

UNIVERSIDAD TECNOLÓGICA ISRAEL



CARRERA DE SISTEMAS INFORMÁTICOS

**“ANÁLISIS DE SOFTWARE DE DESARROLLO 2D - 3D VS HTML5 – APIS Y SU
POSICIONAMIENTO TECNOLÓGICO”**

AUTOR:

Edison Fernando Guamantari Criollo

TUTOR:

Ing. Mario Mejía

Quito - Ecuador

2013

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE TESIS

La abajo firmante, en calidad de estudiante de la Carrera de Sistemas Informáticos declaro que los contenidos de este Trabajo de Graduación, requisito previo a la obtención del Grado de Ingeniero en Sistemas Informáticos, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito, junio del 2013

Edison Fernando Guamantari Criollo

CC: 010459141-7

DEDICATORIA

Este trabajo de graduación va dedicado a mis padres, por darme la vida y por confiar en mí y en lo que hago, por darme todo el apoyo moral, económico, por ellos es por lo que este proyecto tiene su éxito.

AGRADECIMIENTO

Primero quiero agradecer a Dios por darme la sabiduría y paciencia para sacar este tema adelante, después quiero agradecer a mi familia que siempre han estado siempre presentes dándome fuerzas para seguir, a mi tutor por el apoyo y los conocimientos brindados para la paciencia en el tiempo que duró el desarrollo de este proyecto y los conocimientos recibidos.

RESUMEN

En el desarrollo de este proyecto se han analizado las herramientas gratuitas tanto como las pagadas para el desarrollo de aplicaciones y animaciones 3D, Se realizó la investigación del funcionamiento de las 2 tecnologías más importantes en cuanto a ejecución de animaciones como lo son WebGL, Flash Player.

También se realizó un ejemplo para la comparación del rendimiento de las 2 herramientas en dos ordenadores diferentes marcando la diferencia en performance entre una y otra, además de las conocidas como el uso de plugins, APIS y demás herramientas para mantener una ejecución y presentación lo más limpia ya llamativa para los usuarios.

El documento está pensado en las empresas con bajos recursos debido que son las que más le apunta a las herramientas gratuitas por lo que para ellas cada opción es una oportunidad de sobresalir en el mercado del diseño y animaciones y más aún en una tecnología que está en auge como lo es 3D en entornos web.

ABSTRACT

In the development of this project has analyzed the free tools as much as those paid for the development of applications and 3D animations, research was conducted in the functioning of two key technologies in terms of execution of animations such as WebGL, Flash Player.

Also we made an example to compare the performance of two tools on two different computers to make a difference in performance between the two, besides the well known as the use of plugins, APIs and other tools to maintain top performance and presentation clean and appealing to users.

The document is intended for companies with low income because they are the more points it at the free tools so each option for them is an opportunity to excel in the market of design and animation and even more on technology that was such as 3D boom in web environments.

Contenido

CAPITULO I	1
1. INTRODUCCIÓN	1
1.1 ANTECEDENTES	2
1.2 FORMULACIÓN DEL PROBLEMA.....	4
1.3 SISTEMATIZACIÓN	4
1.4 OBJETIVOS	7
1.5 JUSTIFICACION.....	8
1.6 ALCANCE Y LIMITACIONES.....	9
1.7 ESTUDIOS DE FACTIBILIDAD	10
CAPITULO II	13
2. MARCO DE REFRENCIA.....	13
2.1 MARCO TEÓRICO.....	13
2.2 MARCO CONCEPTUAL	14
2.3 MARCO LEGAL	19
2.4 MARCO ESPACIAL.....	19
CAPITULO III	20
3. METODOLOGIA.....	20
3.1 PROCESO DE INVESTIGACION	20
CAPITULO IV	23
4. ANÁLISIS DEL PROBLEMA.....	23
4.1 ANÁLISIS INICIAL.....	23
CONCLUSIONES Y RECOMENDACIONES.....	58
CONCLUSIONES	58
RECOMENDACIONES	60
ANEXOS	62
ANEXO 1: ABREVIACIONES Y ANGLICISMOS	62
ANEXO 2: COMPARACIÓN DE CÓDIGOS.....	63
Código Stage3D para Flash.....	63
Código JavaScript Cubos usando Three.js	67
ANEXO 2: ANÁLISIS FINANCIERO.....	70

CONTENIDO DE FIGURAS

Figura No. 1 Diagrama de actividades tecnología actual.	4
Figura No. 2 Diagrama de causa y efecto.....	5
Figura No. 3 Diagrama de actividades utilizando Html5.	7
Figura No. 4 3ds max Desarrollo animaciones.....	15
Figura No. 5 Diagrama de actividades tecnología actual	23
Figura No. 6 Responsabilidades actores del sistema	27
Figura No. 7 Caso de uso de la tecnología actual usuario.....	28
Figura No. 8 Caso de uso de la tecnología actual Jefe Departamental.....	28
Figura No. 9 Caso de uso de la tecnología actual Diseñador.	29
Figura No. 10 Diagrama de procesos del funcionamiento actual	30
Figura No. 11 Caso de uso de la tecnología propuesta Usuario.....	36
Figura No. 12 Caso de uso de la tecnología propuesta Jefe Departamental.	36
Figura No. 13 Caso de uso de la tecnología propuesta Diseñador.....	37
Figura No. 14 Diagrama de procesos de la tecnología propuesta	38
Figura No. 15 Pipeline programable - ES2.0.....	40
Figura No. 16 Vertex Shader – OpenGL ES2.0.....	41
Figura No. 17 Primitive Assemblies – OpenGL ES.	42
Figura No. 18 Rasterización OpenGL ES.	43
Figura No. 19 Fragments Shaders – OpenGL ES.....	43
Figura No. 20 Shared Memory Renderizado OpenGL ES 2.0	44
Figura No. 21 Soporte a WebGL.....	46
Figura No. 22 Porcentajes de uso versiones de Sistemas Operativos.....	48
Figura No. 23 Porcentaje uso de navegadores.....	49
Figura No. 24 Porcentaje uso versiones de navegadores.	49
Figura No. 25 Ejemplo Realidad Aumentada WebGL.....	51
Figura No. 26 Ejemplo Cubos 3D – AWAY 3D.....	56
Figura No. 27 Ejemplo cubos WebGL con Three.js	57

CONTENIDO DE TABLAS

Tabla 1 Gastos realizados para el desarrollo del proyecto.	13
Tabla 2 Diagnostico riesgos – impactos – soluciones.	23
Tabla 3 Teoría utilizada	24
Tabla 4 Actores del sistema.....	27
Tabla 5 Riesgos del sistema.....	31
Tabla 6 Porcentajes de uso Sistemas Operativos.....	47
Tabla 7 Tabla de compatibilidad Stage3D vs WebGL.....	55
Tabla 8 Tabla de precios de licencias adquiridas por primera vez.....	71
Tabla 9 Tabla de precios de licencias de actualizaciones.....	71
Tabla 10 Pagos de sueldos empleados.....	72
Tabla 11 Tabla de ingresos de la empresa StudioFX	72
Tabla 12 Pago capacitaciones al personal.....	73
Tabla 13 Ganancias de la empresa sin utilizar HTML5	73
Tabla 14 Ganancias de la empresa utilizando HTML5.....	73
Tabla 15 Costos de programas con licencias – Equilibrio.	74
Tabla 16 Sueldos a empleados de la empresa.	74
Tabla 17 Pagos capacitaciones al personal de la empresa – Equilibrio.	75
Tabla 18 Ingresos anuales de la empresa equilibrio utilizando licencias.....	75
Tabla 19 Pago licencias anuales empresa, usando HTML5.....	75
Tabla 20 Pagos capacitaciones al personal HTML5.....	76
Tabla 21 Ingresos de la empresa utilizando HTML5.....	77
Tabla 22 Comparación Ganancias entre software pagado y gratuito.....	77

CAPITULO I

1. INTRODUCCIÓN

Con el paso de los años nos hemos visto envueltos en diferentes cambios con respecto a lo tecnológico por ej.: celulares, computadores, etc.

Sin notar un cambio tan inmenso en lo que es la w3c¹ o internet, que sin quedarse atrás con respecto a otras tecnologías se encuentra en actualización con la tecnología denominada HTML5². La cual incorpora una inmensidad de etiquetas y funciones con respecto a sus antecesores, la actualización más importante realizada por esta tecnología es la denominada Canvas que es una etiqueta que permite ejecutar scripts³ de java en 2D y 3D igualando e incluso superando en algunos aspectos a programas existentes en la red los cuales son pagados y en su caso demasiado caros para ser adquiridos por las empresas.

También estos programas son extremadamente pesados para ejecutarse en computadores con pocos recursos, haciendo que los estudiantes y empresas se vean forzados a renovar sus equipos para poder acceder a dichas tecnologías, que por un lado las empresas están obligadas a realizar debido que se necesitan de recursos para ejecutar contenido 3D por los procesos que se realizan para la generación de las imágenes y el uso de memoria.

¹ W3C El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.

² HTML5 es la quinta revisión importante del lenguaje básico de la World Wide Web.

³ Script es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete

Lo que se busca con esta investigación es incentivar a las empresas a inmiscuirse en el uso de HTML5, debido a que es una tecnología gratuita y la misma se puede ejecutar directamente en la web sin la necesidad de programas de terceros, disminuyendo en gran cantidad los gastos por licencias.

Además de disminuir la carga de tráfico en las páginas web debido a que ya no se necesitan de plugins externos como flash player⁴ y otros, si no que directamente ejecuta javascript⁵ un lenguaje de programación ultraliviano y que es compatible con todos los navegadores acabando con los fantasmas de compatibilidad que existían con los programas para ejecutarse en diferentes plataformas.

1.1 ANTECEDENTES

Las animaciones realizadas por ordenador han tenido constantes actualizaciones tanto es así que pasamos de animaciones 2D a 3D en rangos de tiempo extremadamente cortos, se comenzaron con animaciones únicamente para plataformas de escritorio en la actualidad se está implementado esta tecnología en navegadores web, es decir que el usuario podrá ejecutar animaciones aplicaciones sin la necesidad de ejecutar ningún tipo de programa, en algunos casos lo único que necesitara es instalar un plugin que hará el trabajo de la animación.

Cuando se accede a internet y se navega por diferentes páginas web, existen páginas que para desplegar sus contenidos de manera parcial o

⁴ Flash Player es una aplicación en forma de reproductor multimedia creado inicialmente por Macromedia y actualmente distribuido Adobe.

⁵ JavaScript: es la implementación de Microsoft de ECMAScript.

total, se debe hacer uso de los llamados PLUGINS⁶", estos son complementos que deben ser utilizados en la mayoría de los casos por los usuarios, sin poder añadir funcionalidades a algún sistema y no siempre van a servir para poder desplegar el contenido de una página web, ya que en ocasiones presentan problemas de compatibilidad con el navegador, con el S.O (Sistema Operativo), con la seguridad, y además al ser ejecutados consumen muchos recursos en el computador ocasionando un bajo rendimiento o dejando que este se quede colgado. Flash es un ejemplo de un programa que requiere el uso de plugins, el cual tiene inconvenientes con algunos navegadores, ya que si un usuario desea ver una página web que tiene contenido hecho en este programa y no ha descargado e instalado dicho plugin, el contenido de la página no se podrá mostrar.

El VRML (tecnología con la cual se construyó una librería para contenido 3D en la web llamada Glypher3D) X3D, X3DOM, Firefox Canvas⁷ 3D, Opera Canvas 3D, O3D, etc.

Todas estas tecnologías están relacionadas de alguna manera con los plugins. Entre la lista de este tipo de herramientas se encuentra una nueva tecnología que permite el manejo de este mismo contenido 3D en la web, con la diferencia de que no hace uso de plugins. Es el estándar WebGL⁸, el cual es una aplicación nativa de los navegadores Gecko y de los navegadores basados en WEBKIT. El hecho de no trabajar con plugins brinda al usuario la facilidad de trabajar en la web, sin la necesidad de

⁶ Plugin Es un complemento es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

⁷ Canvas es una etiqueta o elemento en HTML que permite la generación de gráficos en forma dinámica por medio de programación

⁸ WebGL es una especificación estándar que está siendo desarrollada actualmente para desplegar gráficos en 3D en navegadores web.

descargar e instalar el plugin⁶ compatible con el programa, para después poder llevar a cabo la ejecución de algún proceso.

En el país se han desarrollado proyectos tales como “**MODELO TRIDIMENSIONAL DE LA HISTORIA GEOLOGICA DEL VOLCAN COTOPAXI**”⁹, este proyecto ha sido desarrollado “**Cristian Charro Arévalo**” y “**Vinicio Washington Valencia Armijos**”, fue desarrollado en flash, para luego ser implementado en una página Web creada por ellos la misma que necesita que el navegador cuente con el plugin Flash Player⁴ para ejecutarse, este proyecto en la actualidad se puede desarrollar perfectamente en HTML5² con Canvas⁷ y WebGL⁸ el mismo que no necesitaría de ningún plugin instalado para su ejecución.

1.2 FORMULACIÓN DEL PROBLEMA

¿El estudio de rendimiento de HTML5² y sus Apis¹⁰ para desarrollo 3D en la Web ayudaría a las empresas o mejorar su posicionamiento tecnológico?

