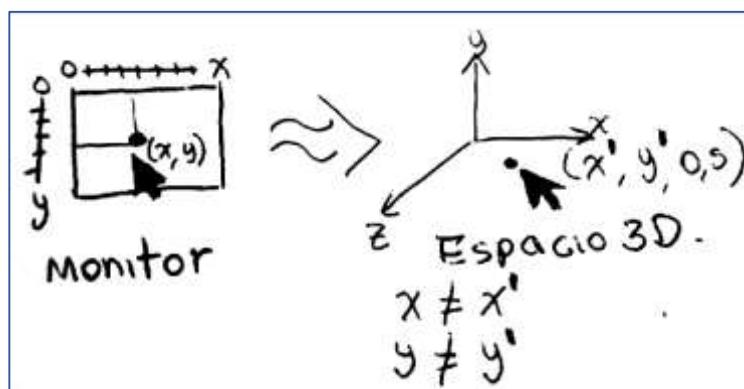


Figura 27 Transformación del coordenadas 2D a 3D - Elaboración Propia

Con todo esto desarrollado es posible realizar la operación de **Mover objetos** a lo largo de la pantalla con el mouse o con el dedo en caso de pantallas táctiles.

4.7.11. Implementar Responsive Design en Template.

La aplicación web tiene que poder ver de la misma manera en cualquier dispositivo, para esto no importa que dimensiones tenga la pantalla esto afortunadamente lo hace JqueryMobile que se explica en el siguiente subtítulo, esta es una captura de la misma página del proyecto en distintos tamaños de monitor.

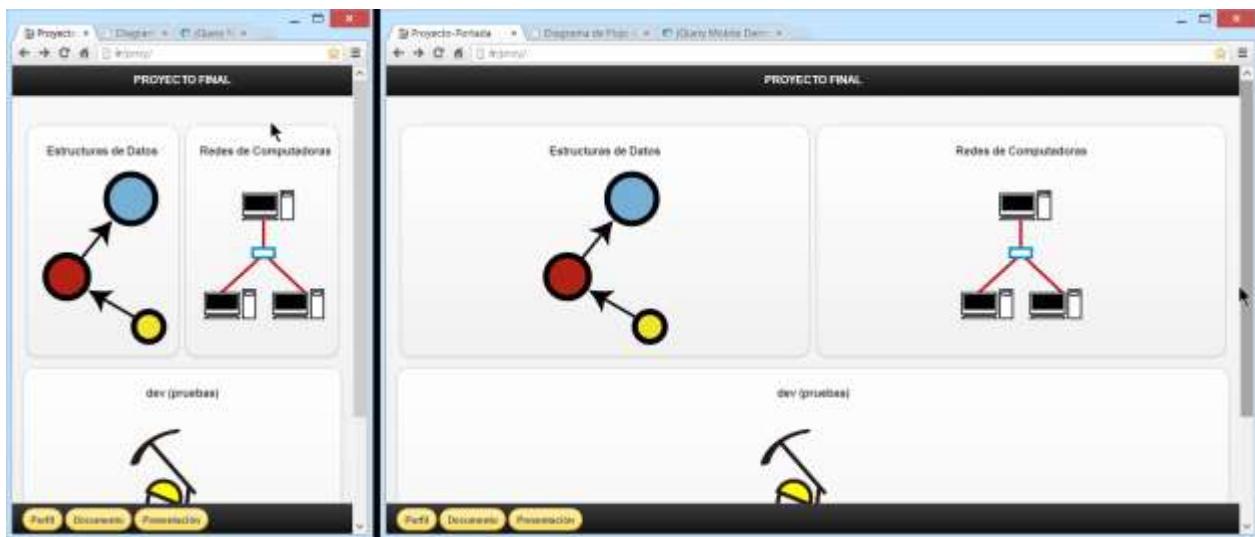


Figura 28. Diferentes tamaños de la misma portada comparadas lado a lado - Elaboración Propia.

Como se puede apreciar el contenido se acomoda fluidamente al tamaño de la pantalla esto es muy útil para presentar una misma presentación en un solo vistazo.

4.7.12. Desarrollo del lado del Cliente.

En el lado del cliente es donde más trabaja la aplicación ya que es en esta parte donde se renderiza la aplicación, para esto se tiene que tener claro el concepto de Modelo Vista Controlador según su definición en (Codeigniter, 2013) “que permite una gran separación entre la lógica y la presentación. Es particularmente bueno para proyectos en los que los diseñadores trabajan en sus archivos de plantillas, ya que el código en estos archivos será mínimo.”

- El Modelo representa sus estructuras de datos. Típicamente sus clases del modelo contendrán funciones que los ayudarán a devolver, insertar y actualizar información de su base de datos.
- La Vista es la información que se presenta al usuario. Una vista será normalmente una página web, pero en Codeigniter, una vista también puede ser un fragmento de página como el encabezado o pie de página. También puede ser una página RSS, o cualquier otro tipo de "página".
- El Controlador sirve como un intermediario entre el Modelo, la Vista y cualquier otro recurso necesario para procesar la solicitud HTTP y generar una página web.

El siguiente gráfico ilustra como los datos fluyen a través del sistema:

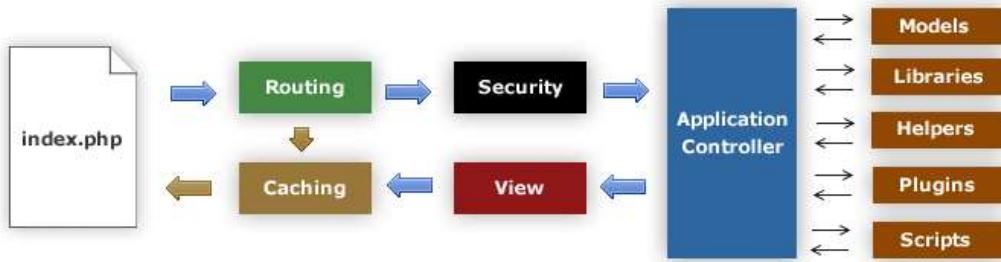


Figura 29 Flujo de datos a Través del sistema - Fuente Manual CodeIgniter

En la parte de la vista es donde se trabaja, bastante con una biblioteca dedicada a dispositivos móviles: JqueryMobile. Que se encarga de hacer que el contenido se adapte a distintos tamaños de pantalla.



Figura 30 Logo JqueryMobile

Se encarga de manejar las plantillas y los elementos de interfaz como botones links y efectos de transición entre contenidos de las diferentes páginas del proyecto, y principalmente hace compatible el contenido en dispositivos móviles.

4.7.13. Crear Imágenes Representativas.

Las imágenes que se usan en las diferentes secciones de la presente aplicación web son imágenes “especiales” porque en si son imágenes vectoriales, la ventaja que esto representa es que no importa cuán grande se amplíe el zoom del navegador, estas imágenes no pierden calidad.

Otra ventaja muy importante es que su peso promedia los 4KB de espacio en disco para este proyecto, eso es excelente para la velocidad de respuesta al cargarse la página en el navegador del cliente.

Para crear estas imágenes vectoriales se usó Adobe Illustrator. Y se las exportó en un formato muy ligero, claro hablo de SVG que actualmente son compatibles en los navegadores... desde hace ya buen tiempo.

Estas son algunas de las imágenes y el proceso de su creación.

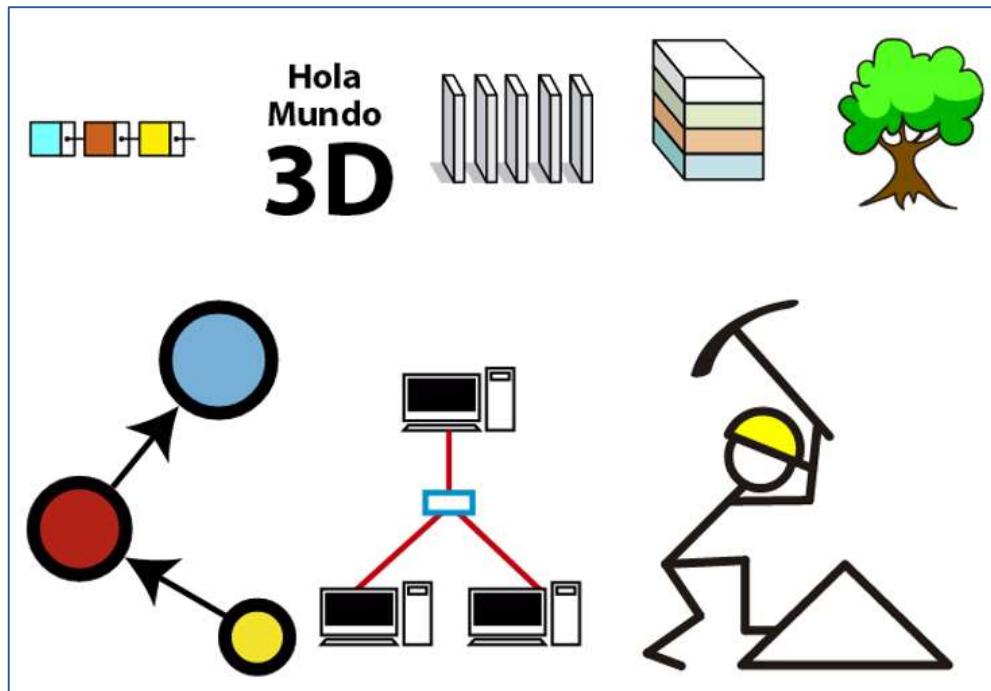


Figura 31 Ilustraciones Vectoriales SVG - Elaboración Propia.

También muestro el proceso de creación, que a pesar de ser sencillo consume tiempo.

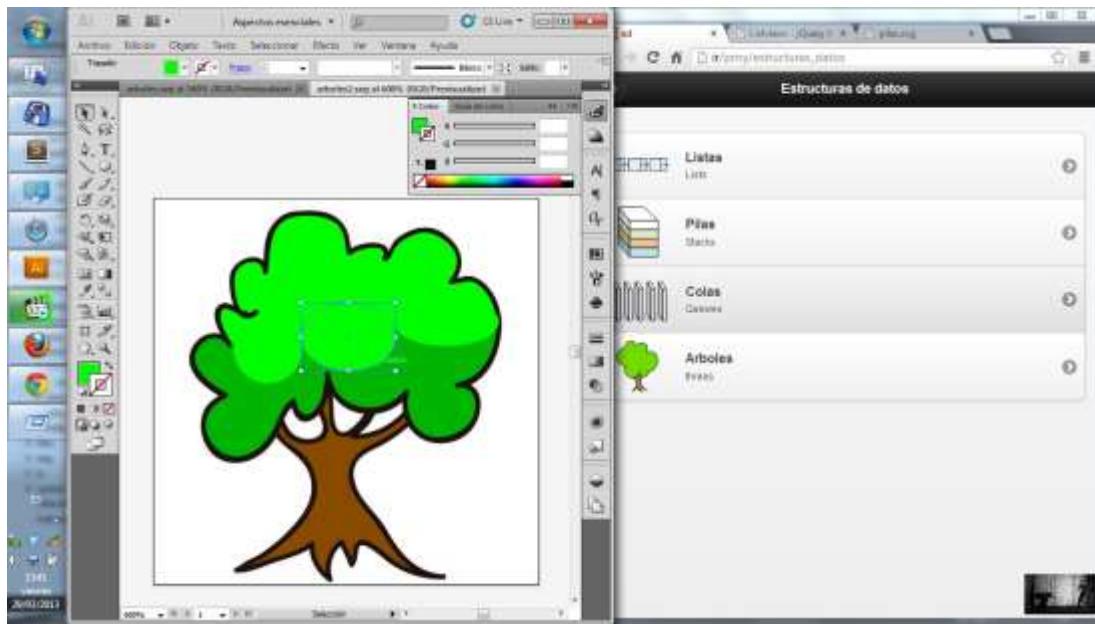


Figura 32 Proceso de creación de las imágenes SVG para el Proyecto - Elaboración Propia.

4.7.14. Definir la forma y colores del tema de la aplicación.

La aplicación web tiene en esencia un objetivo implícito, que es **llamar la atención del usuario** y ya que el contenido se ve en un monitor, normalmente se logra esto utilizando colores llamativos, para las secciones que presentan contenidos como botones y demás.

Hay que tener cuidado con esto porque al usar excesivos colores puestos por donde sea, puede causar un efecto contraproducente.

Entonces siendo cuidadoso con este punto, se definen 2 colores en el tema principal:

- Negro
- Gris Claro casi plateado.

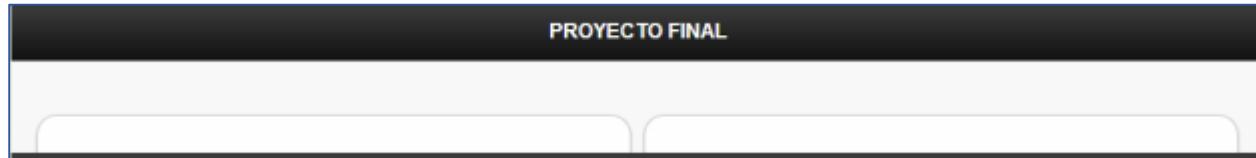


Figura 33 Colores del tema principal. - Elaboración Propia.

Y para que algunos botones no se pierdan o sean imperceptibles, se utiliza 2 colores más el azul y el amarillo. Luego el resto de los botones importantes tienen **imágenes representativas** en su interior que son coloridos.

Los ejemplos, también forman parte importante del proceso de llamar la atención, por lo que al dibujar los elementos se tienen bastante colorido, los ejemplos para esto se pueden apreciar más adelante en su Sprint respectivo.

4.7.15. Crear una Línea.

Para crear líneas en el espacio 3D hay que conocer algo más, el uso de **THREE.Geometry** que básicamente es una estructura que contiene puntos.

En el caso particular de una línea solamente se necesitan 2 puntos. Estos puntos tienen que ser instancias de la clase **THREE.Vector3**, que representa un punto con coordenadas (x,y,z).

```
41
42
43
44     var lineMat = new THREE.LineBasicMaterial( { color: 0x0000ff, opacity: 1, linewidth: 3 } );
45     var geometry = new THREE.Geometry();
46
47     geometry.vertices.push(new THREE.Vector3(0, 0, 0));
48     geometry.vertices.push(new THREE.Vector3(20, 100, 55));
49
50     geometry.verticesNeedUpdate=true;
51
52     line = new THREE.Line(geometry, lineMat, parameters = { linewidth: 400 });
53     scene.add(line);
54
55
56
```

Figura 34 Crear una línea en base a 2 puntos - Elaboración Propia.

El resultado es básicamente una línea en el espacio.

Para optimizar la creación de líneas, ya que se hará más recurrente su uso, creé una función que se encarga de esto.

El código de estas funciones es:

```
431
432
433 function create_line(v1,v2,color){
434     var lineMat = new THREE.LineBasicMaterial( { color: color, opacity: 1, linewidth: 3 } );
435     var geom = new THREE.Geometry();
436     geom.vertices.push( v1 ); geom.vertices.push( v2 );
437     linea = new THREE.Line(geom, lineMat);
438     return linea;
439 }
440
441
```

Figura 35 Función que crea una línea - Elaboración Propia.

4.7.16. Eventos con el mouse para Línea.

La línea en si no tiene eventos asociados a si mismo directamente, sino que es través de la herencia que recibe cuando se vincula un objeto

Explico con detalle el uso de estos eventos heredado más adelante en **Vincular una linea a un objeto**. En este mismo capítulo.

4.7.17. Vincular una línea a un Objeto.

Esta es una funcionalidad de las líneas que se aplica bastante en la interconexión de objetos.

Se entiende por objeto a la **figura 3D** o modelo 3D siendo más técnico, que está dentro de la escena que tiene la capacidad de moverse cuando el mouse lo esté arrastrando.

Cuando el objeto se mueve la **línea** asociada a este tiene que actualizar la posición del **extremo asociado** al objeto.

Los pasos necesarios para hacer que una línea se vincule a un objeto se ven a continuación.

Crear la línea

Crear el objeto

Hacer que uno de los extremos de la línea se asocie a la posición del objeto.

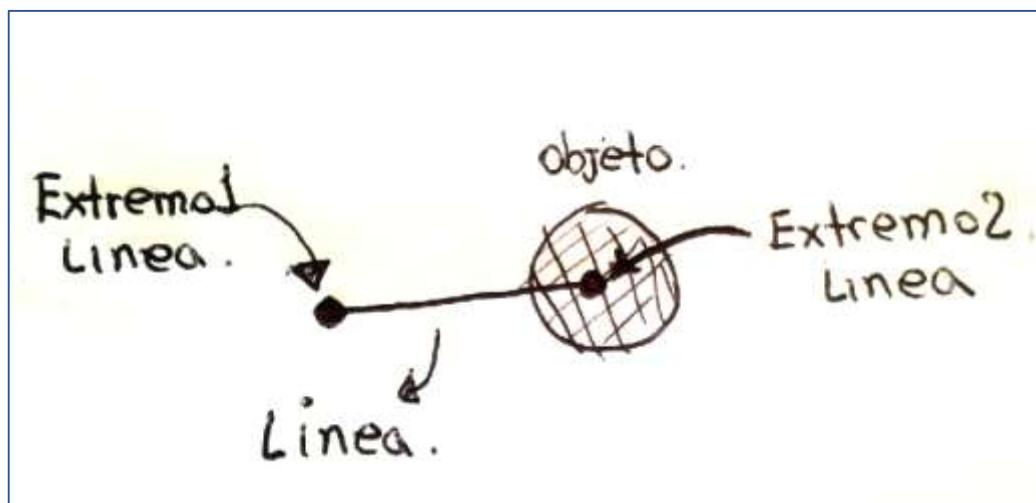


Figura 36 Objeto con un extremo de una línea asociado a el - Elaboración Propia.

Una vez asociado el extremo al objeto, es necesario implementar en el evento que cuando se mueve el objeto se mueva la posición del extremo de la línea, para que se muevan juntos.

Se puede observar el ejemplo un de esta implementación, en la sección dev (Pruebas).

Esto en código se traducen en:

```
135  function onDocumentMouseMove( event ) {
136    event.preventDefault();
137    camera.updateProjectionMatrix();
138    npoint = new THREE.Vector3(camera.position.x,camera.position.y,camera.position.z);
139    mouse.x = ( event.clientX / window.innerWidth ) * 2 - 1;
140    mouse.y = - ( event.clientY / window.innerHeight ) * 2 + 1;
141    var vector = new THREE.Vector3( mouse.x, mouse.y, 0.5 );
142    projector.unprojectVector( vector, camera );
143    var raycaster = new THREE.Raycaster( camera.position, vector.sub( camera.position ).normalize() );
144
145    if ( SELECTED ) {
146      var intersects = raycaster.intersectObject( plane );
147      SELECTED.position.copy( intersects[ 0 ].point.sub( offset ) );
148      puntos = line.geometry.vertices;
149
150      line.geometry.vertices[ '1' ].copy(SELECTED.position);
151      line.geometry.verticesNeedUpdate=true;
152      return;
153    }
154
155    var intersects = raycaster.intersectObjects( objects );
156    if ( intersects.length > 0 ) {
157
158    } else {
159    }
160  }
161
162
163
164
165
166
167
168
169
170
171
172
```

Describiendo un poco: se tiene la línea **line** creada previamente a la cual se le extrae uno de sus extremos el de la posición [“1”] y se le asigna la posición del elemento actualmente seleccionado **SELECTED**.

Luego se obliga a la línea a actualizarse para que los cambios se grafiquen al resultado final en el navegador.

El resultado:

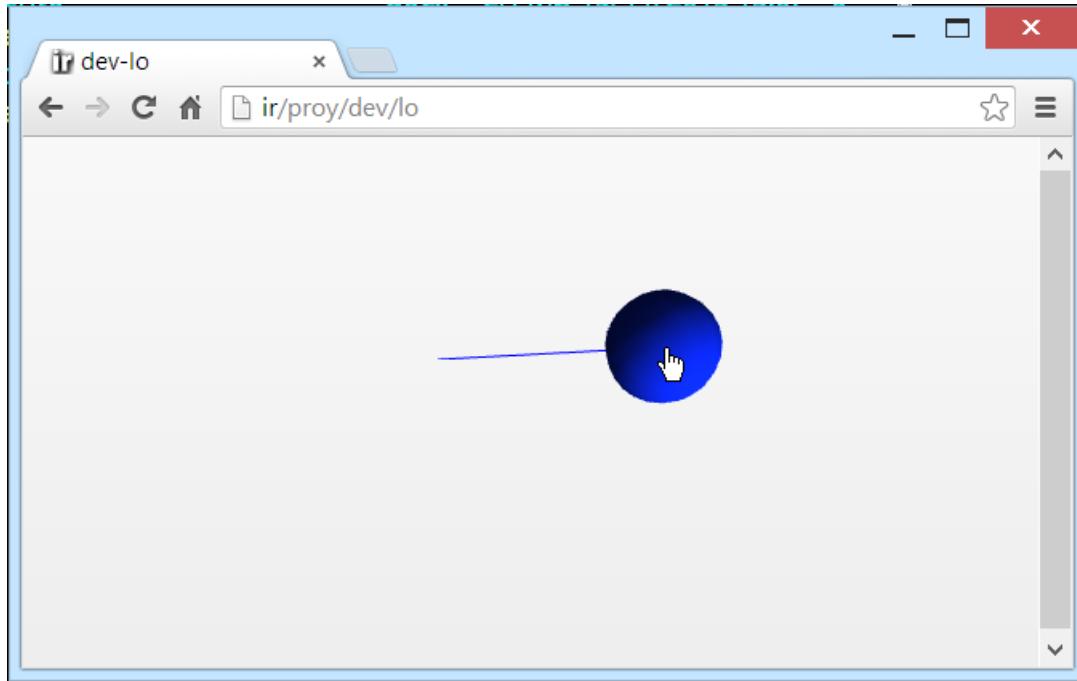


Figura 37 Lina con uno de sus extremos vinculado a un Objeto - Elaboración Propia.

4.7.18. Ordenar Código en forma de Orientado a Objetos.

Hasta esta etapa del aprendizaje de **Three.js**, el resultado de todo lo aprendido está plasmado en la sección dev(Pruebas) donde se puede ver ejemplos de creación básica de objetos, vinculación, movimiento, etc.

Pero estos ejemplos están desarrollados sin seguir una estructura hablando del código, ya que no está organizado en funciones que se puedan reutilizar ni nada de eso.

Para que el código sea mantenible, organizado, entendible y reutilizable, sigo el paradigma orientado a objetos.

Como en JavaScript no existen clases... uso una artimaña para poder emular esta forma de desarrollar. Ahora el código está organizado de la siguiente manera:

```
1  /*----- MAIN -----*/
2
3  //console.log("--->"); MAIN
4
5  var obj = new escenario("areasortcode");
6
7  /*----- ESCENARIO y OBJETOS -----*/
8
9
10 function escenario(div){
11     var camera;
12     var container;
13     var controls;
14     var projector;
15     var renderer;
16     var scene;
17     var light;
18     var npoint,INTERSECTED, SELECTED, plane;
19
20     var objects=[];
21
22     var mouse = new THREE.Vector2(), ...
23
24     var offset = new THREE.Vector3();
25
26     _construct = function(area) { ... }
27
28     var cargarEstilos = function(){ ... }
29
30     var cargarEscena = function(area){ ... }
31
32     var cargarControles = function(area){ ... }
33
34     var cargarEventos = function(){ ... }
35
36     var cargarLuces = function () { ... }
37
38     var cargarLogistica = function () { ... }
39
40     var onWindowResize = function() { ... }
41
42     var onDocumentMouseDown = function(event) { ... }
43
44     var onDocumentMouseMove = function() { ... }
45
46     var onDocumentMouseUp = function() { ... }
47
48     var onEfectiveClick=function(){ ... }
49
50     var animate = function() { ... }
51
52     var render = function() { ... }
53
54     this.crearEsfera = function(){ ... }
55
56     this.getCamera = function() { ... }
57
58     this.getRenderer = function() { ... }
59
60     _construct(div);
61 }
```

Figura 38 Mejora en la organización del código - Elaboración Propia.