1.3 SISTEMATIZACIÓN

1.3.1 DIAGNOSTICO

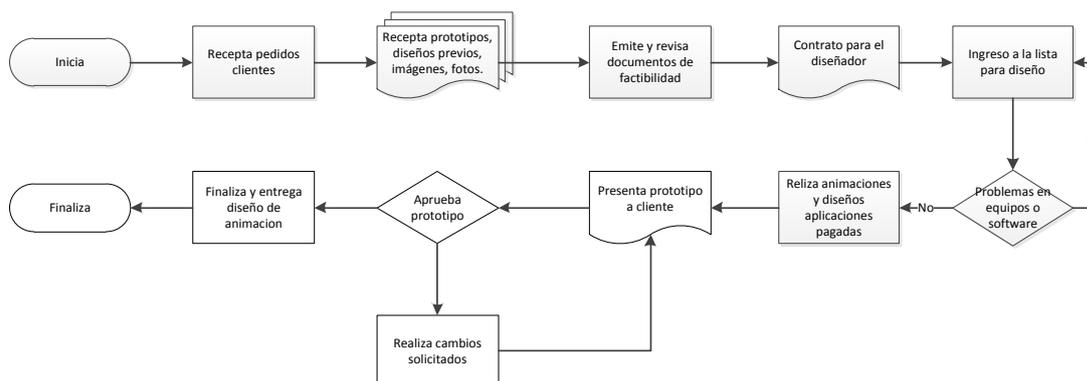


Figura No. 1 Diagrama de actividades tecnología actual.

⁹ MODELO TRIDIMENSIONAL DE LA HISTORIA GEOLOGICA DEL VOLCAN COTOPAXI.

URL: <http://bibdigital.epn.edu.ec/bitstream/15000/360/1/CD-0779.pdf> - Obtenido Diciembre 20, 2012

¹⁰ API representa un interfaz de comunicación entre componentes software.

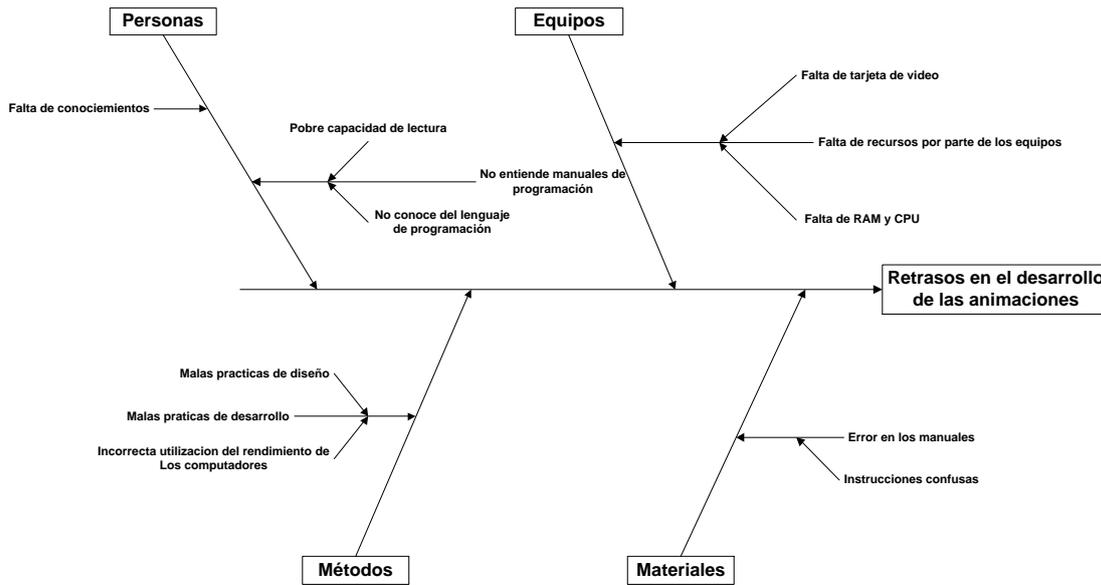


Figura No. 2 Diagrama de causa y efecto.

- El uso de flash supone la necesidad de instalación de plugins para su funcionamiento.
- WebGL⁸ y las animaciones Flash: El diseño y construcción de animaciones sencillas en HTML5² con Canvas⁷ supone unas 10 veces el tiempo que supone hacerlo en Flash, con personal que antes no haya tenido ningún tipo de capacitación.
- La creación de aplicaciones y animaciones equivocadas para dispositivos sabiendo que existen diferencias entre cada uno de ellos por ejemplo entre un ultra portátil, un Tablet y un Smartphone, las prestaciones son inversamente proporcionales a la movilidad, por lo que se pensarían en distintas animaciones para cada dispositivo.

1.3.2 PRONÓSTICO

- El seguir utilizando plugins para la ejecución de animaciones 3D nos mantiene atados a una sola tecnología, quitándonos la posibilidad de probar nuevas y mejores tecnologías, además de

tener problemas con programas no requeridos y virus por parte de los usuarios.

- La no capacitación del personal hará que los planes de trabajo no se cumplan debido a que no se encuentran preparados para el desarrollo en aplicaciones desconocidas por lo que les tocara estudiar primero la aplicación, su lenguaje necesitaran descargar manuales, etc. Los mismo que muchas veces se encuentran en lenguajes de extranjeros lo que tomara el doble o triple de las horas de trabajo planeadas para desarrollar una sola aplicación, no siendo así con el pago de una capacitación los desarrolladores estarán preparados para cumplir con los tiempos propuestos en los planes de trabajo.
- El no separar a los dispositivos en categorías haría que se desarrollen aplicaciones incompatibles para algunos, lo que terminaría en una pérdida de tiempo y recursos ya que si no se paga por la herramienta de desarrollo, si se paga al personal por lo que antes de empezar se debería realizar un estudio de los dispositivos más utilizados por los clientes para desarrollar las aplicaciones fijas para los mismos.

1.3.3 CONTROL DEL PRONÓSTICO

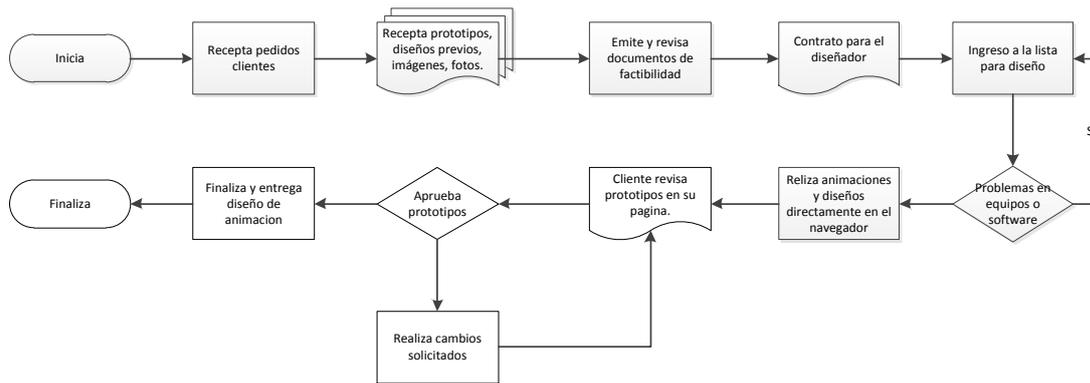


Figura No. 3 Diagrama de actividades utilizando HTML5.

- Con el uso de HTML5² y sus Apis no se necesita de ninguna instalación de plugins por lo que el usuario no instalara programas no deseados ni virus.
- Al realizar capacitación de las persona encargadas del diseño de aplicaciones en 3D en HTML5, ganamos personal con conocimientos suficientes para la creación de páginas Web y conocimientos de las Apis¹⁰ de HTML5.
- Al utilizar la nueva tecnología HTML5 para ejecutar aplicaciones estas se ejecutara en todos los equipos y dispositivos que lo soporten.

1.4 OBJETIVOS

1.4.1 OBJETIVO GENERAL

Realizar una investigación comparativa entre las herramientas de desarrollo 3D pagadas, con HTML5 y sus APIS.

1.4.2 OBJETIVOS ESPECÍFICOS

- Investigar sobre la especificación estándar WebGL⁸, Canvas⁷ de HTML5 y los conceptos programables que se utilizan en las etapas de proyección de esta tecnología.

- Investigar sobre la tecnología usada por flash, principal competidor de WebGL⁸ en esta rama de la tecnología.
- Demostrar a las empresas, que existen posibilidades de software gratuitas para determinadas tareas.

1.5 JUSTIFICACION

1.5.1 JUSTIFICACIÓN TEÓRICA

La teoría estructural funcionalista fue desarrollada por Lasswell; ésta, al igual que casi todas las teorías de la comunicación, hablaba del poder de los medios de comunicación frente al público.

Esta teoría tiene como principio lograr la persuasión por medio de las siguientes preguntas: quién dice que, a través de qué medio, a quién y con qué efecto, preguntas que, en últimas, se referían, sobre todo, al poder político de los medios y al análisis de contenidos de lo que transmitían o emitían. En ella se prestaba especial atención a la superficie de los discursos; es decir, a lo que la información podía lograr en un primer momento.

En la investigación se realizara la comparación entre las tecnologías de WebGL y Flash que son las orientadas a entornos web, con esto podremos responder las preguntas que plantea Lasswell en su teoría.

1.5.2 JUSTIFICACIÓN PRÁCTICA

La investigación de las tecnologías que se está realizando tendrá la capacidad de informar a los clientes las posibilidades que poseen en cuanto a software libre teniendo en cuenta que estas tecnologías siempre están en constante evolución dándonos la posibilidad de seguir mejorando continuamente.

La comparación entre las tecnologías se realizara sin problemas debido a que se poseen las herramientas necesarias para ello, además de tener los conocimientos en cuanto a usabilidad de los mismos.

Con la comparación de las tecnologías esperamos que los usuarios no noten diferencias negativas si no un mejor rendimiento de las mismas.

Con esto se tratara de ganar la aceptación de la mayoría de los usuarios que optan por tecnologías pagadas.

1.5.3 JUSTIFICACIÓN METODOLÓGICA

La experimentación científica, muchos de nuestros conocimientos nos lo proporciona la experiencia y es un método que te permite sentirte más seguro de lo que se está haciendo. A demás admite la modificación de variables, lo cual nos da vía libre para la corrección de errores y el mejoramiento de nuestra investigación. También podríamos agregar que como futuros ingenieros de sistemas aplicamos mucho este método, puesto que debemos buscar una solución de calidad, efectiva, funcional y de satisfacción a las necesidades del cliente, un ejemplo muy común en nuestra área seria la implementación de un software, siempre se realizan muchas pruebas.

1.6 ALCANCE Y LIMITACIONES

1.6.1 ALCANCE

El alcance del proyecto va abarcar la comparación de animaciones 3D entre las teologías mencionadas, por lo que no se tomaran en cuenta diseños de imágenes, aplicaciones, realidad aumentada que también forman parte de investigación más no de desarrollo para una posterior comparación.

1.6.2 LIMITACIONES

Las posibles limitaciones para elegir entre tecnologías, podrían ser que las empresas ya posean infraestructuras dedicadas para una sola tecnología haciendo imposible el cambio, otro punto en contra sería que el personal este demasiado desactualizado en las nuevas tecnología haciendo que las capacitaciones realizadas sean poco productivas debido a que no tienen las bases para realizar o producir animaciones en esta tecnología por lo que la empresa preferirá continuar con la ya aplicada.

En otros casos existirán empresas que tengan la capacidad de migrar pero desistirán de las mismas debido a que con el uso de tecnologías pagadas la empresa funciona de lo mejor y no estarán de acuerdo en correr riesgos incluyendo tecnologías desconocidas para ellos.

1.7 ESTUDIOS DE FACTIBILIDAD

1.7.1 TÉCNICA

WebGL⁸ como lo comentamos anteriormente es una herramienta que hace uso de la herramienta OpenGL¹¹ que fue desarrollada para entornos 3D pero en escritorio. WebGL mediante Canvas⁸ realiza la comunicación con OpenGL realizando las animaciones dentro del navegador web sin la necesidad de instalar plugins haciendo más amigable y fácil de usar las aplicaciones creadas, cabe desatacar que en el caso de usar APIS¹⁰ como lo es Three.js¹² librería JavaScript para

¹¹ OpenGL (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D.

¹² Three.js es la biblioteca JavaScript que está haciendo furor en la comunidad de desarrolladores de juegos y animaciones 3D.

renderización de 3D por lo que todo el código manejado es en base a este lenguaje.

Por otro lado tenemos Adobe Flash, que hace uso de la API¹⁰ AWAY3D¹³ para comunicarse con DIRECTX¹⁴ y así ejecutar las animaciones 3D, y visualizarlas en el navegador web haciendo uso de la su plugin Flash Player que desde la versión 11, permite la ejecución de aplicaciones 3D.

Para realizar la investigación de este tema tan importante para las empresas como lo es que tipo de tecnología aplicar en sus negocios, se poseen los conocimientos necesarios para la herramienta denominada HTML5², en lo que se necesita ayuda es en los sistemas de desarrollo 3D, por lo que se requerirá el apoyo de personas alternas a la empresa y al grupo tecnológico, buscando apoyo de personal de otras carreras como lo sería diseño, arquitectura, para analizar el desempeño de estas herramientas a nivel económico y empresarial.

1.7.2 OPERATIVA

Para que la empresa pueda realizar un análisis de las herramientas gratuitas se realizarán capacitaciones al personal de la empresa para que la migración sea lo más limpia posible sin causar daños ni pérdidas a la empresa por lo que se realizaran estudios de factibilidad operativa para cada uno de los integrantes del departamento de la empresa.

Para realizar la implementación de la nueva tecnología se debe dar a conocer los beneficios de la misma a los usuarios para que estén al

¹³ Away3D Es una librería utilizada por Flash para generar animaciones 2D y 3D.

¹⁴ DirectX es una colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo.

tanto del tema y no se un cambio brusco para ellos, por el costo de la aplicación debido a que pueden pensar que por ser gratuita les pueden causar problemas o fallar en la implementación, cosa que no sucederá si se realizan las capacitaciones pertinentes al grupo de trabajo.

Con esto el equipo de trabajo y empresa estarán listos para la implementación y uso de la nueva herramienta.

También se tiene en cuenta que la herramienta gratuita entraría a competir directamente con Flash Player⁴ y Adobe Flash que son herramientas destinadas al entorno web, por lo que no causara mayor cambio en cuanto a diseño 2D y 3D para entornos de escritorio, también observando que Adobe Flash es la herramienta más barata en cuanto a licencias, por lo que la implementación puede ser llevada a cabo manteniendo una fusión entre las tecnologías para sacarle el máximo provecho de las mismas, lo que se busca con el sistema gratuito es que los usuario no necesiten instalar plugins en sus páginas web, para hacer uso de las animaciones debido a que en algunos casos no serán capaces de descargar e instalar dichos plugins.

1.7.3 ECONÓMICA

Para la investigación de este tema no se necesitan de demasiados recursos económicos, por lo que no será un impedimento sino un pro debido a que este recurso no gastado en tecnología se lo puede aplicar para lograr encontrar otros tipos de ayuda para la investigación como lo sería personal para cada área o en caso de que no se desee se podría buscar alumnos que realicen pasantías ayudando en la investigación.

Los gastos serían los siguientes:

Gastos	Precios
Impresiones	50\$
Copias	50\$
Internet	100\$
Hojas	20\$
Esferos	5\$
Lápices	5\$
Borradores	5\$
Transporte	100\$
Total	335\$

Tabla 1 Gastos realizados para el desarrollo del proyecto.

CAPITULO II

2. MARCO DE REFERENCIA

2.1 MARCO TEÓRICO

Teoría conductista: Se centra en la conducta observable intentando hacer un estudio totalmente empírico de la misma y queriendo controlar y predecir esta conducta.

Aplicación multimedia

Se tomaron las características de un ejemplo para una capacitación de una empresa.

- El formato es de libro electrónico, donde existen sonidos, dibujos, imágenes, etc.
- Se tiene enlaces con diccionario, bibliografías, etc.
- Resalta los textos ya consultados.
- Cuenta con una herramienta de búsqueda.
- Es interactivo el aprendizaje.

- Se puede introducir texto como comentarios.
- Cuenta con una evaluación, la cual después de haber sido contestada, emite una calificación mostrando las respuestas bien contestadas como las que están mal.

Se estudiara la teoría de **paradigma funcionalista** que hace alusión a que la sociedad es algo funcional y que se va compensando una cosa con la otra; es decir, ha de existir un pobre para que haya un rico. Todo está relacionado para formar una unidad.

Las teorías a usar para lograr terminar con la investigación serán la de los programas involucrados en el tema los mismo que serían los pagados herramientas de desarrollo que su fin es lucrar por los beneficios que entregan a las empresas para diseñar sus modelos, y las herramienta gratuita que están surgiendo con fuerza como lo es HTML5² con sus respectivas APIS, que están entregando el mismo tipo de modelados pero sin costo por sus beneficios.

2.2 MARCO CONCEPTUAL

2.2.1 HERRAMIENTAS DE 3D PAGADAS.

3DS MAX

Es una aplicación orientada a objetos para el diseño en 3 Dimensiones desarrolladas por kinetex, puede tomar objetos a partir de los modelos a otros formatos distintos del que maneja 3D Studio Max, basado fundamentalmente en el sistema operativo Windows 9x/NT. Este producto permite integrar las escenas creadas en 3D con los personajes u objetos reales, tiene su funcionamiento sobre sistemas que sean

multiprocesador y tenga conexión de dispositivos como mouse, teclados, etc., siendo así una de las aplicaciones más utilizadas en el desarrollo de videojuegos se han creado títulos como Tom Raider, Halo, etc.



Figura No. 4 3ds max Desarrollo animaciones

<http://bit.ly/WmCzAS>

3D Studio Max posee un gran número de Plugins además de ser potente y estable, pero lo que no lo hace tan acatable es su alto precio por la aplicación, los requisitos de hardware son muy elevados y no están disponibles para todos los usuarios.

FLASH PLAYER Y ADOBE FLASH

Un reproductor de películas Flash, que es un tipo de archivo muy conocido por la mayoría de los navegadores. Es, sin duda, la herramienta más útil para jugar video juegos online y el plugin más importante a la hora de ir a una página creada en Flash. Se lo puede descargar, gratuitamente, para cualquier navegador. Los archivos flash se crean con Adobe Flash.

Flash Player⁴, está desarrollado en versiones para todos los idiomas y es el plugin más simple creado por Adobe. Sin duda, es una herramienta

necesaria para el buen uso de un navegador y para una buena visualización de una página web.

Para realizar este tipo de proyectos se pueden realizar mediante adobe flash el precio de este programa es de 700\$.

2.2.2 HERRAMIENTAS GRATUITAS

HTML5

HTML5 es la actualización más reciente lanzada por parte de la w3c de HTML. Este lenguaje especifica dos variantes de sintaxis, la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML. Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Todavía se encuentra en modo experimental, esto fue indicado por la propia W3C; aunque ya es usado por múltiples desarrolladores web por sus avances, mejoras y ventajas.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se le recomienda al usuario común actualizar a la versión más nueva de su navegador preferido, para poder disfrutar de todo el potencial que provee.

OPENGL ES 2.0

En todo lo relacionado al funcionamiento interno de WebGL⁸ tenemos que referenciarlos a OpenGL¹¹ ES 2.0. La Open Graphics Library (OpenGL) es una librería usada para visualizar datos 2D y 3D, posee múltiples propósitos que soporta aplicaciones para la creación de

contenido digital tales como diseño mecánico y arquitectónico, prototipos virtuales, simulaciones, videojuegos, etc.

En el mundo del desarrollo 3D existen 2 APIs¹⁰ encargadas del renderizado de las imágenes las mismas son: DirectX¹⁴ y OpenGL¹¹.

DirectX es la principal herramienta utilizada para la ejecución de animaciones en sistemas que usan Windows como sistema operativo y es la normalmente usada por la mayoría de juegos de esta plataforma.

OpenGL es multi-plataforma usada mayoritariamente en sistemas Linux, Mac OS X y en algunos casos Microsoft Windows.

OpenGL ES: es una versión simplificada de la existente librería de OpenGL para sistemas más pequeños que contienen muchas más restricciones que la mayoría de plataformas.

La versión ES (Embedded Systems) está dirigida a dispositivos con limitada capacidad de proceso y memoria, poco ancho de banda, sensibilidad al gasto de energía y falta de Hardware. Ha sido diseñada según estos criterios:

- OpenGL es una librería muy grande y compleja y la misma que se desea adaptar para dispositivos restringidos de hardware. Para conseguirlo se ha tenido que suprimir toda la redundancia existente en la librería. Un buen ejemplo es que la geometría en OpenGL se puede usar en Modo Inmediato, Display Lists o Vertex Arrays. Sin embargo en el caso de OpenGL ES solo se pueden usar Vertex Arrays.

- Eliminar la redundancia es importante pero mantener la compatibilidad con OpenGL también. En la medida de lo posible se ha intentado que la funcionalidad de OpenGL ES sea un subconjunto de las funcionalidades de OpenGL para la compatibilidad de las aplicaciones.
- Se han añadido nuevas características para controlar el consumo de energía.

Por ejemplo la precisión de los datos en los Shaders¹⁵. La precisión de los datos puede incrementar o disminuir el rendimiento de los Shaders y por lo tanto el consumo energético.

- Se han preocupado de ofrecer un mínimo de características para la calidad de las imágenes. Muchos dispositivos móviles tienen pantallas pequeñas por lo que requieren que la calidad de los píxeles dibujados sea la mejor posible.

CANVAS

El elemento <Canvas> HTML5² se utiliza para dibujar gráficos, sobre la marcha, a través de secuencias de comandos, normalmente JavaScript.

El elemento <Canvas> es sólo un contenedor para gráficos. Se debe utilizar una secuencia de comandos para dibujar en realidad los gráficos.

Canvas tiene varios métodos para rutas de dibujo, cajas, círculos, personajes e imágenes.

¹⁵ Shaders es cualquier unidad escrita en un lenguaje de sombreado que se puede compilar independientemente.

JQUERY

jQuery es un framework¹⁶ Javascript, producto que sirve como base para la programación avanzada de aplicaciones, que aporta una serie de funciones o códigos para realizar tareas habituales. Por decirlo de otra manera, framework son unas librerías de código que contienen procesos o rutinas ya listos para usar. Los programadores utilizan los frameworks para no tener que desarrollar ellos mismos las tareas más básicas, puesto que en el propio framework ya hay implementaciones que están probadas, funcionan y no se necesitan volver a programar.

2.3 MARCO LEGAL

En el caso que las empresas desearan hacer uso de estas tecnologías, se debe estudiar las posibilidades legales debido a que pudieron haber firmado contratos con las empresas dueñas de las herramientas de diseño, por lo que tendrán que esperar a que terminen las cláusulas del mismo para poder cambiar de tecnología.

Al migrar de tecnología las empresas de igual manera tendrán que atenerse a las leyes de derecho de copyright.

2.4 MARCO ESPACIAL

La investigación que se realiza en este documento se tratara de aplicar en la empresa que laboro “**Mega Hierro**” que tiene su página desarrollada en **Flash**, se tratara de desarrollar una aplicación y

¹⁶ Framework es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado.

presentarla como parte de prototipo para pruebas y futuras incorporaciones en la empresa.

CAPITULO III

3. METODOLOGIA

3.1 PROCESO DE INVESTIGACION

3.1.1 UNIDAD DE ANALISIS

El lugar de aplicación de la investigación será en las empresas publicas debido a los escasos recursos que estas tienen para conseguir sistemas pagados de desarrolló 3D, también haciendo hincapié en que los desarrolladores busca más puestos en empresas públicas.

La información que se estudiara para la aplicación del desarrollo será tomada de empresas que hayan empezado hacer uso de tecnologías gratuitas pensando en un futuro en el cual ya no se necesite pagar sabiendo que existen opciones de la misma calidad y en algunos casos de mejor desempeño y valor.

3.1.2 TIPO DE INVESTIGACION

La investigación documental se la utilizara como parte esencial de un proceso de investigación científica, constituyéndose en una estrategia donde se observa y reflexiona sistemáticamente sobre realidades (teóricas o no) usando para ello diferentes tipos de documentos.

Esta investigación permite obtener nuevos conocimientos en el campo de la realidad social, o bien estudiar una situación para diagnosticar

necesidades y problemas a efectos de aplicar los conocimientos con fines prácticos (investigación aplicada).

3.1.3 METODO

El objetivo de la investigación es brindar a los responsables de las áreas investigadas un camino estructurado por el que arriben a los resultados esperados por la empresa siguiendo un plan.

La experimentación científica, muchos de nuestros conocimientos nos lo proporciona la experiencia y es un método que nos permite sentirnos más seguros de lo que se está haciendo. Además admite la modificación de variables, lo cual nos da vía libre para la corrección de errores y el mejoramiento de nuestra investigación. También podríamos agregar que como futuros ingenieros de sistemas aplicamos mucho este método, puesto que debemos buscar una solución de calidad, efectiva, funcional y de satisfacción a las necesidades del cliente, un ejemplo muy común en nuestra área sería la implementación de un software, siempre se realizan muchas pruebas.

3.1.4 TECNICA

Las técnicas que utilizaremos para la investigación serán:

Experimentación, observación. Además de utilizar elementos físicos que nos ayude a probar las técnicas; laptop, papel. Necesitaremos habilidades personales como la vocación para charlar, para mantener contacto con las persona encargas del desarrollo de las aplicaciones.

Conocimientos técnicos y administrativos, llevar lo que se requiere.
Conocer el negocio que se va auditar.

Actualización permanente: constantemente van saliendo avances de las tecnologías existentes por lo que se necesita actualizarse sobre ello.

3.1.5 INSTRUMENTO

Observación: Que es el instrumento de investigación por excelencia, debido a que esta incluye procedimientos informales hasta los más sistematizados.

La más importante ventaja es que hace posible obtener información del comportamiento tal y como ocurre. La contraparte es que el observador necesita presenciar el fenómeno y esto muchas veces es imposible

La observación tiene diferentes objetivos:

- Recabar información propia para la investigación.
- Establecer una comparación entre las encuestas realizadas o complementar estos datos.

CAPITULO IV

4. ANÁLISIS DEL PROBLEMA

4.1 ANÁLISIS INICIAL

4.1.1 COMO FUNCIONA ACTUALMENTE

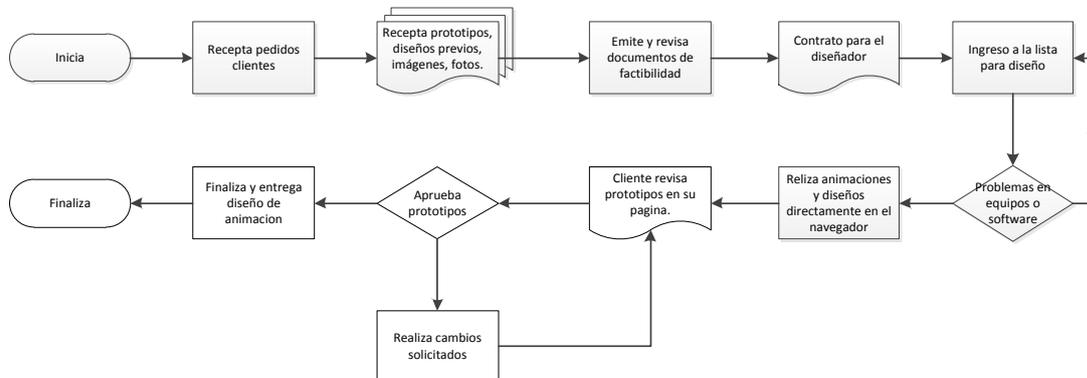


Figura No. 5 Diagrama de actividades tecnología actual

4.1.1.1 Documento de Visión

<p>Problema Afecta a</p>	<ul style="list-style-type: none"> • Utilización de plugins para la ejecución de aplicaciones 3d. • Tiempo de desarrollo 10 veces mayor usando HTML5² que Flash, en el caso de que no se tengan los suficientes conocimientos. • Desarrollo de animaciones para equipos y dispositivos equivocados supone una pérdida de tiempo
<p>Impacto</p>	<ul style="list-style-type: none"> • Instalación de programas innecesarios o en algunos casos virus al descargar el plugin. • La no capacitación del personal hará que los tiempos de desarrollo sigan en crecimiento en vez de disminuir debido al desconocimiento de la tecnología. • El no estudiar los equipos disponibles y sus capacidades para ejecutar aplicaciones 3D, hará que se desarrollen equivocadas para equipos y dispositivos.
<p>Solución</p>	<ul style="list-style-type: none"> • Usando HTML5 no se necesita de ningún plugin para ejecutar animaciones 3D. • Se ganan conocimientos y mejor personal al capacitarlos en la nueva tecnología. • Las aplicaciones se ejecutaran en todos los equipos y dispositivos que soporten WebGL⁸.