Con esta forma de organizar el código, ahora es más sencillo escalar esto para proyectos más grandes de forma organizada, atendiendo de forma más precisa errores que pueden surgir, al momento de desarrollar y además es más Comprensible para que otros desarrolladores puedan dar continuidad al proyecto.

4.8. Sprint 2.

Para este sprint se seleccionaron las siguientes tareas:

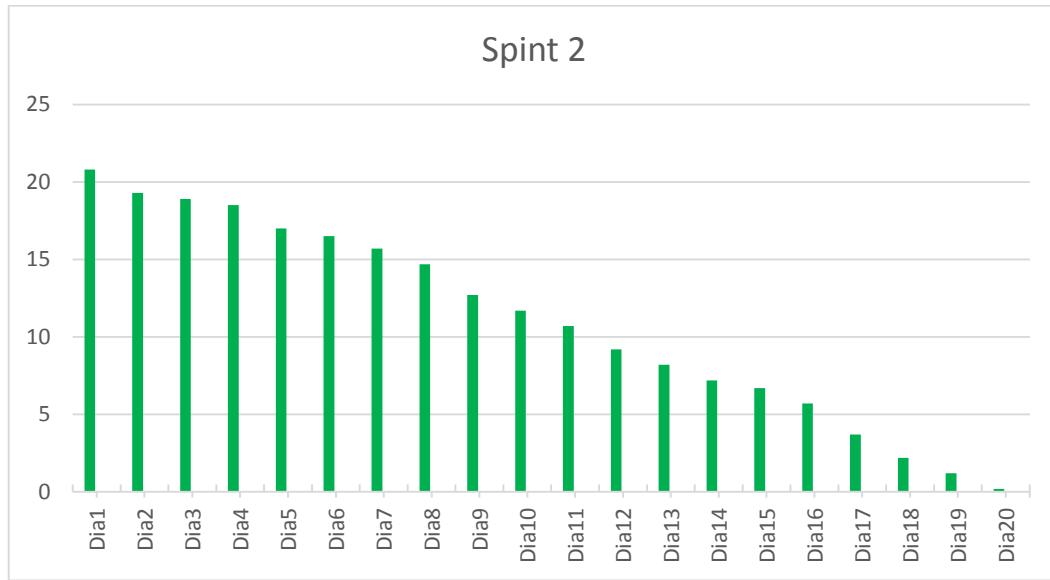
N	Grupo	Tarea	Sprint	D	H	Prior
11	UX	Desarrollar las vistas Estructuras de Datos	2		3	1
14	UX	Desarrollar vista Plantillas: listas pilas arboles	2	3		1
38	Animación	Motor de Animación entre Tween.js y Director.js	2	1		1
32	Animación	Crear animación para listas	2	2		1
7	UX	Optimizar Experiencia de Usuario.	2	2		2
24	Interactividad	Eventos para los botones de la interfaz interactiva	2		6	2
29	Interactividad	Vincular los 2 extremos de línea a 2 Objetos	2	3		2
33	Animación	crear animación para Pilas	2	2		2
39	Animación	Encadenar Animaciones una después de Otra	2		6	2
4	Compatibilidad	Implementar Responsive Design en panel 3D.	2		4	3
25	Interactividad	Touch del mouse para dispositivos móviles	2		6	3
31	Animación	eventos disparadores de animaciones	2		6	3
34	Animación	Crear animación para Arboles	2	3		3
40	Animación	Usar Recursividad y callbacks en JavaScript	2	2		3

Total 18,00 31,00
Total Días 21,88

Cuadro 10 Resumen de Tareas del 2do Sprint - Elaboración Propia

4.8.1. Burndown Chart - Sprint 2

Este es un gráfico que representa el progreso día a día del 2do Sprint.



Cuadro 11 Burndown Chart del Sprint 2 - Elaboración Propia

4.8.2. Implementar Responsive Design en panel 3D.

Los ejemplos de la aplicación se muestran en el navegador sobre un panel estático que adquiere el tamaño de la primera vez que se carga la página del navegador, pero existe un problema:

Cuando se redimensiona el tamaño del navegador el tamaño del espacio donde se muestra el contenido 3D, no se redimensiona y se queda

Esto causa el siguiente problema:

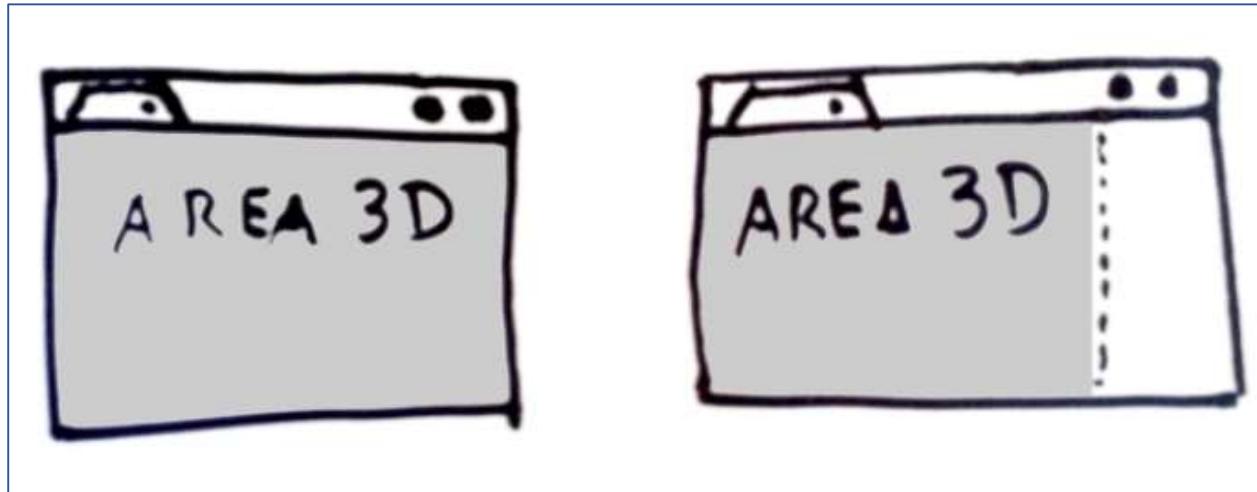


Ilustración 2 Problema de redimensión del Navegador - Elaboración Propia

En esta figura, A la izquierda se puede ver la página cargada al principio, (cubre todo el espacio del navegador). Pero cuando se redimensiona a la derecha se ve el problema mencionado.

Para solucionar esto se crea un evento que es capaz de realizar una actualización al tamaño de espacio donde se renderiza el contenido 3D cada vez que el Panel se redimensiona, ojo solo cuando se redimensiona. En otro caso está en espera hasta que esto suceda. Para no consumir recursos adicionales.

```
183
184
185
186     var onWindowResize = function() {
187         camera.aspect = window.innerWidth / window.innerHeight;
188         camera.updateProjectionMatrix();
189         renderer.setSize( window.innerWidth, window.innerHeight );
190     }
191
192
193
```

Figura 39 Funcion del Evento de Redimensión de la Ventana – Elaboración Propia.

Y se llama a esta función desde el conjunto de llamadas a eventos con la siguiente línea.

```
111
112
113     window.addEventListener( 'resize', onWindowResize, false );
114
115
```

Figura 40 llamada a Evento de Redimensión de la Ventana – Elaboración Propia.

Con esto se logra el objetivo de que el contenido 3D se acomode al tamaño del navegador cuando este se redimensione y además se centre al medio de la escena.

4.8.3. Optimizar Experiencia de Usuario.

Según la definición de experiencia de usuario (Wikipedia, 2013) “La experiencia de usuario es el conjunto de factores y elementos relativos a la interacción del usuario, con un entorno o dispositivo concretos, cuyo resultado es la generación de una percepción positiva o negativa de dicho servicio, producto o dispositivo.

La experiencia de usuario depende no sólo de los factores relativos al diseño (hardware, software, usabilidad, diseño de interacción, accesibilidad, diseño gráfico y visual, calidad de los contenidos, buscabilidad o encontrabilidad, utilidad, etc) sino además de aspectos relativos a las emociones, sentimientos, construcción y transmisión de la marca, confiabilidad del producto, etc.

Para que el usuario no tenga dificultad cuando inicie por primera vez a utilizar el programa, y lo maneje intuitivamente se aplican algunas reglas de mejora de experiencia de usuario que hacen de esta aplicación web, intuitiva y de fácil manejo.”

4.8.4. Desarrollar las vistas Estructuras de Datos.

En esta sección se desarrolla del lado del servidor siguiente en la estructura de Modelo Vista Controlador, utilizando las bondades de Codeigniter donde, en las vistas se ven en el siguiente condigo:

```
1
2
3  <?php
4  class Ed extends CI_Controller{
5      function index(){
6          $data = array( ... );
7          $this->load->view('ed/portada', $data);
8      }
9
10     public function listas($titulo){
11         $this->load->view('ed/listas/'.$titulo);
12     }
13
14     public function pilas($titulo){
15         $this->load->view('ed/pilas/'.$titulo);
16     }
17
18     public function colas(){
19         $this->load->view('ed/colas');
20     }
21
22     public function arboles($titulo){
23         $this->load->view('ed/arboles/'.$titulo);
24     }
25 }
26 ?>
27
28
```

Figura 41Llamadas a las distintas vistas desde Codeigniter - Elaboración Propia.

4.8.5. Desarrollar vista Plantillas: listas pilas árboles.

Una vez que uno ingresa a la sección de Estructuras de datos, uno puede encontrar en su interior, un conjunto de subsecciones

En estas sub secciones en las que se trabajó para cumplir esta tarea:

Con el uso de JQueryMobile. Y también organizando la plantilla HTML para que se muestre de manera correcta en distintos dispositivos respetando el tema de Responsive Design.



Figura 42 Captura de La sección de Estructuras de datos. - Elaboración Propia

4.8.6. Eventos para los botones de la interfaz interactiva.

Algo muy interesante es poder poner controles sobre el panel 3D, donde se muestra el contenido, esto tiene formas y formas de hacerse posible.

Primero experimenté con un Script Llamado **dat.gui.min.js** que es una herramienta para que provee una interfaz gráfica para modificar variables JavaScript en tiempo real, es de propósito general, se puede adaptar para el uso en el panel 3D.



Figura 43 Interfaz Gráfica dat.gui - fuente <https://code.google.com/p/dat-gui/>

Lo utilice, pero no es lo que estaba buscando para poner controles a la aplicación. Puesto que buscaba algo más simple y que se adapte al tema de la aplicación.

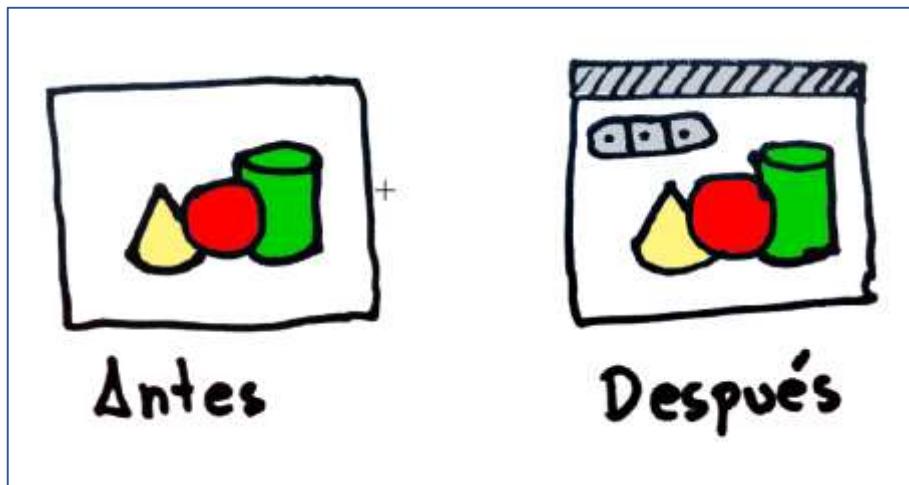


Figura 44 Esquema de los controles de la aplicación - Elaboración Propia.

La forma más conveniente que encontré es utilizar los mismos **componentes que provee JqueryMobile**, puesto que sus botones y barras de navegación permiten hacer sentir al usuario que está en un mismo ambiente de exploración.

JqueryMobile **no se integra por defecto con el panel 3D**, así que tuve que hallar la forma de integrarlos, básicamente crear la plantilla HTML, y en la sección donde aparece el Área 3D incluir los Tag¹⁴s necesarios para los diferentes botones y darles vida mediante JQuery, y usando los eventos para que realicen las diferentes acciones que les corresponde a cada uno de los controles.

El código para realizar todo esto implica CSS¹⁵, HTML, Y JavaScript, disperso en distintos lugares, así que simplemente muestro el resultado de integrar JqueryMobile con el Área 3D:

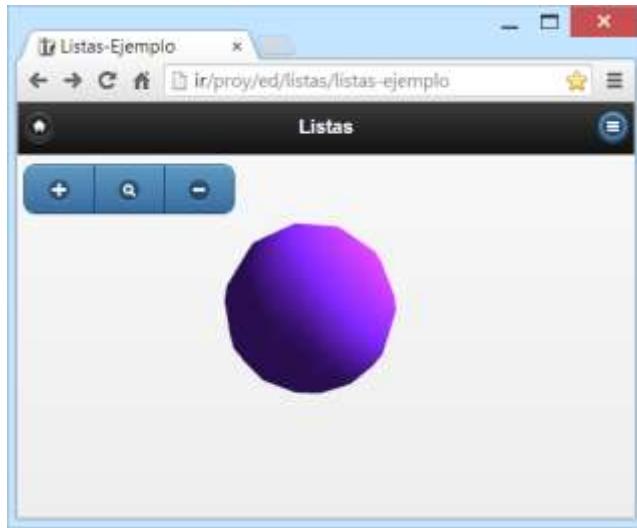


Figura 45 Captura que muestra los Controles de la Aplicación Web - Elaboración Propia.

¹⁴ Tag: a veces llamados "etiquetas" en español, son los "comandos" que los programas navegadores leen e interpretan para armar y dar forma a las páginas de Internet.

¹⁵ CSS: Las hojas de estilo en cascada (Cascading Style Sheets, o sus siglas CSS) Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML

Como se puede ver los controles (botones y barra de herramientas) concuerdan con el resto de la aplicación en cuanto a uniformidad de la interfaz de usuario. Y el resultado por lo menos personalmente es satisfactorio.

4.8.7. Touch del mouse para dispositivos móviles.

Para esta sección se trabajó en el sentido de que existen eventos especiales para los dispositivos móviles, como el tap Swipe, etc. pero afortunadamente resulta que los eventos necesarios para manipular los objetos en el Área 3D, trabajan de igual manera que los eventos para el mouse.

Por esta razón se hace uso directamente los eventos para el mouse, resultando el funcionamiento interactivo de la misma manera en el equipo de escritorio o laptop como en los dispositivos móviles con pantalla táctil.

4.8.8. Vincular los 2 extremos de línea a 2 Objetos.

La implementación de esta funcionalidad tiene el objetivo de uso en posteriores implementaciones donde se convino el uso de vinculaciones para crear cadenas de objetos vinculados entre sí.

Vincular 2 objetos a través de una línea es una tarea que involucra varios aspectos, primero veamos el resultado al que se quiere llegar.

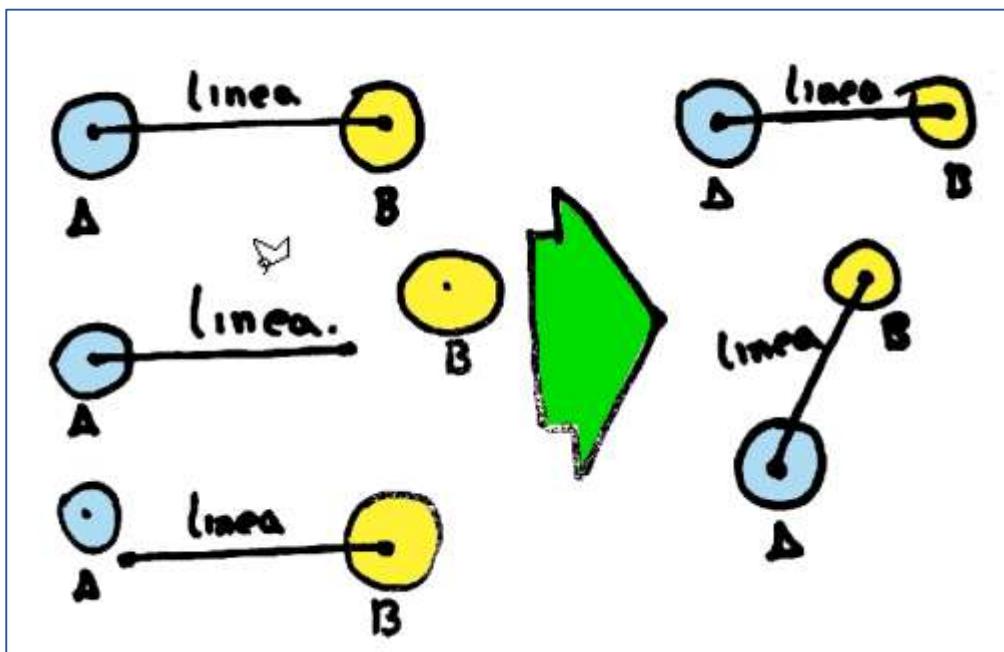


Figura 46 Problemas de vincular 2 objetos a través de una línea - Elaboración Propia.

Al crear una línea y los objetos en cuestión, estas son independientes en el espacio, cuando uno intenta mover uno de los objetos, la línea no se mueve a la par con el objeto, como hacer esto posible?.

Primero, es necesario entender que cuando uno mueve un objeto, este constantemente **actualiza su posición**, a lo largo del recorrido del mouse o el dedo en el caso de pantalla táctil.

También es importante comprender que una línea tiene 2 extremos, y cada extremo es independiente para asociarse a un objeto por extremo, en este caso 2 objetos 2 extremos una línea.

La idea consiste que en cada vez que el mouse se mueva mientras se mueve un objeto, la línea se actualice la posición del Extremo Asociado al Objeto.

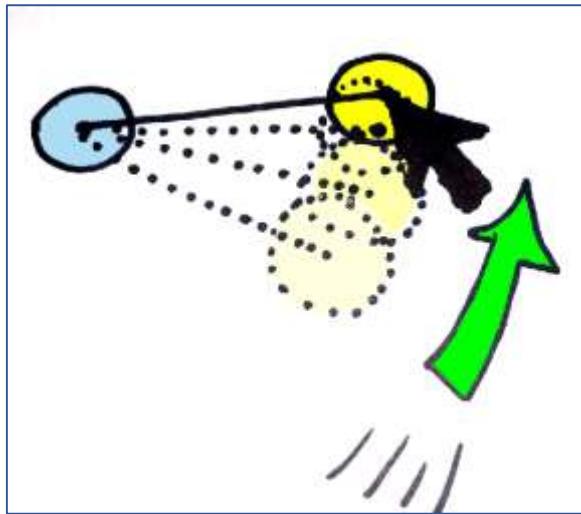


Figura 47 Esquema de vinculación entre línea y objetos - Elaboración Propia.

El Código necesario para crear la línea y 2 Objetos se vinculen a él es el siguiente:

```

368
369
370     if (anterior) {
371         linea = create_line(
372             new THREE.Vector3(
373                 object.position.x,
374                 object.position.y,
375                 object.position.z
376             ),
377             new THREE.Vector3(
378                 object.position.x,
379                 object.position.y,
380                 object.position.z
381             ),
382             0x777777)
383
384     lineas.push(linea)
385     group.add(linea);
386     anterior.lineas[0]={"linea":linea,"extremo":0}
387     sub_i= anterior.lineas.length-1; /*posicion del ultimo*/
388     ext = anterior.lineas[sub_i].extremo;
389     linea.geometry.vertices[0].copy(anterior.position)
390     linea.geometry.verticesNeedUpdate=true;
391
392     object.lineas = [ null, {"linea":linea,"extremo":1} ]
393     linea.geometry.verticesNeedUpdate=true;
394
395     linea.nombre = "linea:"+ (vertexObjects.length-1)
396     object.dato = obj_text
397     console.log("->:"+object.dato)
398 }
399 else {
400     object.lineas = []
401     object.dato = obj_text
402     console.log("FA"+object.dato)
403 }

```

Figura 48 Código para Vincular 2 objetos a una línea - Elaboración Propia.

Y el evento que actualiza la posición de la línea mientras el objeto vinculado a esta se implementa dentro de la función de evento **onDocumentMouseMove**:

```
222
223     if (SELECTED.lineas) {
224         for (var i = 0; i < SELECTED.lineas.length; i++) {
225             if (SELECTED.lineas[i]) {
226                 ext = SELECTED.lineas[i].extremo;
227                 SELECTED.lineas[i].linea.geometry.vertices[ext].copy(SELECTED.position)
228                 SELECTED.lineas[i].linea.geometry.verticesNeedUpdate=true;
229             }
230         }
231     }
232 
```

Figura 49 Código para Actualizar la posición de la línea mientras se mueve un objeto - Elaboración Propia.

El resultado de esta implementación permite reutilizar para crear lo siguiente, que posteriormente explico a detalle:

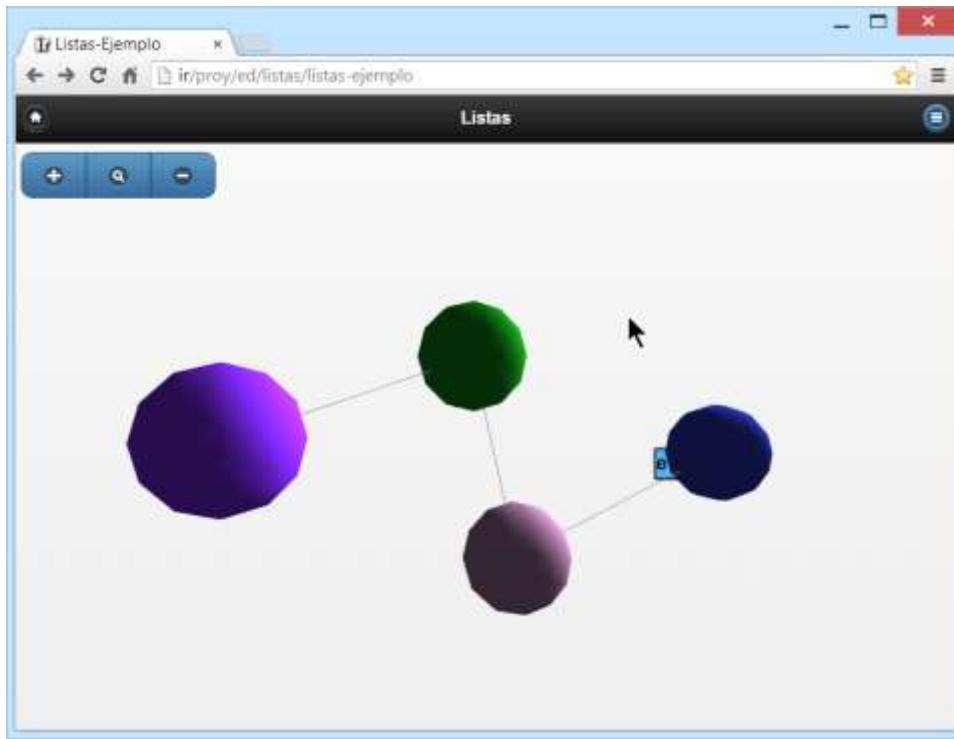


Figura 50 Captura del resultado de vincular varios objetos y varias líneas - Elaboración Propia.