Tabla 2 Diagnostico riesgos – impactos – soluciones.

Para	Análisis de software de desarrollo 2D - 3D vs HTML5 ² – APIS y su posicionamiento tecnológico
Quien	Mega Hierro
Nombre Producto	Adobe Flash
Que	Animaciones 3D
Nuestro Producto	HTML5 y sus Apis ¹⁰

Tabla 3 Teoría utilizada

4.1.1.2 HTML5 y Apis

La ventaja que nos da utilizar HTML5, trae consigo infinidad de beneficios nos da la posibilidad de utilizar diferentes Apis y programas para desarrollar aplicaciones 3D, tenemos distintas Apis por ej.: CSS3¹⁷, Three.js¹², Canvas⁷.

Las opciones que encontramos son realmente sorprendentes y marcan una importante evolución en lo que nos ofrece HTML5. En el mundo Mobile, WebGL está entrando, pero hay que tener muy en cuenta que es una característica que puede bajar considerablemente el nivel de batería.

El universo de posibilidades que nos abre WebGL es inmenso y parece ser maravilloso, pero si pensamos en aplicarlo a nuestros proyectos debemos tener en cuenta: compatibilidad, rendimiento y seguridad.

Nuevas Apis

- El protocolo HTTP permite a las computadoras conectarse a los servidores, los cuales envían archivos (HTML, CSS, imágenes, etc.) a los clientes. Es un proceso de ida y vuelta. El HTTP no permite conexiones continuas de forma nativa. Una vez que se envía algo, el

¹⁷ CSS3 Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML).

cliente deja de estar conectado al servidor, por lo tanto los datos y las interacciones no pueden fluir de forma continua.

- Los WebSockets ¹⁸ mejoran el HTTP ¹⁹ para crear verdaderas conexiones bidireccionales que pueden mantener los datos fluyendo entre el cliente y el servidor. También funciona con múltiples clientes (y/o dispositivos), por lo tanto se pueden hacer conexiones “cross-device” en tiempo real. El HTTP es como usar “walkie-talkies” para comunicarse, mientras que los Web Sockets es como utilizar teléfonos.
- Tradicionalmente, los servidores guardan los datos para los sitios y aplicaciones web. Para poder leer o escribir estos datos los clientes tienen que conectarse con los servidores.
- Hoy en día es posible configurar una base de datos en el cliente, no solo en el servidor. Los datos pueden ser salvados y leídos desde estas bases de datos en el cliente. Local storage, session storage, y Index DB son ejemplos tempranos de bases de datos locales.
- Una de las grandes hipótesis de la Web es que tienes que estar en línea para usarla. Sin embargo, en el futuro cada vez más interacciones serán posibles sin estar conectados. “Cache manifest” te permite guardar información en el cliente para acceder sin estar “online”.
- Los archivos en el sistema local no se integran fácilmente con los navegadores Web. Pero ahora hay siete nuevas Apis para tratar con

¹⁸ WebSockets es una tecnología que hace posible abrir una sesión de comunicación interactiva entre el navegador del usuario y un servidor.

¹⁹ El protocolo de transferencia de hipertexto (HTTP, HyperText Transfer Protocol) es el protocolo usado en cada transacción de la Web (WWW).

los sistemas de archivos en la Web: file API¹⁰, reader/writer/system, blob URLs & builder, drag and drop. A través de estas APIs, los archivos pueden no solo ser usados en el navegador sino que también pueden ser pareados, manipulados, escritos y salvados. Incluso se puede extraer la información que está dentro de estos archivos: metadatos, tamaño del archivo, todo en el cliente.

- `getUserMedia` es una API de javascript⁵ que recoge y crea transmisiones de audio y video desde los periféricos de tu computadora (cámara, micrófono) en el navegador web. Con transmisiones de medios, podemos mostrar y manipular feeds en tiempo real de audio y video en las páginas web.
- El Web audio permite reproducir archivos en el navegador pero también se puede hacer uso de efectos como reverb, eco, etc.
- APIs de dispositivo: permite conectarte con las capacidades de los dispositivos, como la vibración, el estado de la batería, lista de contactos y más, en el navegador Web.
- Canvas: un espacio en una página Web donde se puede dibujar formas, colores, etc. También puede funcionar de forma invisible para hacer cálculos detrás de escena para elementos de dibujo.
- WebGL: una API para hacer dibujos 3D en el navegador Web.
- Controles: no todo el mundo tiene un teclado y un mouse. Los navegadores ahora también están disponibles en consolas de videojuegos que tienen controles únicos, así como “touch screens” y más. Esta API te permite usar todo el rango de entrada en estos dispositivos.

Actor	Responsabilidad	Actividad
Gerente	Velar por el bienestar de la empresa.	Aprobar capacitaciones y control de personal
Jefe departamental	Responsable de los diseñadores y del departamento.	Encargado de la planificación de responsabilidades de cada uno de los diseñadores.
Diseñadores	Cumplir con las responsabilidades encargadas.	Realizar animaciones encomendadas por parte del jefe departamental.

Tabla 4 Actores del sistema.

4.1.1.3 Modelo de Negocio

 Gerente	RESPONSABILIDADES
	<ul style="list-style-type: none"> - Gestión del cambio y desarrollo de la empresa. - Habilidades cognitivas. - Habilidad comunicativa. - Liderazgo - Motivación y dirección del personal - Integridad moral y ética.
 Usuario	RESPONSABILIDADES
	<ul style="list-style-type: none"> - Necesidades de información. - Necesidad de productos. - Aplicación de nuevas tecnologías.
 Jefe Departamental	RESPONSABILIDADES
	<ul style="list-style-type: none"> - Responsable de la empresa. - Conducción, ejecución y supervisión de tareas de los empleados. - Registro de contratos de usuarios - Encargado de planeación de desarrollo de los productos
 Diseñadores	RESPONSABILIDADES
	<ul style="list-style-type: none"> - Diseño de imagen corporativa. - Diseño de folletos. - Diseño de logotipos. - Diseño 3D - Retoque gráfico.

Figura No. 6 Responsabilidades actores del sistema

4.1.1.4 Caso de uso del negocio

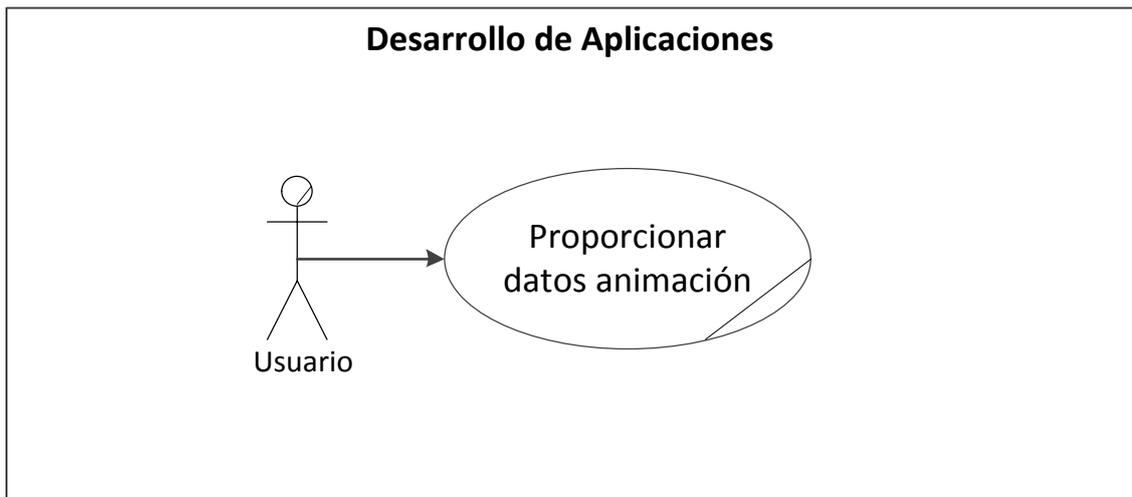


Figura No. 7 Caso de uso de la tecnología actual usuario.

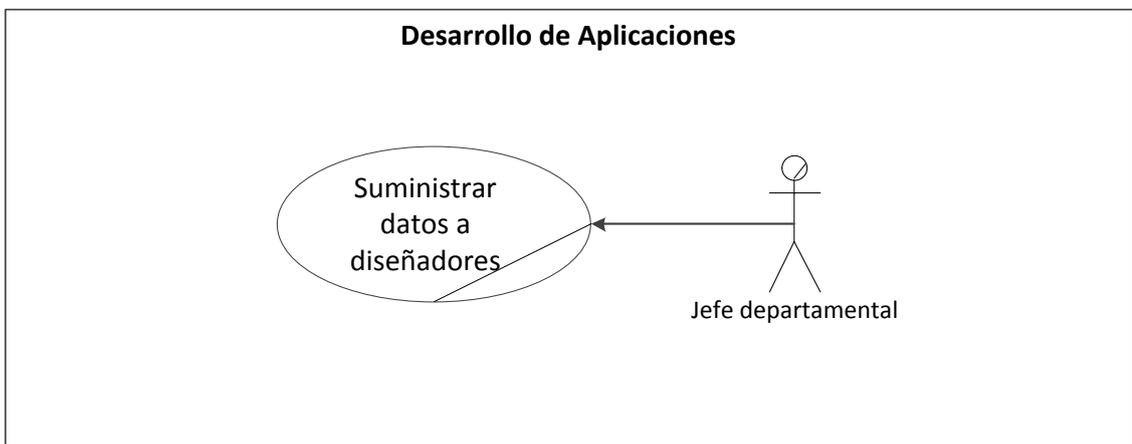


Figura No. 8 Caso de uso de la tecnología actual Jefe Departamental.

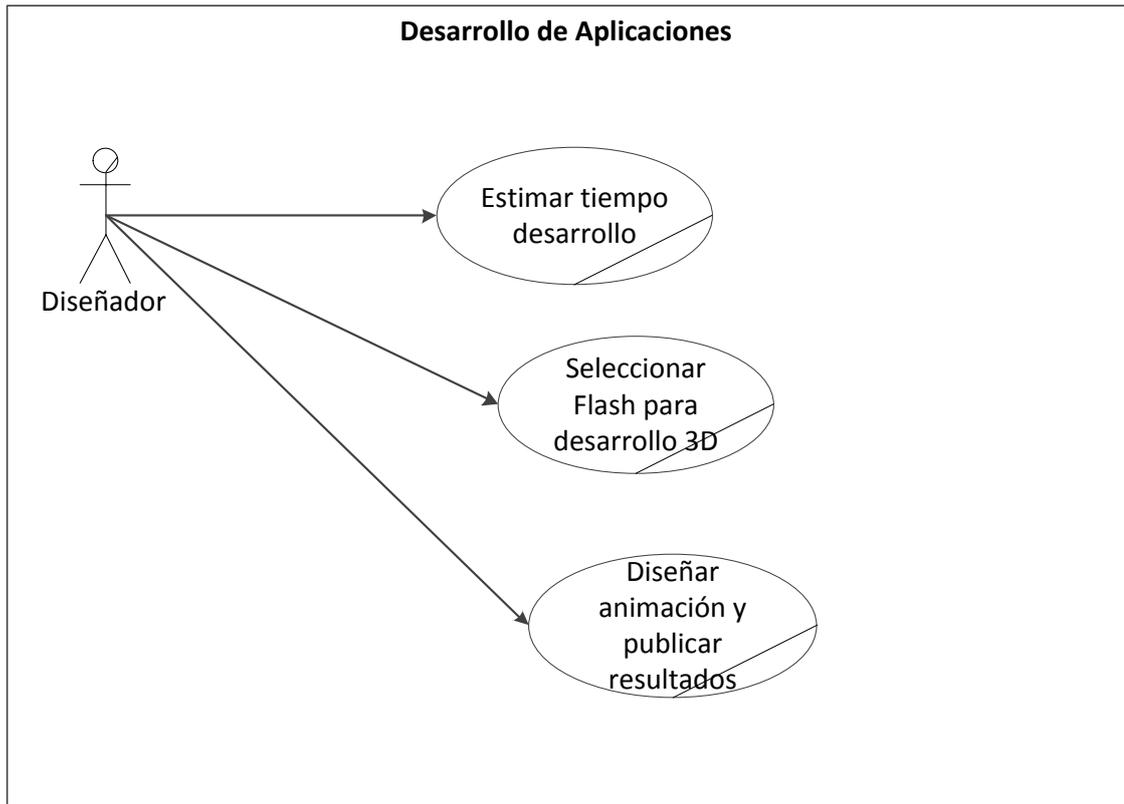


Figura No. 9 Caso de uso de la tecnología actual Diseñador.