4.8.9. Motor de Animación entre Tween.js y Director.js

Para poder **animar** que es uno de los **objetivos principales de este proyecto**, se tiene que es necesario recurrir a herramientas externas al API de Three.js, puesto que no tiene un animador nativo, intuitivo de implementar.

Por esta razón investigando encontré 2 Librerías que se encargan de este trabajo: Tween.js y Director.js.

4.8.9.1. Director.js

Al experimentar con Director.js, que es una librería implementada por la misma comunidad de los desarrolladores de Three.js, me di cuenta que no contaba con mucha documentación y se dejó de actualizar desde hace ya más de 2 años, lo cual me indica que no tiene muchos usuarios trabajando con esta librería.

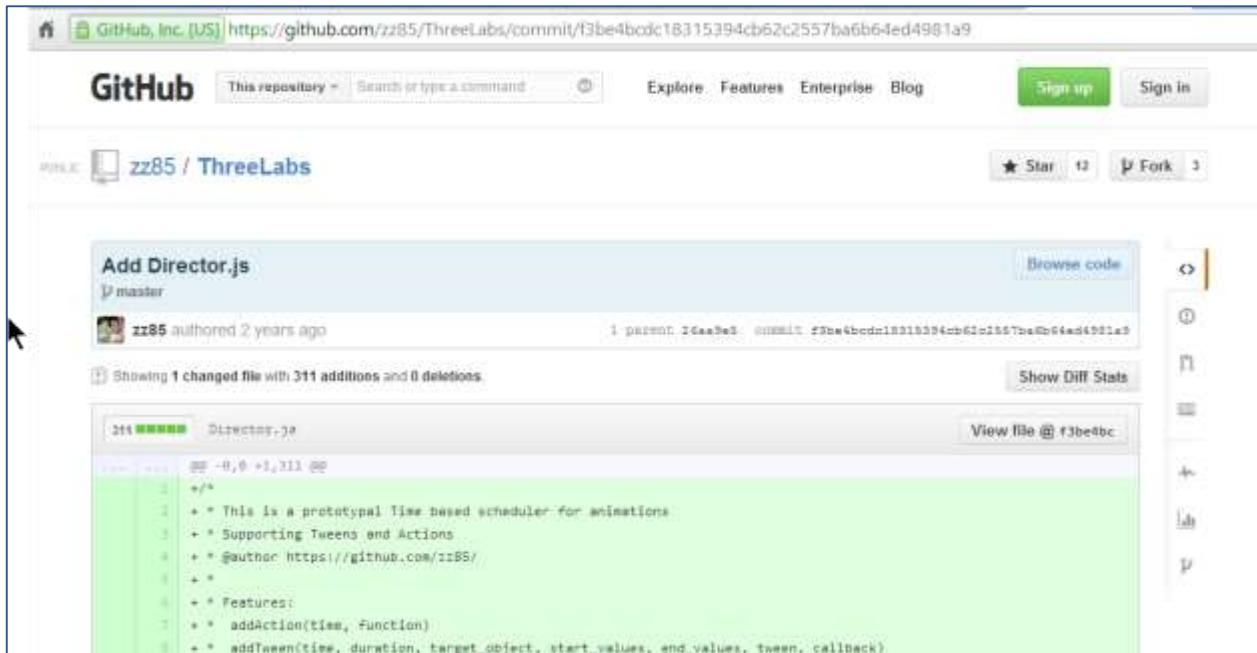


Figura 51 Página en GitHub del Director.js - Fuente <https://github.com>

Haciendo un esfuerzo por animar utilizando esta librería me di cuenta que no cuenta con documentación necesaria. Y los resultados de la animación no son precisos, ya que varían en los tiempos en diferentes corridas. A demás que no encontré la forma de tener múltiples animaciones simultaneas.

Por estas razones Descartado! Para utilizar en este proyecto.

4.8.9.1. Tween.js

Una alternativa para animar, es esta, Tween.js, es una librería en JavaScript desarrollada con el objetivo de crear Interpolaciones dados 2 Estados: Inicial, Final, asociados a un tiempo determinado, es de propósito general, que es que se puede usar en diferentes proyectos no solo para 3D.



Figura 52 Página de Tween.js – Fuente: <http://www.createjs.com/#!/TweenJS>

Estos estados pueden ser cualquier cosa, color, tamaño, posición, rotación, etc., lo que lo hace muy flexible.

Tween.js provee a demás diferentes tipos de interpolación, a parte de la lineal, existen la elástica, y otros que permiten hacer las interpolaciones más interesantes.

El uso además de encadenado de eventos permite realizar animaciones tanto secuenciales como paralelas.

Este es el código de una pequeña animación para cambiar de posición un determinando objeto a través del área del espacio 3D.

```
469
470
471
472
473     var pos_ini = { x: find_anterior.position.x, y: find_anterior.position.y, z: find_anterior.position.z };
474
475     new TWEEN.Tween( pos_ini )
476         .to( { x: find_actual.position.x, y: find_actual.position.y, z: find_actual.position.z }, tiempo )
477         .easing( TWEEN.Easing.Elastic.InOut )
478         .onUpdate(function (){
479             aural.position.set(pos_ini.x, pos_ini.y, pos_ini.z)
480             aural.lookAt(npoint);
481         })
482         .start();
483
484
485
```

Ilustración 3 Código de la animación para mover un objeto - - Elaboración Propia.

4.8.10. Eventos disparadores de animaciones.

Las animaciones no empiezan cuando se carga la página, esto quiere decir que existe un determinado momento definido por el usuario que esta interactuando con la aplicación web, cuando estas animaciones deben darse de inicio.

Una animación puede ser una, o una secuencia desencadenada por accionar la primera de ellas, bajo este concepto, es necesario definir cuáles son las acciones para dar inicio a las animaciones.

Para el caso de inicio de una animación, en esta aplicación, se establece que la presión de ciertos botones en la interfaz gráfica, accionan la secuencia de animaciones.

Por ejemplo:

```
116
117
118
119
120     $(document).ready(function(){
121         $("#nuevo-elemento").click(function(){
122             addElement();
123             $("#onamea").val("");
124         })
125
126         $("#buscar-elemento").click(function(){
127             findElement( $("#onameb").val() );
128             $("#onameb").val("");
129         })
130
131
132         $("#eliminar-elemento").click(function(){
133             deleteElement( $("#onamee").val() );
134             $("#onamee").val("");
135         })
136     });
137
138
139
140
```

Figura 53 Lugares donde al presionar con el cursor se Inician las Animaciones – Elaboración Propia

4.8.11. Encadenar Animaciones una después de Otra.

Una vez sabiendo cómo crear una animación, como llamarla, es necesario entender cómo hacer que se pueda secuenciar una animación después de la otra.

Esto se logra gracias al uso de callbacks y recursividad explicados en el siguiente subtítulo, el encadenado se refiere a que cuando se tiene diferentes animaciones, cada una de estas creadas e iniciadas, por defecto inician el instante que se las crea.

Esto causa que todas las animaciones se den al mismo tiempo, lo cual no siempre es lo que se desea.

Por esta razón, es necesario entender como secuenciar las animaciones, el siguiente gráfico explica que es lo que se necesita.

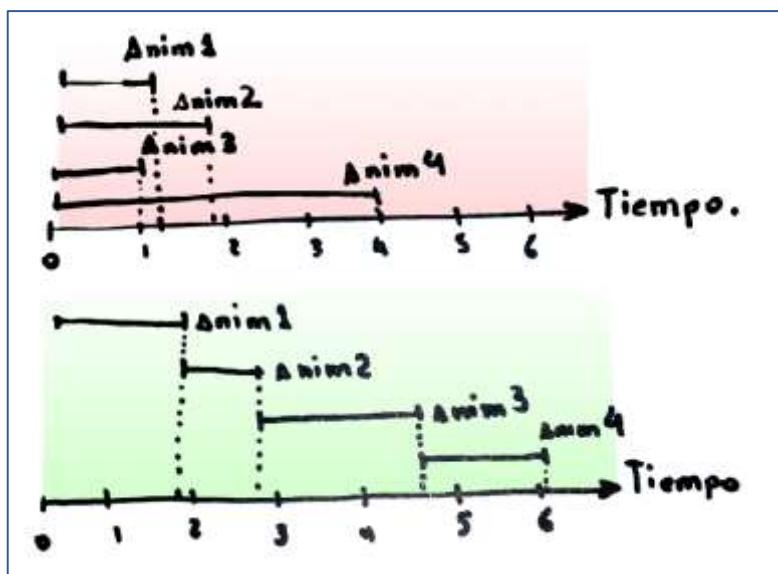


Figura 54 Secuencia de Animación - Elaboración propia

4.8.12. Usar Recursividad y callbacks en JavaScript.

Para hacer posible que las secuencias del anterior subtítulo funcionen correctamente, JavaScript afortunadamente cuenta con Callbacks, que esperan a que una animación inicie justo cuando otra haya terminado.

La segunda funcionalidad necesaria es la recursividad, ya que con un iterador como For o while, no se puede hacer posible la secuencia, puesto que el iterador inicia las animaciones al tiempo que recorre el arreglo donde se almacenan.

Por esta razón la recursividad, hace que de almacene en una Pila de memoria, la animación actual, y hasta que termine recién llama a la siguiente llamada de animación.

```

480
481
482     new TWEEN.Tween( pos_ini )
483         .to( { x: find_actual.position.x,y: find_actual.position.y,z: find_actual.position.z }, tiempo )
484         .easing( TWEEN.Easing.Elastic.InOut )
485         .onUpdate(function (){ })
486     })
487     .start()
488     .onComplete(function(){
489         if( find_actual.data == elm && lst.buscar(elm) ){
490             var pos_ini3 = { x: 0 };
491             new TWEEN.Tween( pos_ini3 )
492                 .to( { x: 255 },5000 )
493                 .easing( TWEEN.Easing.Linear.None )
494                 .onUpdate(function(){
495                     $("titulo-main").css("color", "rgb(0,228,0)");
496                 })
497                 .start()
498                 .onComplete(function(){ })
499             })
500         }
501     })
502     .else
503         fbucle(arr,i+1,find_actual);
504     })
505
506
507
508
509
510
511
512
513
514
515
516
517

```

Figura 55 Callbacks para secuenciar 2 animaciones - Elaboración Propia

4.8.13. Crear animación para listas.

Las listas de simple como se vio teóricamente en el marco referencial tienen funcionalidades básicas, estas son:

- Inserción
- Eliminación
- Búsqueda.

Existen otras, pero son esas funcionalidades que se incluyen en el presente la aplicación web, por ser las más utilizadas.

Para poder desarrollar estas funcionalidades gráficamente es necesario utilizar básicamente animaciones, eventos, y las reglas para que se vayan ejecutando secuencialmente. Como se vio anteriormente se utiliza la librería externa, Tween.js, para este objetivo.

4.8.13.1. Inserción

La animación de inserción de un elemento en la lista es sencilla, simplemente se van añadiendo elementos al final de la lista, es necesario brindar un procedimiento que introduzca nuevos elementos, cada elemento nuevo tiene que contener un nombre.

la secuencia de acciones para insertar un Objeto es como sigue...

- Presionar el botón de inserción
- Solicitar un nombre para el nuevo elemento que se insertará
- Terminar la inserción presionando un botón.



Figura 56 Secuencia de eventos para crear un objeto - Elaboración propia

Este conjunto de acciones, crea un objeto en la escena con el nombre que acaba de escribir:



Figura 57 Ejemplo de añadir un elemento en Listas - Elaboración Propia

4.8.13.1. Buscar

Encontrar un elemento en la lista es un proceso de recorrido que es muy interesante mostrarlo gráficamente, el procedimiento de secuencia de acciones para buscar un elemento es el mismo que el de la inserción, pero el resultado es lo que varía.

Para buscar un elemento en la lista se podría mostrar el proceso recursivo de búsqueda mientras va comparando el elemento buscado con el elemento actual en el recorrido recursivo de la lista... esto se ve plasmado en la sección de ejemplo de listas en la aplicación web.

En la figura siguiente se tiene una lista donde se fueron añadiendo los elementos que se pueden apreciar:

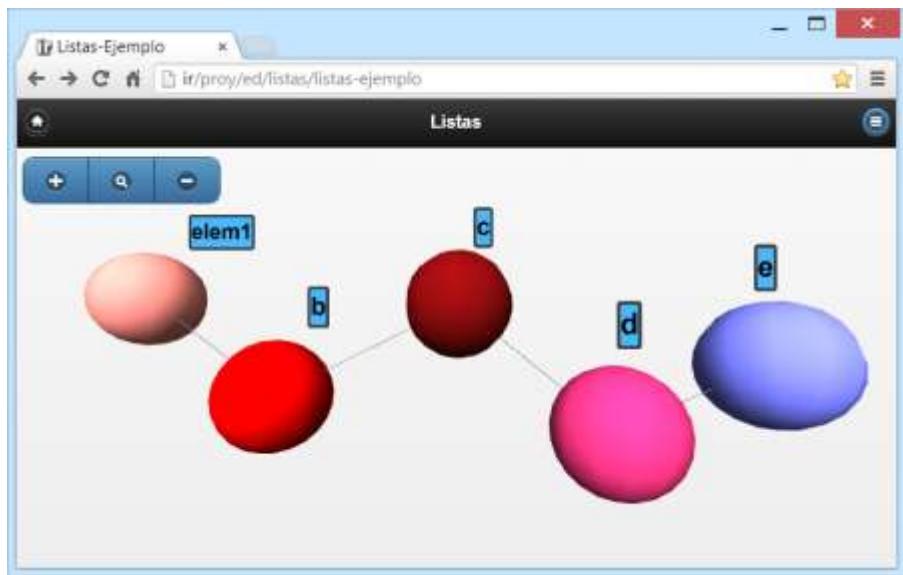
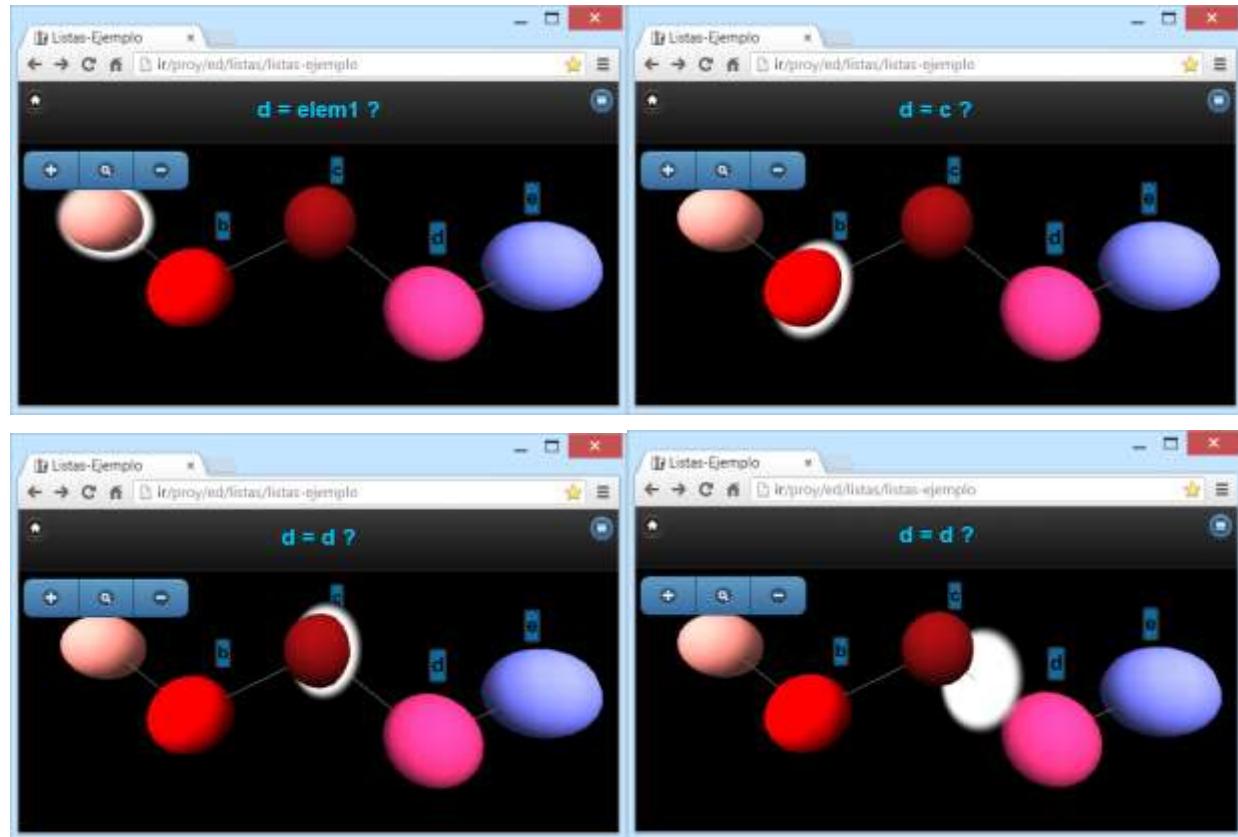


Figura 58 Lista de elementos - Elaboración Propia

En donde si buscamos por ejemplo el elemento “d” el recorrido se muestra al usuario de la siguiente manera: primero se oscurece la pantalla, seguidamente, se va moviendo un Aura de color blanco a través de los nodos o elementos, comparando lo buscado con la situación actual del Aura.

Se termina de iterar cuando se encuentra el elemento o cuando se llega al final de la lista sin haberlo encontrado.

Esta es una secuencia de de encontrar el elemento “d”



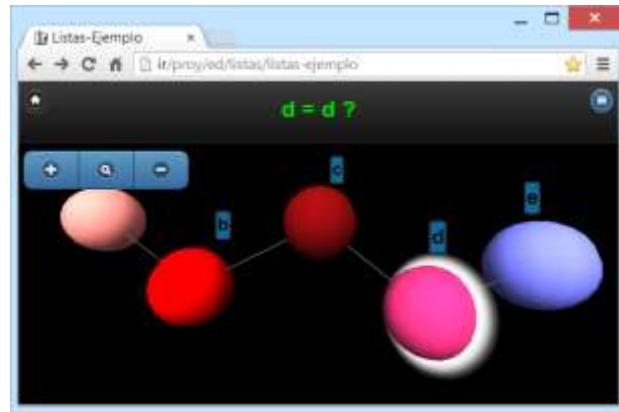


Figura 59 Secuencia de Búsqueda en una lista - Elaboración Propia

Cuando se encuentra el elemento las letras donde se muestra la comparación en la barra superior cambian de color a Verde lo cuan indica que hubo éxito al encontrar el elemento.

4.8.13.2. Eliminar

Eliminar un elemento en una lista implica 3 casos:

- el elemento esta entre varios nodos.
- El Elemento es el penúltimo Elemento
- El Elemento es el último Elemento.

4.8.14. Crear animación para Pilas.

Esta estructura sigue la filosofía de que se puede insertar elementos solo al tope del conjunto de elementos el último en ser ingresado, gráficamente se puede demostrar sus funcionalidades de manera muy llamativa. Como se muestra a continuación:

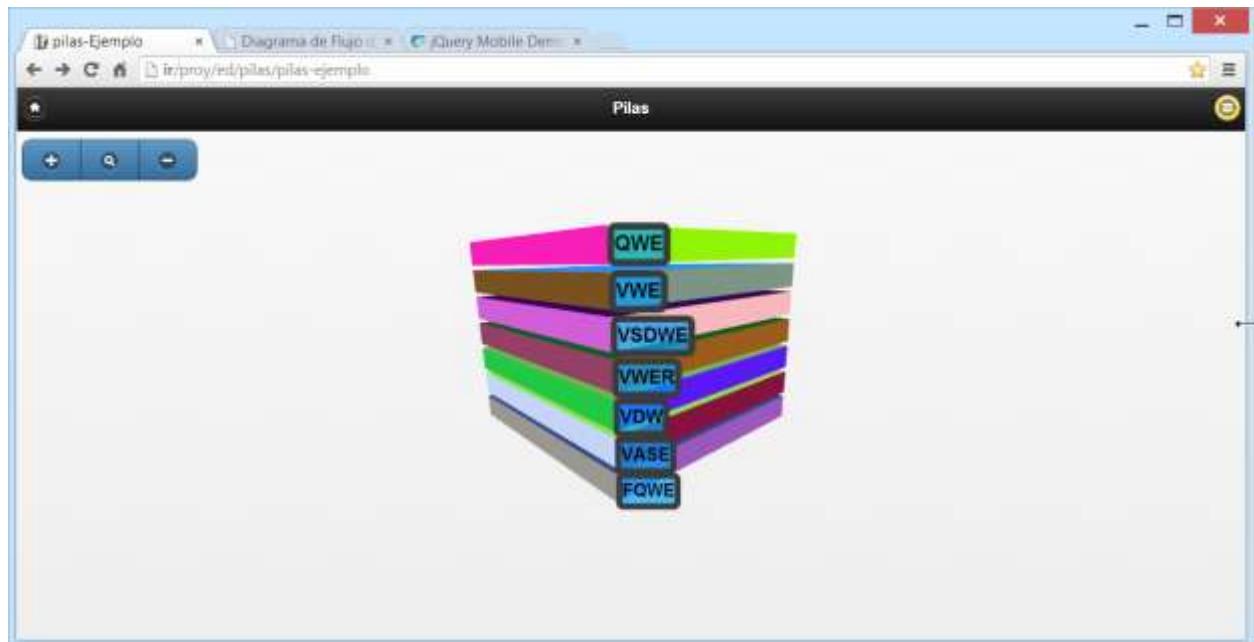


Figura 60 Ejemplo de Pilas en la Aplicación -Elaboración Propia

Donde uno puede añadir elementos (Push) buscar el elemnto tope de la Pila (Top) y eliminar el elemento Tope de la Pila(pop).

4.9. Sprint 3.

Para este sprint se seleccionaron las siguientes tareas

N	Grupo	Tarea	Sprint	D	H	Prior
12	UX	Desarrollar las vistas Redes	3		5	1
15	UX	Desarrollar vista Plantillas: sección de Redes	3	2		1
37	Animación	Desarrollar animaciones para Establecimiento de C.	3	3		1
20	Objetos 3D	Importar Objetos de 3Ds Max mediante la API Three.js	3	1		2
35	Animación	Desarrollar animaciones para Transporte de P.	3	3		2
17	Objetos 3D	Crear un modelo 3D de una PC	3		3	3
18	Objetos 3D	Crear un modelo 3D de una Laptop	3		3	3
19	Objetos 3D	Crear un modelo 3D Router	3		2	3
36	Animación	Desarrollar animaciones para Redirección de P.	3	3		3

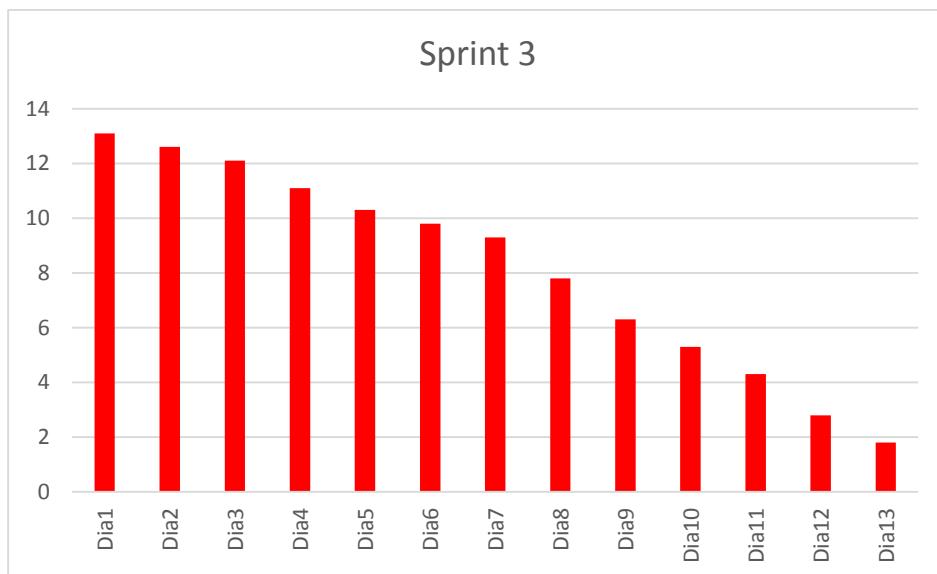
Total 12,00 13,00

Total Días 13,63

Cuadro 12 Resumen de Tareas del 3er Sprint - Elaboración Propia

4.9.1. Burndown Chart - Sprint 3

Este es un gráfico que representa el progreso día a día del 3er Sprint.