4.1.1.5 Diagrama de Actividades

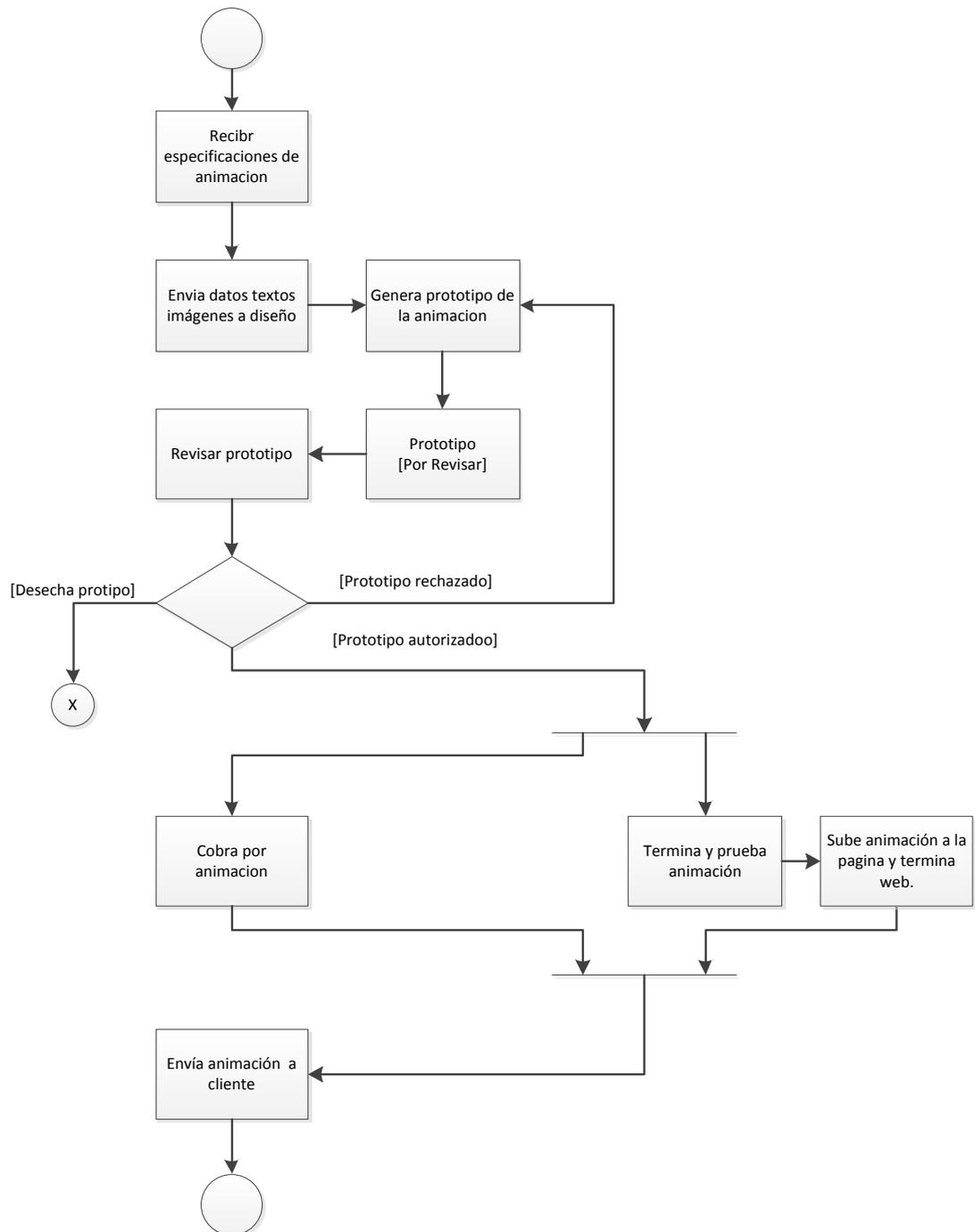


Figura No. 10 Diagrama de procesos del funcionamiento actual

4.1.1.6 Análisis de Riesgos

Riesgos del proyecto

Incremento en costes: El proyecto está basado en precios para el desarrollo de las animaciones, por lo que el cambio de algún precio significa un cambio en los gastos.

Riesgos técnicos

Equipos y dispositivos: Incapacidad de conseguir equipos para realizar las respectivas pruebas de las animaciones desarrolladas para la comparación de las tecnologías.

Personal: Que el personal sea incapaz de comprender la nueva tecnología, incapacitando el desarrollo de las animaciones para las respectivas pruebas dentro de la empresa.

Riesgos de negocio

Presupuesto: El desarrollo de la investigación está pensada para realizarse con un presupuesto ya estudiado, en este caso pueden ocurrir riesgos de levantamiento de precios en ciertos productos o estudios, comprometiendo los tiempos del estudio.

4.1.1.7 Valoración de riesgos

- **Baja.-** Es la que tiene repercusiones sólo en la operación diaria y se puede solucionar sin muchos inconvenientes.
- **Media.-** Es la que causa daños que tardan un tiempo, pero no comprometen el desarrollo del proyecto.
- **Alta.-** En caso de no ser solucionados pueden causar el fin del desarrollo del proyecto.

Descripción	Prioridad
Riesgos del Proyecto	
• Incremento en costes	(Bajo)
Riesgos Técnicos	
• Equipos y dispositivos	(Medio)
• Personal	(Alto)
Riesgos de Negocio	
• Presupuesto	(Alto)

Tabla 5 Riesgos del sistema

4.1.1.8 Medios de control

Control Riesgos del proyecto

Incremento en costes: Se desarrollara una planificación económica que cuente con la capacidad de acoplarse a cambios de precios.

Control de riesgos técnicos

Equipos y dispositivos: Se realizara el desarrollo de animaciones específicas para los equipos con los que se cuenta para evitar problemas de compatibilidad.

Personal: Se cuenta con una persona que es capaz de acoplarse a las nuevas tecnologías, por lo que se puede realizar las animaciones sin retrasos.

Control de Riesgos de Negocios

Presupuesto: La planeación económica se adapta a los cambios de precios para el desarrollo del proyecto.

4.1.1.9 Requerimientos Documentación

En este punto haremos uso de la teoría funcionalista, en donde nos responderemos las preguntas que nos plantea.

¿Para quién?

El desarrollo de la investigación se realiza para los diseñadores de la empresa así como los usuarios de las mismas, de estos actores se verán beneficiados terceros como por ejemplo el gerente.

¿Qué queremos?

Tratar de demostrar las posibilidades que tienen los actores para desarrollar animaciones 3D, con el uso de tecnologías gratuitas y orientadas directamente a la Web.

¿Cómo lo queremos?

Desarrollando un análisis comparativo entre las herramientas disponibles en el mercado Adobe Flash y HTML5² con sus Apis¹⁰, generando datos de funcionamiento, características, rendimiento, lenguajes de programación.

¿Dónde?

Se aplicara la nueva tecnología en los departamentos de diseño y sistemas para una mejor presentación de la imagen de la empresa, haciendo uso de las 2 tecnologías de las cuales se pretende obtener información, para encontrar la más idónea para la empresa.

Para realizar la comparación de las tecnologías se realizara un animación 3D en flash con Away 3D¹³ y en HTML5 con sus Apis, la animación se trata de generar cubos y presionando una tecla se irán creando nuevos cubos con el fin de ir recabando información del rendimiento de cada de una de las tecnologías.

Los tiempos de desarrollo con la nueva herramienta pueden llegar a duplicarse debido a la falta de experiencia y desconocimiento del lenguaje por parte de los usuarios por lo que se pueden utilizar los dos tipos de herramientas a la ves para evitar pérdidas y falta de resultados para los clientes.

Con esto se espera que los usuarios se interesen por la nueva tecnología y la terminen implementado en sus empresas, debido a que se obtendrán resultados positivos al ser sistemas gratuitos.

4.1.1.10 Análisis Documental

Herramienta actual Flash - Stage 3D

Stage3D realiza la representación 3D en tiempo real. Con esta herramienta, se puede sacar el máximo provecho de las capacidades de hardware o la aceleración GPU²⁰ del ordenador, directamente desde Flash.

La liberación de esta herramienta fue un gran evento para los desarrolladores de Flash. Debido a que es capaz de utilizar la aceleración 3D haciendo posible la creación de animaciones para los juegos y aplicaciones que no eran posibles anteriormente.

La calidad y la complejidad de los juegos han aumentado exponencialmente. La aceleración de hardware ha mejorado y la capacidad de las aplicaciones, permitiendo la prestación mejorada y veloz, la cual es necesaria para modelos complejos, efectos realistas.

Con el 98% de uso en el mercado, Flash Player⁴ es la herramienta por excelencia. Los juegos y aplicaciones desarrolladas estuvieron disponibles inmediatamente para casi cualquier ordenador en el mundo. Los usuarios pueden introducir la URL²¹ de la animación en un navegador y empezar a usarla. Para la utilización de dichas animaciones se necesita la instalación

²⁰ GPU es un coprocesador dedicado al procesamiento de gráficos u operaciones de coma flotante, para aligerar la carga de trabajo del procesador central en aplicaciones como los videojuegos y o aplicaciones 3D interactivas.

²¹ URL es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, vídeos, presentaciones digitales, etc.

del plugin necesario. El usuario no necesita instalar nada complementariamente: No se necesitan bibliotecas especializadas de ejecución, versiones de sistema operativo o hardware especial.

Herramienta estudiada HTML5 con WebGL

WebGL es una DOM²² API¹⁰ para crear gráficos en el navegador. Al ser una API ofrece una serie de llamadas a una librería especializada en renderizado 3D. Esta librería está basada en OpenGL¹¹ ES 2.0. WebGL trabaja bajo el elemento Canvas de HTML5² por lo tanto es accesible desde cualquier lenguaje compatible con DOM, Javascript básicamente.

En una rápida conclusión, para conocer al lenguaje de acceso a WebGL⁸ habría que referirse a Javascript y para explorar la API de WebGL habrá que referirse a OpenGL ES 2.0

El resto de navegadores que implementan WebGL no adoptan esta misma estructura sobretodo Firefox que no usa WebKit como sistema de renderizado. Aun habiendo estas diferencias es importante entender como un navegador tan importante como Google Chrome trabaja internamente para ofrecer WebGL. En aplicaciones intensas bajo WebGL, como el caso de este proyecto, el navegador va a tener un papel muy importante en el rendimiento de la aplicación sobre todo en momentos de mucha carga gráfica y de procesado. Conocer el funcionamiento interno nos ayudara adoptar medidas para adaptar nuestra aplicación al sistema local.

²² DOM El Modelo de Objetos del Documento (DOM) es una interfaz de programación de aplicaciones (API) para documentos HTML.

4.1.2 Elaboración

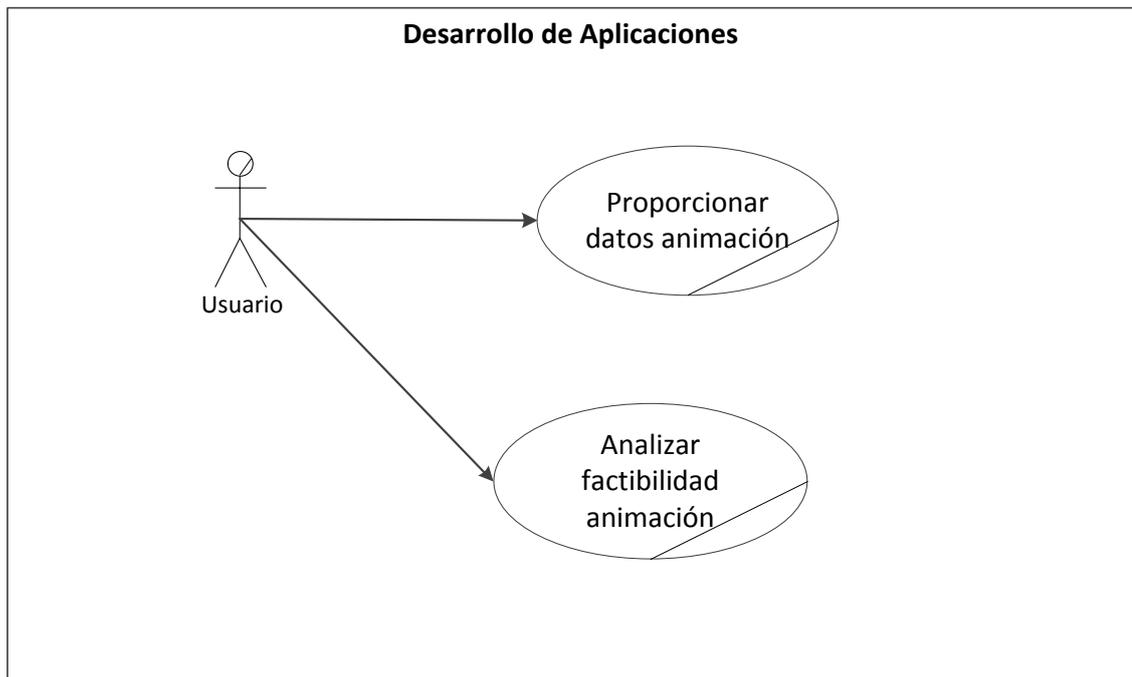


Figura No. 11 Caso de uso de la tecnología propuesta Usuario.