Cuadro 13 Burndown Chart del Sprint 3 - Elaboración Propia

4.9.2. Desarrollar las vistas Redes.

La sección de redes contempla esta subdivida en 3 ejemplos para mostrar esto al usuario se hace uso de las reglas establecidas para mejorar la experiencia de usuario.

El resultado de implementar esta sección es gracias a utilizar las imágenes creadas anteriormente, organizadas para cada ejemplo.

Se llama utilizar el siguiente código en la plantilla de la portada de la vista de redes, esto sobre el entorno de trabajo de Codeigniter.

La clase que crea las subsecciones del contenido es:

```
2  <?php
3  class Redes extends CI_Controller{
4      function index(){
5          $data = array(
6              'contenido' => array(
7                  'transporte' => array(
8                      'urllink' => 'redes/transporte',
9                      'urlimg' => 'img/redes/transporte.svg',
10                     'name' => 'Transporte de Paquetes.',
11                     'comment' => 'Transpote'
12                 ),
13                 'conexion' => array(
14                     'urllink' => 'redes/conexion',
15                     'urlimg' => 'img/redes/conexion.svg',
16                     'name' => 'Establecimiento de Conexion TCP',
17                     'comment' => 'Conexiones TCP'
18                 ),
19                 'redireccion' => array(
20                     'urllink' => 'redes/redireccion',
21                     'urlimg' => 'img/redes/redireccion.svg',
22                     'name' => 'Redireccion de paquetes ',
23                     'comment' => 'Redireccion a travez de Proxis y Fiewalls'
24                 )
25             )
26         );
27     }
28
29
30     $this->load->view('redes/portada',$data);
31 }
32 public function transporte(){
33     $this->load->view('redes/transporte/transporte-ejemplo');
34 }
35 public function conexion(){
36     $this->load->view('redes/conexion/conexion-ejemplo');
37 }
38 public function redireccion(){
39     $this->load->view('redes/redireccion/redireccion-ejemplo');
40 }
41 }
42 ?>
43 }
```

Figura 61 Clase para llamar las Vistas de la Sección de redes - Elaboración Propia

4.9.3. Desarrollar vista Plantillas: sección de Redes

En la parte del cliente es necesario crear una plantilla para vincular cada una de las secciones con la página que corresponde esto se hace usando un ciclo que itera un arreglo que es enviado desde el servidor, que contiene la información necesaria para crear estos links.

```

2 <script type="text/javascript">
3   window.location=<?php echo $this->config->base_url(''); ?>;
4 </script>
5 <div data-role="page" id="redes" data-title="redes" data-theme="c" >
6   <div data-role="header" data-position="fixed">
7     <a href=<?php echo $this->config->base_url(''); ?>" data-icon="home" data-iconpos="notext"> </a>
8     <h1>Redes</h1>
9   </div> I
10  <div data-role="content">
11    <ul data-role="listview" data-inset="true">
12      <?php
13      foreach ($contenido as $key => $value) {
14        echo '<li>';
15        echo '<a href="'.$this->config->base_url($value['urllink']).'" data-inline="true" data-transition="slide">';
16        echo '';
17        echo '<h2>'.$value['name'].'</h2>';
18        echo '<p>'.$value['comment'].'</p>';
19        echo '</a></li>';
20      }
21    </ul>
22  </div>
23 </div>
24
25
26 </div>
27 </div>
28
29

```

Figura 62 Plantilla para la sección de Redes - Elaboración propia

4.9.4. Crear un modelo 3D de una PC

Par crear estos modelos se utiliza un programa muy completo llamado 3DSMax. Una PC, representa un nodo en una infraestructura de red, lo que nos ayuda a mostrarlo en Área 3D del navegador, para las secciones del contenido educativo de Redes.

El Modelado de la PC en base a objeto primitivos, modificados en cuanto a tamaño y forma.

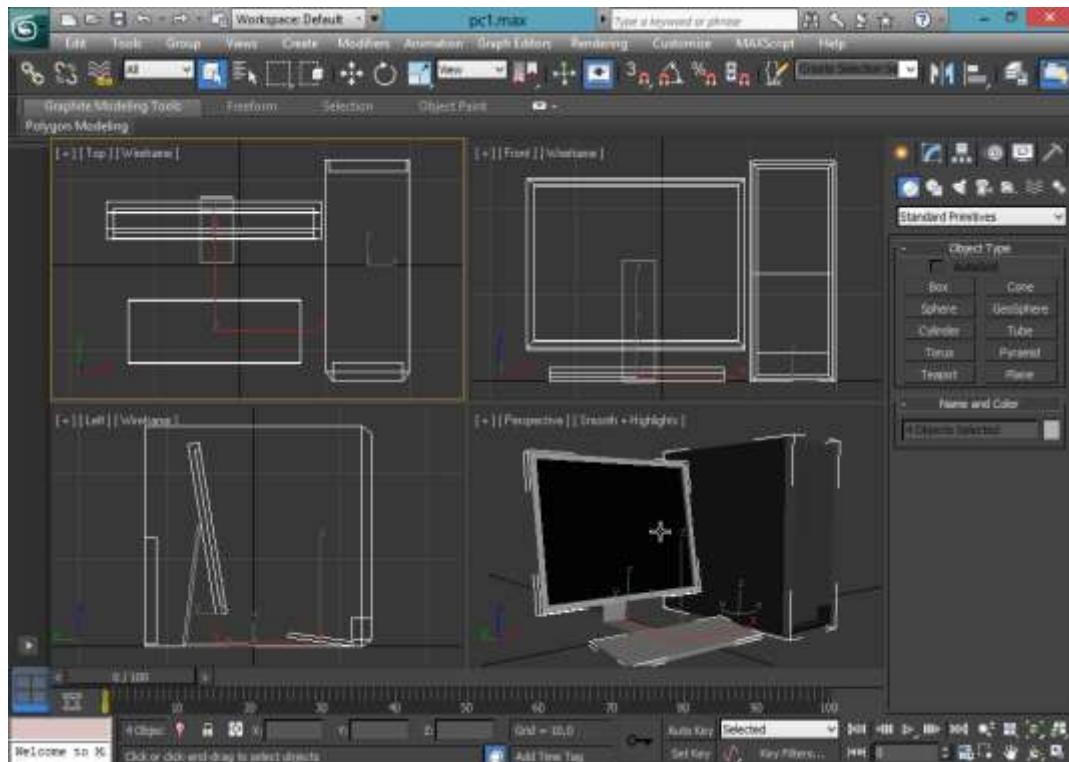


Figura 63Modelo 3D: PC - Elaboración Propia.

4.9.5. Crear un modelo 3D de una Laptop

Este es el diseño de una laptop en base a figuras geométricas básicas:

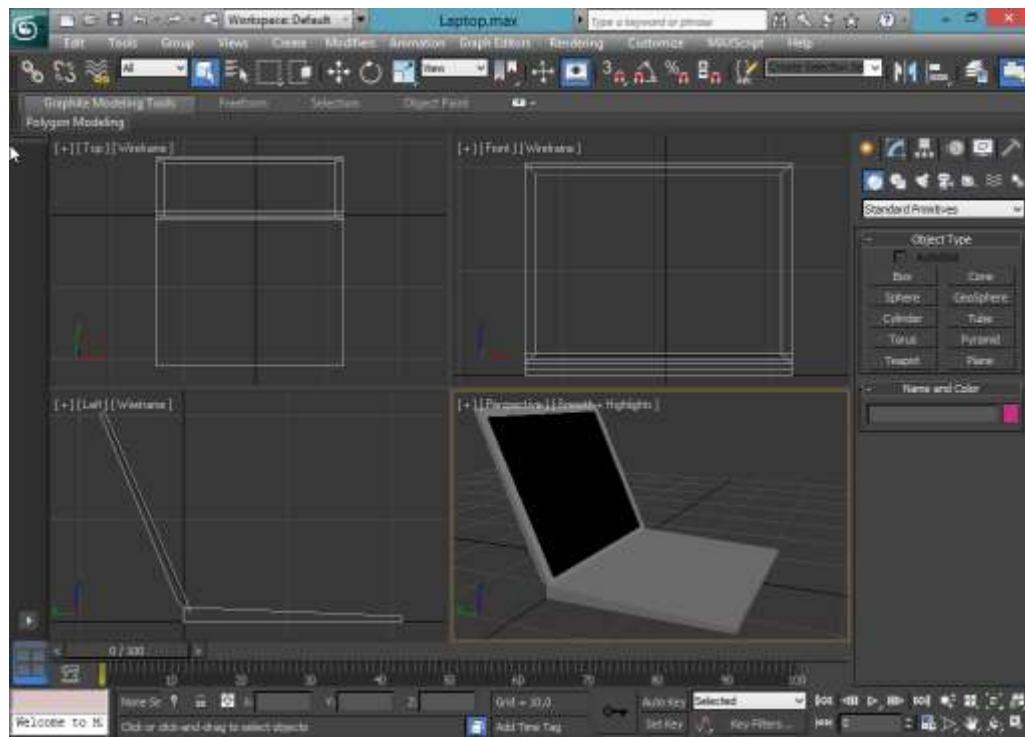


Figura 64 Modelo 3D: Laptop - Elaboración Propia.

4.9.6. Crear un modelo 3D Router

Modelo 3D para un representar un Router de redes de computadoras.

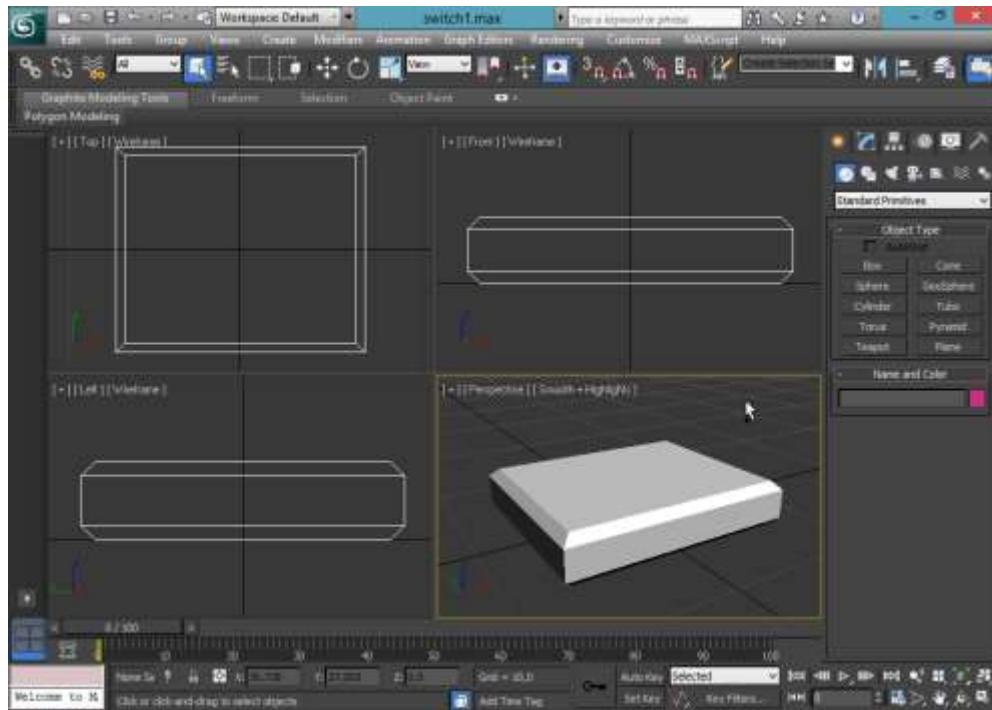


Figura 65 Modelo 3D: Router - Elaboración Propia.