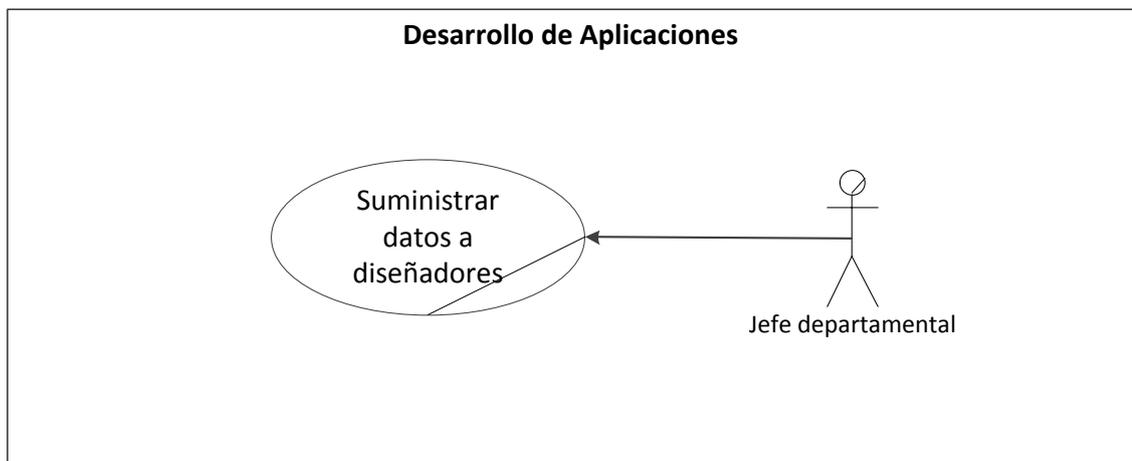


Figura No. 12 Caso de uso de la tecnología propuesta Jefe Departamental.

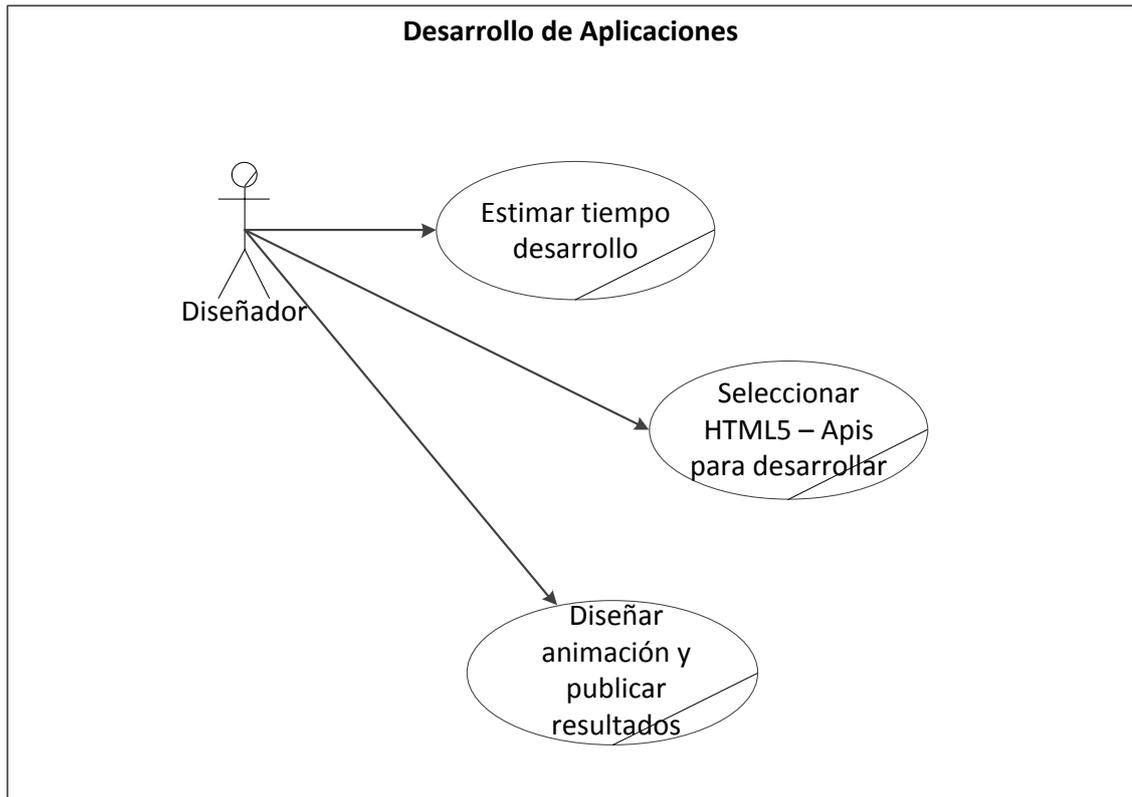


Figura No. 13 Caso de uso de la tecnología propuesta Diseñador.

4.1.3 Diagrama de Actividades Propuesta

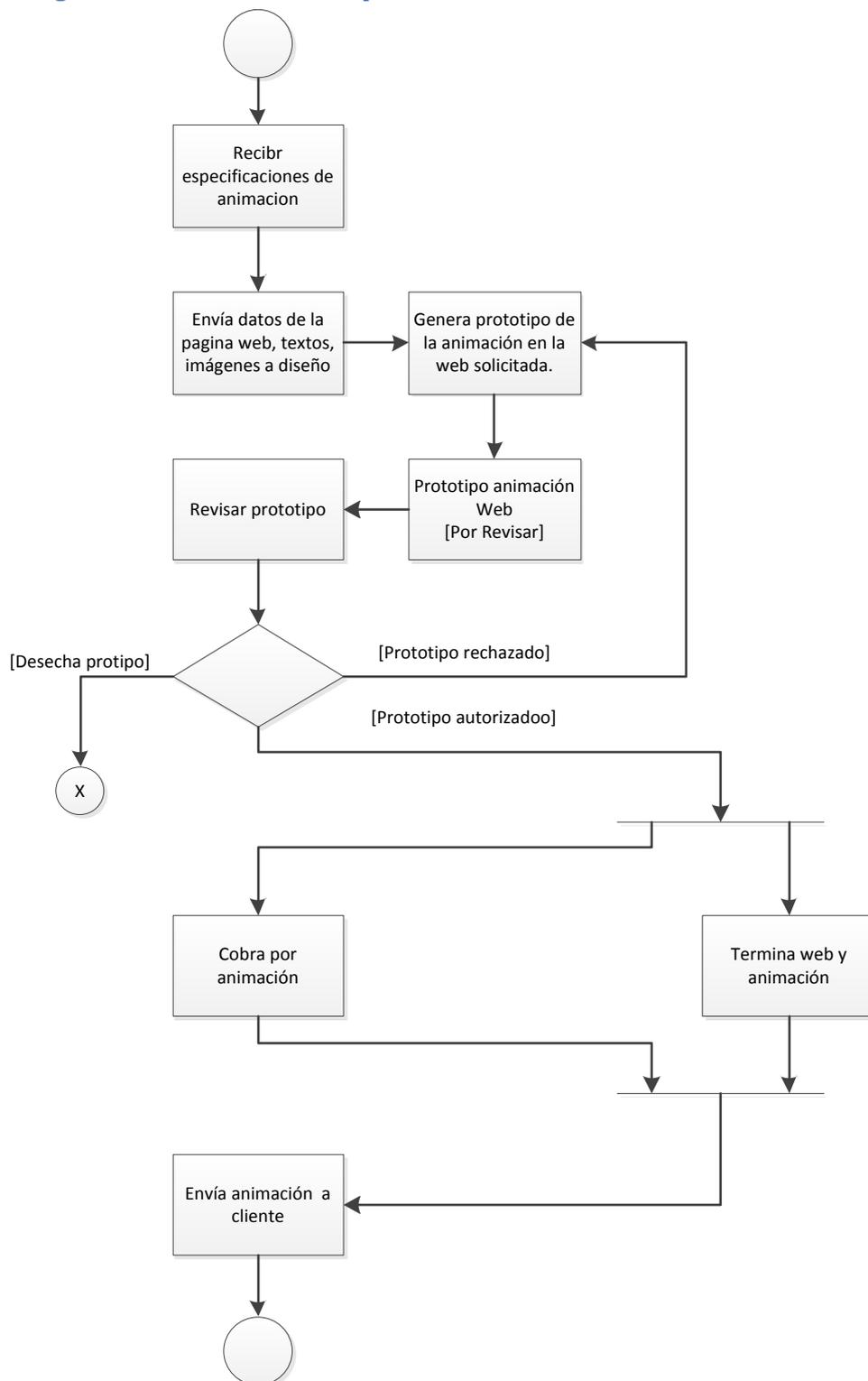


Figura No. 14 Diagrama de procesos de la tecnología propuesta

Funcionamiento WebGL Arquitectura

En este modelo se utiliza JavaScript⁵ para obtener a través del DOM²² (Dynamic Object Model) el elemento Canvas⁷ de HTML5². Una vez obtenido, se define el contexto WebGL⁸, por medio del cual accedemos a la API de WebGL. Por eso se dice que, técnicamente es un enlace (binding) de JavaScript para usar la implementación nativa de OpenGL ES¹¹ 2.0. Este último, se encarga de comunicarse con el driver de la tarjeta gráfica y así poder realizar la aceleración por hardware en la GPU. Como vemos, se diferencian los eventos que ocurren en el espacio de usuario y en el kernel.

La API¹⁰ de WebGL interactúa bien con el resto de plataformas web; específicamente se proporciona apoyo para la carga de texturas 3D a partir de imágenes HTML o video, y la entrada del teclado y mouse se manejan mediante los conocidos eventos DOM²².

Esta API es de muy bajo nivel por lo que se requieren de buenos conocimientos de programación y matemáticas 3D.

Una característica importante es que, brinda la posibilidad de visualizar contenidos 3D en la web sin la necesidad de utilizar plugins, ya que se encuentra implementado en el navegador.

Otra característica relevante es que es un estándar web, Un estándar web, es un conjunto de especificaciones técnicas en constante evolución y de mejores prácticas para construir sitios web. Con ello se facilita el mantenimiento, la usabilidad, la interoperabilidad y la calidad de los trabajos.

Utilizar un estándar asegura una larga vida a los proyectos, ya que provee por lo menos una pequeña estructura de mantenimiento. También asegura que la mayoría de las personas puedan visitar el sitio web, sin importar que navegador se esté utilizando. La compatibilidad hacia adelante y hacia atrás es posible.

En el caso particular de WebGL⁸, su API¹⁰ está basado en un estándar de gráficos 3D familiar y ampliamente aceptado.

No utilizar un estándar tiene varias desventajas, como poco soporte, mayor exposición a problemas de seguridad y problemas de rendimiento. Otro problema que se puede dar, es una dependencia absoluta del proveedor, el cual puede tener un costo elevado.

Pipeline de renderizado de WebGL

El pipeline que usa WebGL es exactamente el mismo del usado por parte de OpenGL ES¹¹ 2.0. Este implementa un pipeline gráfico programable mediante Shading es decir niveles de oscuridad mediante la aplicación de medios de comunicación de áreas oscuras a claras.

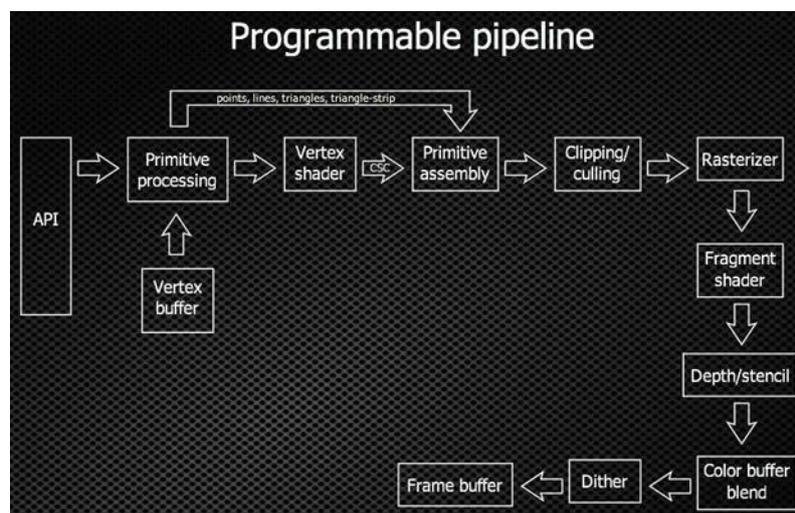


Figura No. 15 Pipeline programable - ES2.0
<http://bit.ly/wJU8Dy>

Vertex Shaders

Los vertex shaders son programas que se llaman una vez por cada vértice en la escena. Haciendo una escena simple en un solo cuadro, con un vértice en cada esquina este sería llamado cuatro veces. Su trabajo consiste en realizar algunos cálculos, como la iluminación, las transformaciones geométricas, etc. Averigua la posición del vértice, y también transmite algunos datos para el fragment shader.

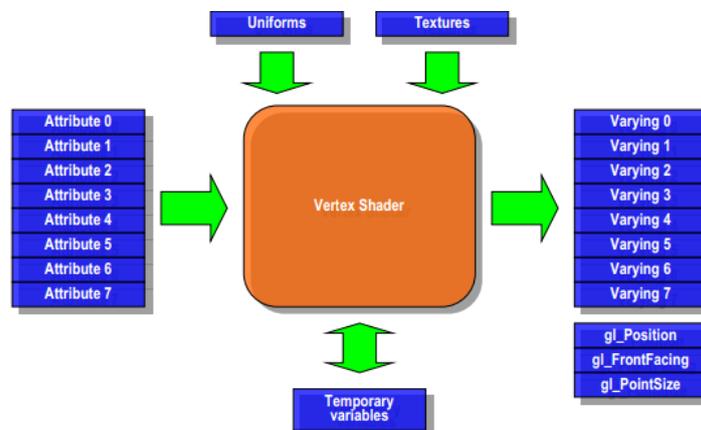


Figura No. 16 Vertex Shader – OpenGL ES2.0
<http://bit.ly/T5jD4n>

- Attributes - Datos por vértice en los Vertex Arrays.
- Uniforms - Datos constantes en el Vertex Shader.
- Samplers - Tipo específico que representa que textura usar en el caso que se use.
- Shader Program - Código fuente o ejecutable que describe las operaciones que serán ejecutadas en cada vértice.
- Varyings - Datos de salida que pueden ser usados en etapas posteriores del pipeline.