4.9.7. Importar Objetos de 3DSMax mediante la API Three.js

El objetivo de importar objetos en este caso de 3DSMax, es para poder utilizar cualquier modelo 3D que en este programa de graficación 3D se pueda crear, para luego añadirlo en el Área 3D del navegador creado en esta aplicación Web.

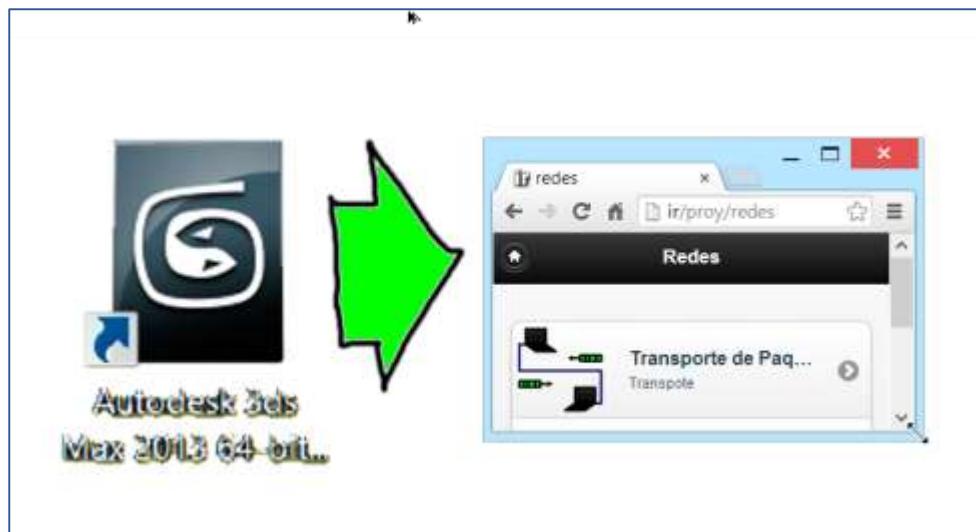


Figura 66 Exportar 3DsMax a Objeto compatible Web – Elaboración Propia.

El Procedimiento consiste en:

- Tener centrado el Modelo 3D en la escena del programa 3DSMax.
- Exportar como archivo de Extensión .OBJ
- Tener instalado Python y usar este Script **convert_obj_three.py** que está incluido en las utilidades de la API de Three.js, ejecutar el siguiente comando:

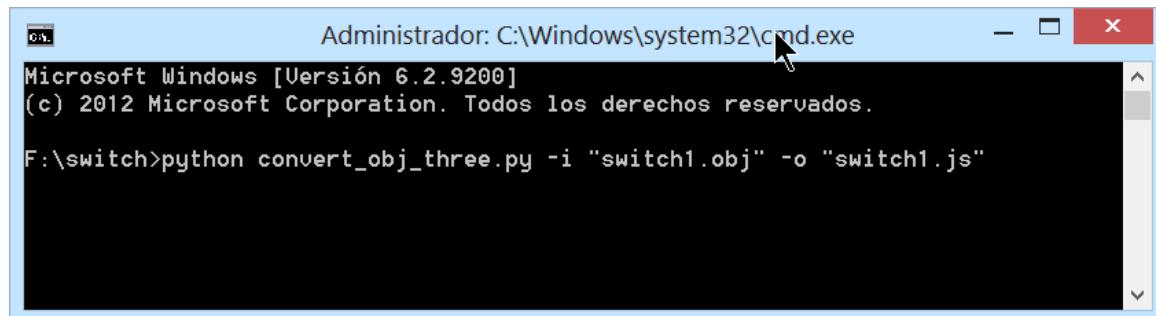


Figura 67 Comando Python para convertir un Archivo OBJ a JS – Elaboración Propia

- Esto genera un archivo en formato JSON que representa el Modelo 3D.
- Y luego se carga de la siguiente manera el objeto mediante la API de Three.js.

```

53
54     //cargar OBjecto externo (Toma un tiempo)
55     models3d = [{url:base_url_js+"img/other/laptop/laptop1.js",nam:"mono",scale:10},
56                 {url:base_url_js+"img/other/pc/pc.js",nam:"cilindro",scale:10},
57                 {url:base_url_js+"img/other/switch/switch1.js",nam:"curva",scale:10}]
58     loader = new THREE.JSONLoader();
59     funload(models3d,0)
60     function funload(arr,i){
61         if(arr[i]){
62             loader.load( models3d[i].url, function( geometry, materials ) {
63                 materials[0].shading = THREE.SmoothShading;
64                 var material = new THREE.MeshFaceMaterial( materials );
65                 mesh = new THREE.Mesh( geometry, material );
66                 mesh.scale.set( models3d[i].scale, models3d[i].scale, models3d[i].scale );
67                 mesh.position.y = Math.random() * 900-900;
68                 mesh.position.x = Math.random() * 900-900;
69                 mesh.position.z = Math.random() * 900-900;
70
71                 mesh.nam = models3d[i].nam
72                 obj.push(mesh);
73                 scene.add( mesh );
74                 funload(arr,i+1);
75             } );
76         }
77     }
78 }
```

Ilustración 4 Importación de Archivos Exportado de OBJ a JS - Elaboración Propia

4.9.8. Desarrollar animaciones para Establecimiento de Conexión.

En esta sub-sección de Redes es suficiente con demostrar cual es el proceso de conexión que siguen 2 nodos para poder establecer una conexión.

Este proceso es explicado teóricamente en varias biografías, es conocido como Three-Way Handshake.

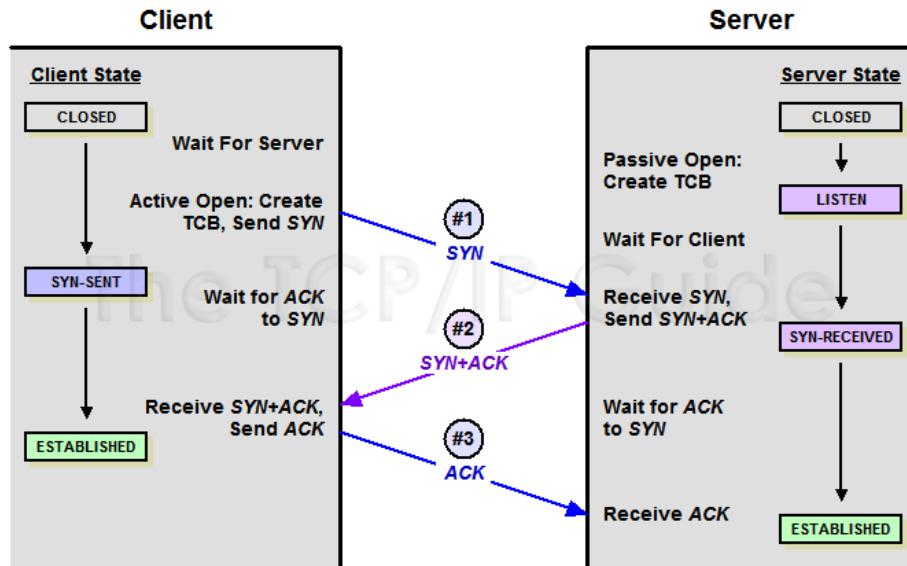


Figura 68 Three-Way Handshake Fuente:
http://www.tcpipguide.com/free/t_TCPConnectionEstablishmentProcessTheThreeWayHandsh-3.htm

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES.

5. Conclusiones y recomendaciones.

5.1. Conclusiones.

Es un logro muy importante haber experimentado con las tecnologías plasmadas en este proyecto de grado,

Entre estas tecnologías tenemos WebGL, que gracias a la Librería Three.js, uno puede acceder a funcionalidades del acelerador Grafico de los equipos o dispositivos, desde el Navegador Web. Permitiendo aprovechar de esta manera las capacidades graficas de los distintos dispositivos,

Three.js es una librería en JavaScript que permite crear un escenario, incluir objetos, luces, cámaras, e interactuar con ellos en diversas maneras, en un entorno 3D, **facilita de sobremanera el desarrollo de este tipo de aplicaciones**, ya que abstrae bastante complejidad al momento de desarrollar, brindando funcionalidades muy intuitivas.

Para realizar animaciones, **TWEEN.js es la mejor alternativa**, es otra librería desarrollada en JavaScript muy bien sincronizada, que permite encadenar una serie de animaciones secuencial y paralelamente. Brindando muy buenos resultados. En la aplicación de este proyecto contribuyó bastante en cuanto a las animaciones.

La aplicación web creada en este proyecto, contempla la facilidad de acceso del contenido del mismo, para que se pueda visualizar desde cualquier dispositivo y plataforma, esto es una forma de tener disponible esta herramienta en cualquier lugar y momento (con conexión a internet).

Todo este esfuerzo es para adaptarse principalmente a las nuevas tecnologías que están emergiendo, para brindar una mejor forma de hacer entender los conceptos, a los estudiantes en general a las personas.

5.2. Recomendaciones.

Simplemente motivar la creación de contenido interactivo y en 3 Dimensiones sobre la WEB, que aumente el deseo de adquirir conocimientos.

El uso de la librería Three.js, debería ser la primera opción, cuando uno quiera desarrollar para la web. Por su facilidad de aprendizaje, y resultados que llaman mucho la atención.

Una recomendación técnica es que las secciones con contenido de la aplicación web, se muestran en diferentes navegadores, Firefox, Chrome, Internet Explorer, Safari, pero para tener mayor fluidez al visualizar el contenido 3D, de cada uno de los ejemplos, se recomienda tener activado **WebGL**, lo cual dará una experiencia mucho más rápida que usando **Canvas**.

BIBLIOGRAFÍA

- Aula Virtual UCV.* (Oct de 2011). Obtenido de <http://aula.virtual.ucv.cl/wordpress/ventajas-y-desventajas-del-uso-de-las-tic-en-la-educacion/>
- Codeigniter.* (nov de 2013). Obtenido de <http://ellislab.com/codeigniter/user-guide/>
- Flanagan, D. (2002). *JavaScript: The Definitive Guide* (4 ª Edición edición ed.).
- Tanenbaum. (2003). *Wikipedia*. Obtenido de http://es.wikipedia.org/wiki/Red_de_computadoras
- Udacity.* (Junio de 2013). Obtenido de <https://www.udacity.com/course/cs291>
- Wikipedia.* (jun de 2013). Obtenido de http://es.wikipedia.org/wiki/Estructura_de_datos
- Wikipedia.* (24 de jun de 2013). Obtenido de [http://es.wikipedia.org/wiki/Pila_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica))
- Wikipedia.* (10 de nov de 2013). Obtenido de <http://es.wikipedia.org/wiki/WebGL>
- Wikipedia.* (nov de 2013). Obtenido de http://es.wikipedia.org/wiki/Estructura_de_datos
- Wikipedia.* (Nov de 2013). Obtenido de [http://es.wikipedia.org/wiki/Lista_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Lista_(inform%C3%A1tica))
- Wikipedia.* (nov de 2013). Obtenido de [http://es.wikipedia.org/wiki/Pila_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Pila_(inform%C3%A1tica))
- Wikipedia.* (nov de 2013). Obtenido de [http://es.wikipedia.org/wiki/%C3%81rbol_\(estructura_de_datos\)](http://es.wikipedia.org/wiki/%C3%81rbol_(estructura_de_datos))
- Wikipedia.* (nov de 2013). Obtenido de [http://es.wikipedia.org/wiki/Cortafuegos_\(inform%C3%A1tica\)](http://es.wikipedia.org/wiki/Cortafuegos_(inform%C3%A1tica))
- Wikipedia.* (nov de 2013). Obtenido de http://es.wikipedia.org/wiki/Experiencia_de_usuario
- Wikipedia.* (jun de 2013). Obtenido de [http://es.wikipedia.org/wiki/Lista_\(estructura_de_datos\)](http://es.wikipedia.org/wiki/Lista_(estructura_de_datos))