Primitive Assembly

Después del procesamiento del Vertex Shader, la siguiente etapa del pipeline es la primitive assembly denominada como una primitiva la cual es un objeto geométrico que puede ser representado gráficamente por OpenGL ES¹¹. Así que después del Vertex Shader los vértices son ensamblados en formas geométricas individuales tales como triángulos, líneas o puntos – Sprite con sus respectivas posiciones. En esta etapa también se guarda el estado de si la forma geométrica está dentro o no del frustum de visión de la pantalla. Si la primitiva está enteramente fuera del frustum este es descartado y no se trata en etapas posteriores. Si la primitiva está parcialmente fuera del frustum tendrá que ser recordada para ajustarse automáticamente al frustum. Este proceso es conocido como clipping. Si se activa la opción de culling se ejecutara un proceso de caras que descartara aquellas que no estén orientadas con lo normal hacia el punto de visión. Después de ejecutadas las opciones culling y clipping la primitiva esta lista para pasar a la siguiente etapa del pipeline.

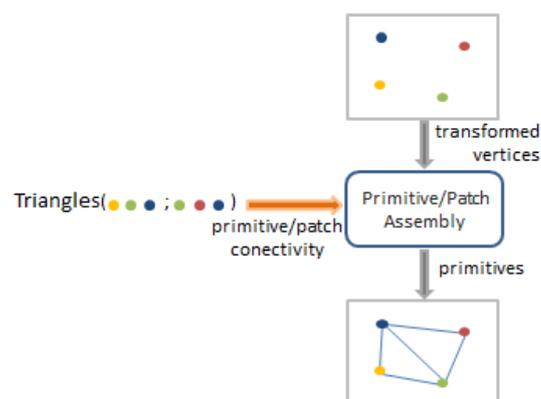


Figura No. 17 Primitive Assemblies – OpenGL ES.
<http://bit.ly/QRsEBW>

Rasterización

Las primitivas llegan a esta etapa y se van renderizando en el orden en que fueron recibidas, el resultado de la rasterización de una primitiva es una secuencia de fragmentos.

Los fragmentos son conjuntos de estados que se utilizan para calcular datos finales para píxeles en el frame buffer de salida. El estado de un fragmento incluye su posición en el espacio de la pantalla. Estos fragmentos serán tratados posteriormente por el Fragment Shaders.

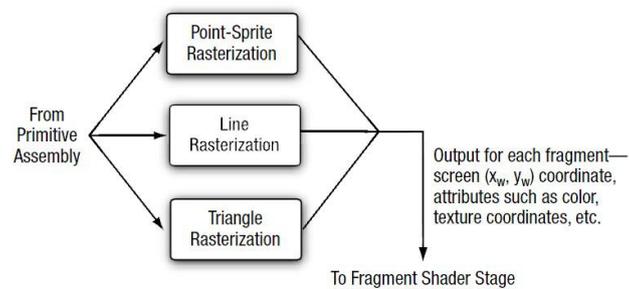


Figura No. 18 Rasterización OpenGL ES.
<http://bit.ly/QRsEBW>

Fragmentes Shaders

Son programas que se llaman una vez por cada píxel creado en la renderización de la escena. Los fragment shaders pueden realizar cálculos de iluminación, pero su labor más importante es establecer el color final del píxel.

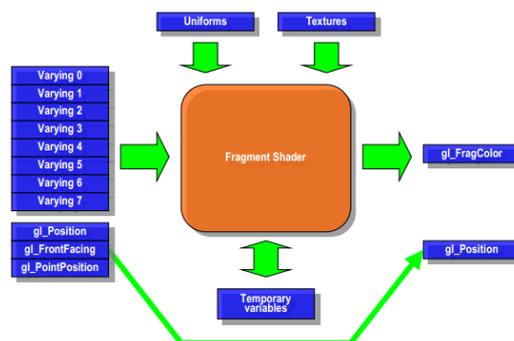


Figura No. 19 Fragments Shaders – OpenGL ES
<http://bit.ly/T5jD4n>

GPU Process

El proceso de ejecución de animaciones conocido como GPU process (unidad de procesamiento gráfico), se ejecuta a través de un sandbox para que el proceso de renderizado no haga uso directo de las llamadas 3D que son realizadas por el sistema operativo, esto con el fin de evitar la infiltración de terceras personas al sistema y evitar posibles daños o robos de información. Este proceso se realiza mediante cliente - servidor con la aplicación.

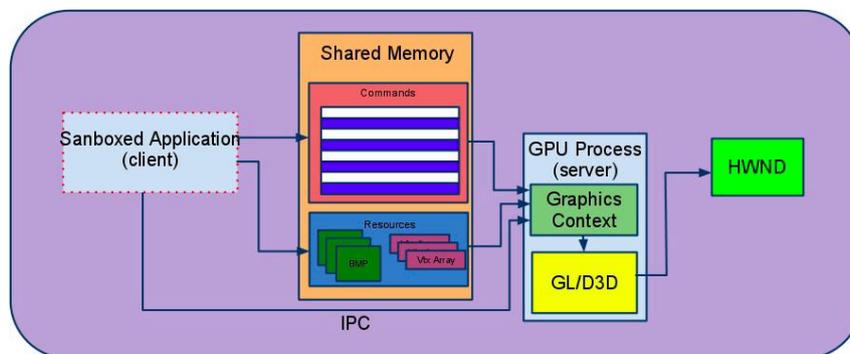


Figura No. 20 Shared Memory Renderizado OpenGL ES 2.0
<http://bit.ly/T5jD4n>

- Cliente - El Renderer Process es el cliente y en vez de ejecutar directamente las instrucciones en la unidad de proceso gráfico, las serializa y las deposita en un Buffer de comandos en un trozo de memoria compartido entre el y el proceso servidor.
- Servidor - El GPU Process trabaja aislado y recoge los datos serializados, los pasa y los ejecuta apropiadamente en la unidad de proceso gráfico.

Como vemos en el gráfico, la Shared Memory es el RPC Buffer donde residen todos los recursos de nuestra aplicación en cada llamada. Como casi todas las llamadas de OpenGL no tienen valores de retorno el cliente y el servidor pueden trabajar asincrónicamente para mantener un buen

rendimiento y no tener que validar ningún retorno. En algunos modos de debug o llamadas muy específicas es necesario controlar retornos para recoger ciertos valores del servidor, para estos casos se usa un mecanismo IPC. Los beneficios de que el GPU Process se ejecute aislado son:

- Seguridad - La lógica de renderizado trabaja aislada no compartiendo recursos básicos.
- Robustez - Si el GPU Process se interrumpe o deja de funcionar no interfiere en el resto de procesos y no para la ejecución del navegador.
- Uniformidad - Estandarizando OpenGL ES¹¹ 2.0 como la API de renderizado permite mantener un único código para las diferentes distribuciones de Chromium.

Por último vemos como el contexto gráfico accede directamente a las librerías nativas del Sistema Operativo GL, Direct3D, o Direct3D mediante angle.

[Soporte para WebGL Octubre 2012](#)

En la siguiente ilustración se visualizan las versiones de los diferentes navegadores y su soporte para WebGL⁸.

WebGL - 3D Canvas graphics - other

Method of generating dynamic 3D graphics using JavaScript, accelerated through hardware

Resources: [Firefox blog post](#) [Polyfill for IE](#) [Instructions on enabling WebGL](#) [Webkit blog post](#) [Tutorial](#)

Global user stats*:

Support:	28.76%
Partial support:	24.78%
Total:	53.54%

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	BlackBerry Browser	Opera Mobile	Chrome for Android	Firefox for Android
19 versions back			4.0									
18 versions back			5.0									
17 versions back			6.0									
16 versions back		2.0	7.0									
15 versions back		3.0	8.0									
14 versions back		3.5	9.0									
13 versions back		3.6	10.0									
12 versions back		4.0	11.0									
11 versions back		5.0	12.0									
10 versions back		6.0	13.0		9.0							
9 versions back		7.0	14.0		9.5-9.6							
8 versions back		8.0	15.0		10.0-10.1							
7 versions back		9.0	16.0		10.5							
6 versions back		10.0	17.0		10.6							
5 versions back		11.0	18.0	3.1	11.0			2.1				
4 versions back	5.5	12.0	19.0	3.2	11.1	3.2		2.2		10.0		
3 versions back	6.0	13.0	20.0	4.0	11.5	4.0-4.1		2.3		11.0		
2 versions back	7.0	14.0	21.0	5.0	11.6	4.2-4.3		3.0		11.1		
Previous version	8.0	15.0	22.0	5.1	12.0	5.0-5.1		4.0		11.5		
Current	9.0	16.0	23.0	6.0	12.1	6.0	5.0-7.0	4.1	7.0	12.0	18.0	15.0
Near future	10.0	17.0	24.0		12.5				10.0	12.1		
Farther future		18.0	25.0									

Figura No. 21 Soporte a WebGL
<http://caniuse.com/webgl>

Como podemos observar en la imagen el navegador que más soporte da a WebGL⁸ es Chrome que viene dando soporte desde su versión 8.0 pasando de parcialmente soportado ha soportado en su versión 23 que es la actualmente lanzada. Seguido de Firefox que da un soporte parcial desde su versión 4 hasta la 18 que es la actual.

En cuanto a los navegadores de móviles los que más soporte dan son BlackBerry Browser 10 y Firefox para android en su versión 15.

Como podemos observar el navegador más popular de los usuarios IE no da soporte en ninguna de sus versiones debido a que Microsoft no apoya la tecnología de visualización 3D, temiendo por problemas de seguridad que puedan ocurrir con su sistema operativo por lo que no se buscan problemas y simplemente no adhieren el soporte para WebGL a su navegador.

Uso de Sistemas Operativos

En los siguientes gráficos observamos los porcentajes de uso de los diferentes sistemas operativos hasta octubre del 2012.

Windows sigue siendo el sistema operativo más utilizado, seguido por Mac OS X y Linux indudablemente WebGL es el más soportado por los 2 primeros sistemas operativos y en el que los navegadores tienen más compatibilidad y soporte.

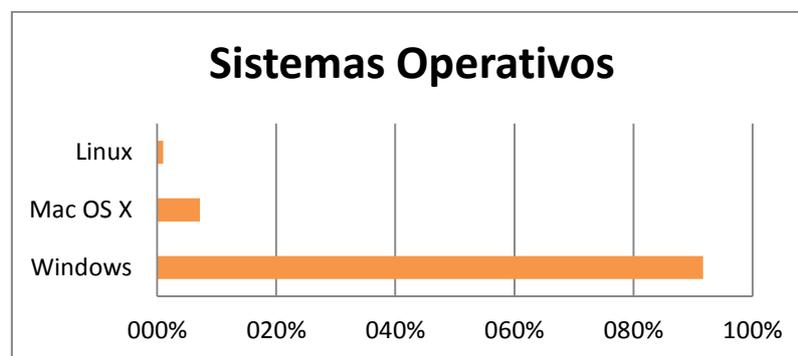


Tabla 6 Porcentajes de uso Sistemas Operativos

En el siguiente gráfico observamos las versiones de los sistemas operativos más utilizados en este caso los últimos sistemas operativos lanzados por parte de Microsoft son los que dominan el mercado, seguidos por los sistemas de Mac que actualmente siguen subiendo en el Rankin de preferencia por parte de los usuarios.

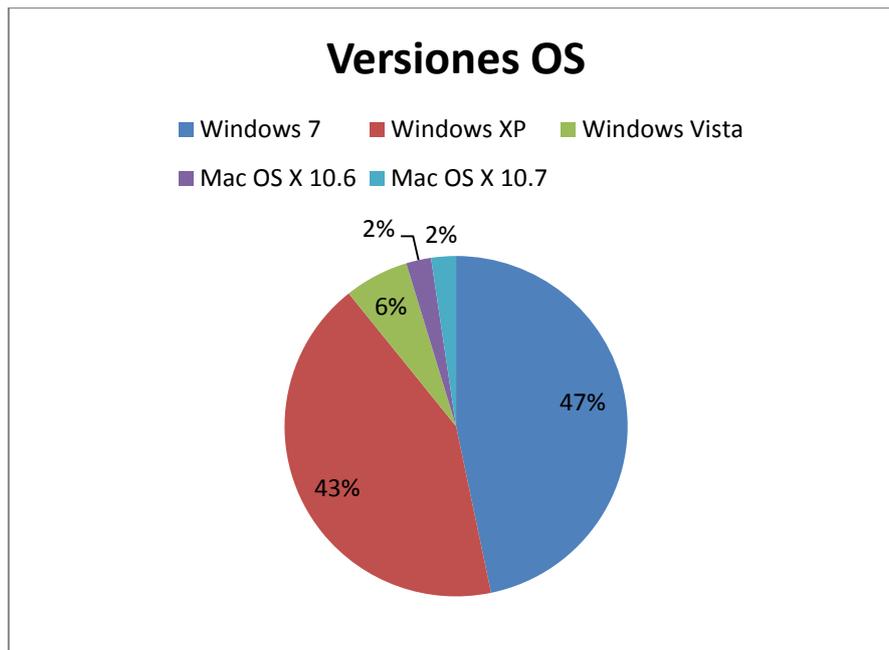


Figura No. 22 Porcentajes de uso versiones de Sistemas Operativos

Uso Navegadores

En cuanto a los navegadores tenemos como indiscutible ganador a IE como el más preferido por los usuarios pero por el momento el más odiado por los desarrolladores de animaciones 3D que prefieren WebGL⁸, debido a que IE no da soporte para este sistema de animaciones volviendo inservible al elemento Canvas dentro del e imposibilitando la ejecución de WebGL, por lo que los desarrolladores se ven obligado al uso de otras tecnologías y plugins para la ejecución de este tipo de animaciones.

Los navegadores que lo siguen son Firefox y Chrome, navegadores que dan soporte a WebGL tenemos un rápido aumento en los últimos tiempos de preferencia por el navegador de Google el cual está dando alcance a Firefox y se espera que en poco tiempo se capas de remplazar a IE como el navegador más utilizado, esto da una gran ventaja a WebGL ya que es completamente soportado por Chrome haciendo predecir que va a ser una

gran competencia para flash player⁴ en el mercado de las animaciones dentro de los navegadores.

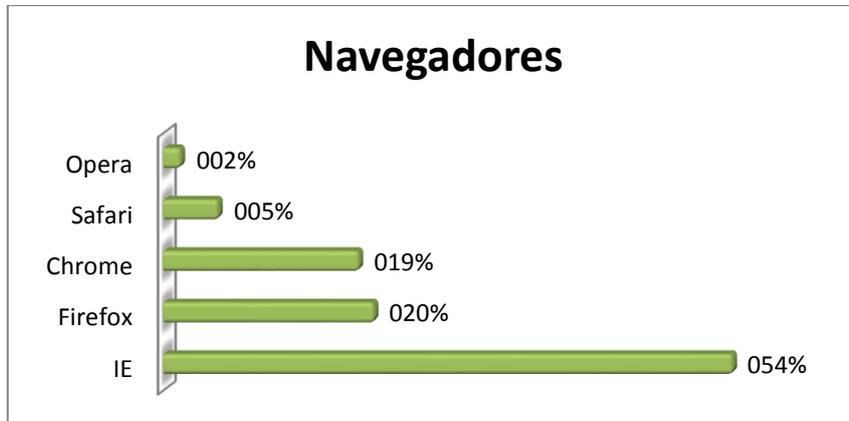


Figura No. 23 Porcentaje uso de navegadores

En cuanto a versiones de navegadores la versión más utilizada de IE es la 8.0 seguida por la versión 22.0 de Chrome y Firefox.

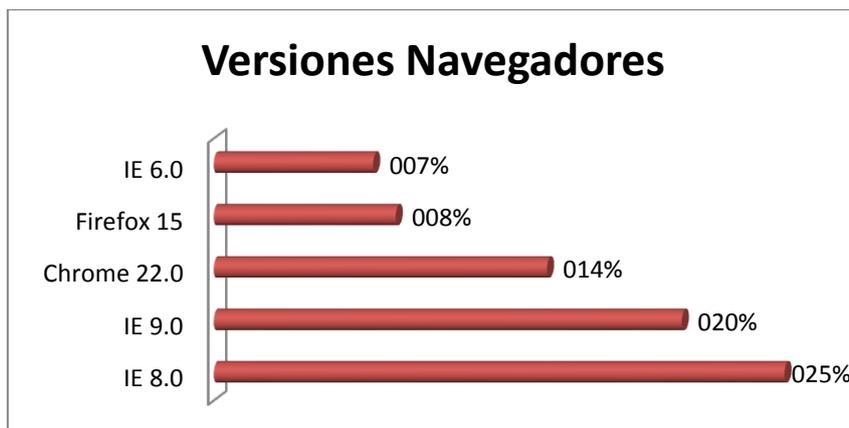


Figura No. 24 Porcentaje uso versiones de navegadores.

Viendo estos datos parece que el futuro de WebGL⁸ está asegurado. Si tenemos en cuenta que el Sistema Operativo que mejor lo soporta es el que más se usa (Windows) y lo mismo para el navegador (Chrome) que su uso está creciendo cada día más y es cuestión de tiempo para que se proclame el más usado, se puede decir con cierta seguridad que el soporte de WebGL solo va a seguir creciendo.

Hay que remarcar que Firefox tiene menos soporte que Chrome en un 47% vs el 95% de Chrome ya que Chrome usa una alternativa para los ordenadores con graficas no permitidas, SwiftShader. Safari soporta WebGL pero viene desactivado por defecto por lo tanto su porcentaje es casi nulo. Con el tiempo, todos los navegadores que tengan WebGL tendrán el 100% de soporte, ya que las nuevas tarjetas gráficas vendrán con drivers más seguros y ajustados a las necesidades actuales. Así que el futuro de esta tecnología está asegurado en cuanto a soporte. Sin embargo hace falta analizar que hay otras tecnologías existentes en el mundo Web. El único punto negro es IE por la falta de soporte.

WebGL y la Realidad Aumentada

Al igual que Flash Player⁴ WebGL⁸ tiene la posibilidad de generar animaciones con realidad aumentada para ello hace uso de JavaScript. Utiliza la librería JSARToolKit con el API getUserMedia de WebRTC que nos permite generar aplicaciones de realidad aumentada en la Web mediante la integración de estas APIS con WebGL. JSARToolKit trabaja con el elemento Canvas.

Debido a que necesita leer la imagen de fuera del elemento canvas, es necesario que esta sea leída desde el mismo origen que la pagina es decir CrossOrigin para evitar el robo o el uso inadecuado de imágenes cross-Domain en este caso de la textura.

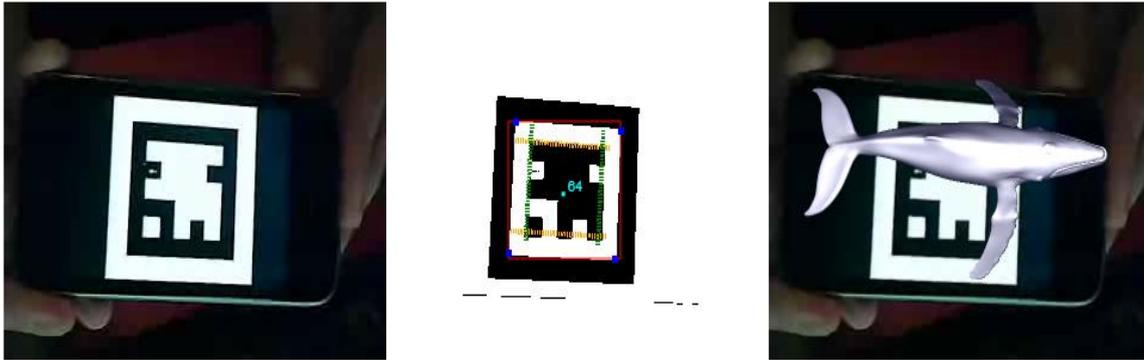


Figura No. 25 Ejemplo Realidad Aumentada WebGL
bit.ly/TMIP09

Seguridad

Como se mencionó anteriormente WebGL⁸ permite la interacción directa entre el navegador y el hardware de la maquina por lo que se lo interpreta como un sistema muy permisivo, dando la posibilidad a los hackers de ejecutar líneas código infinito bloqueando las tarjetas gráficas.

Se han detectado vulnerabilidades con respecto a los controladores debido a que estos controladores no estaban pensados en interactuar directamente con la web por lo que no se tenía en mente una seguridad web, problema que tendrá Microsoft si algún día decide permitir el soporte para WebGL o uno de sus productos en el caso de ingresar al mundo 3D.

WebGL no es un plugin, como Flash o ActiveX, sino un built-in del navegador y no proporciona el acceso nativo extremo que por ejemplo tiene ActiveX, no se puede escribir en disco, no se puede acceder a la memoria principal y no se puede ejecutar código CPU fuera del SandBox de Javascript. Sus problemas de seguridad están relacionados solo con el Hardware gráfico.

Robo de Imágenes Cross-Domain.

Uno de los puntos fundamentales de la seguridad en la especificación DOM²² son las fronteras de los dominios. Estas fronteras previenen acceder a contenido autenticado y confiado de un dominio a otro. WebGL⁸ permitido el uso de textura Cross-Domain es decir entre diferentes dominios. Es totalmente aceptable poder cargar una imagen externa, fuera de tu dominio, en tu DOM porque no tienes forma de acceder al contenido y descifrar que estas mostrando. Pero WebGL mediante un Shader malicioso puede llegar a interpretar esa imagen, convertida en textura y actuar en consecuencia leyendo los pixeles correspondientes. La respuesta de los navegadores fue muy rápida y contundente: negar texturas Cross-Domain y activar la política del mismo origen. Esta política es una medida de seguridad para Scripts en la parte de cliente y previene que un documento o Script cargado en un origen pueda cargarse o modificar propiedades de un documento desde un origen diferente.

Denegación de servicio.

Un ataque de denegación de servicio, también llamado ataque DoS, es un ataque a un sistema de computadoras o red que causa que un servicio o recurso sea inaccesible a los usuarios legítimos. Normalmente provoca la pérdida de la conectividad de la red por el consumo del ancho de banda de la red de la víctima o sobrecarga de los recursos computacionales del sistema.

El procedimiento de este ataque es el siguiente:

- 1.- Un usuario visita un sitio donde reside un script WebGL malicioso.

2.- El componente WebGL⁸ sube cierta geometría y cierto código en forma de Shader a la tarjeta gráfica del usuario.

3.- El código o la geometría se aprovecha de Bugs o Exploits en los drivers de la tarjeta gráfica.

4.- El Hardware grafico puede ser atacado causando que todo el sistema se paralice.

La solución adoptada por los navegadores que sufran esta vulnerabilidad ha sido restringir el acceso a todas aquellas tarjetas gráficas con drivers vulnerables. En ningún caso el Exploit puede ofrecer control de la maquina o robo de información.

4.1.4 Construcción

Rendimiento WebGL y Molehill (Stage3D)

Para realizar las demostraciones del funcionamiento de las 2 tecnologías más sobresalientes en el entorno 3D, se ha hecho uso de los siguientes sistemas Aea3D para Stage3D y Three.js para WebGL.

Away3D 4 Es la última versión del popular motor de 3D Flash, construido para utilizar Stage3D. Esta aplicación tiene una gran comunidad en línea y un buen mantenimiento posee un conjunto de APIs y de documentos. Se encuentra actualmente en alfa y algunas características se pierden, por ejemplo, no hay ningún objeto de sistemas de partículas y los filtros / support shaders es bastante limitado.

Three.js es un código abierto WebGL motor 3D es una biblioteca JavaScript que funciona como un engine esta librería nos permite crear de

manera simple objetos tridimensionales tales como cámaras, iluminación materiales además de permitirnos trabajar con texturas y escenarios los cuales también se pueden animar. Todo esto lo hace aprovechando las ventajas que introduce HTML5² y la introducción de la etiqueta Canvas⁷ y la innovación que provee WebGL para la web 3D.

Plataformas compatibles

Para ejecutar las aplicaciones en tres dimensiones mediante aceleración por hardware con Stage3D se necesita de Flash11 o superior y una tarjeta gráfica que soporte OpenGL ES¹¹ y DirectX, en el caso de que no se pueda ejecutar en modo hardware la animación caerá en ejecución por software la misma que es 5 a 10 veces más lento. Esta detección se la puede realizar de manera manual en el software de renderización, y manejarla de manera más apropiada.

La ventaja de Stage3D es que se puede ejecutar en internet Explorer con la ayuda de Flash Player.

WebGL⁸ no requiere plugins y posee soporte actualmente en Chrome, Firefox, Safari y Opera, siendo IE el único que no la posee. WebGL en Chrome corriendo bajo Windows utiliza un ANGLE que convierte WebGL a DirectX y le da un buen rendimiento.

En el Mac, tanto Stage3D WebGL se traducen en OpenGL, así que no hay mucha diferencia en el rendimiento. Ninguna de las opciones descritas anteriormente se ejecutan en los dispositivos móviles (iOS / Android), sin embargo se espera que ambos en el futuro posean soporte para estos sistemas operativos móviles.

Comparación del rendimiento

Se ha probado las dos demostraciones en un MacBook Pro (con una NVIDIA GeForce GT 330M) y una gama media Windows 7 PC de escritorio (con chipset Nvidia GeForce 7050 / nForce 620i integrado). Su rendimiento puede variar.

Navegador	Stage3D	WebGL
Mac/Chrome	60 FPS(OpenGL)	60 FPS
Mac/Firefox	60 FPS (OpenGL)	60 FPS
Mac/Safari	60 FPS (OpenGL)	60FPS
Windows/Chrome	10 FPS	30 FPS
Windows/Firefox	10 FPS	21 FPS
Windows/IE	14 FPS	No soportado

Tabla 7 Tabla de compatibilidad Stage3D vs WebGL

En la tabla podemos observar que tanto Stage3D como WebGL en Mac funcionan bien teniendo en cuenta que se posee el hardware necesario de lo contrario funciona mal.

Stage3D en Windows funciona bien teniendo los recursos de hardware necesarios en este caso se contaba con chipset Nvidia integrados por lo que el rendimiento sobre es 60 es de 10 FPS²³ en Chrome y Firefox, pero se observa un aumento de calidad al ejecutarse bajo IE en 4 puntos. Un nivel de animación muy bajo.

WebGL⁸ en Windows tiene un nivel más alto de animación que Stage3D, Chrome es el que mejor soporte lo da con una velocidad de 30 FPS

²³ FPS En informática estos fotogramas están constituidos por un número determinado de píxeles que se distribuyen a lo largo de una red de texturas. La frecuencia de los fotogramas es proporcional al número de píxeles que se deben generar, incidiendo en el rendimiento del ordenador que los reproduce.

además de observar una integración similar de WebGL⁸ con Firefox que posee una velocidad de 21 FPS²³, esto hace ver como los navegadores se van acoplando a la nueva tecnología debido que hasta hace poco Firefox no soportaba WebGL si no era con aceleración los recursos necesarios.

En Mac, la demo Stage3D es más lento para inicializar y el frame-rate tartamudea tiene más puntos que su contrincante WebGL. Stage3D da mezcla de color más interesante y reflejos especulares.

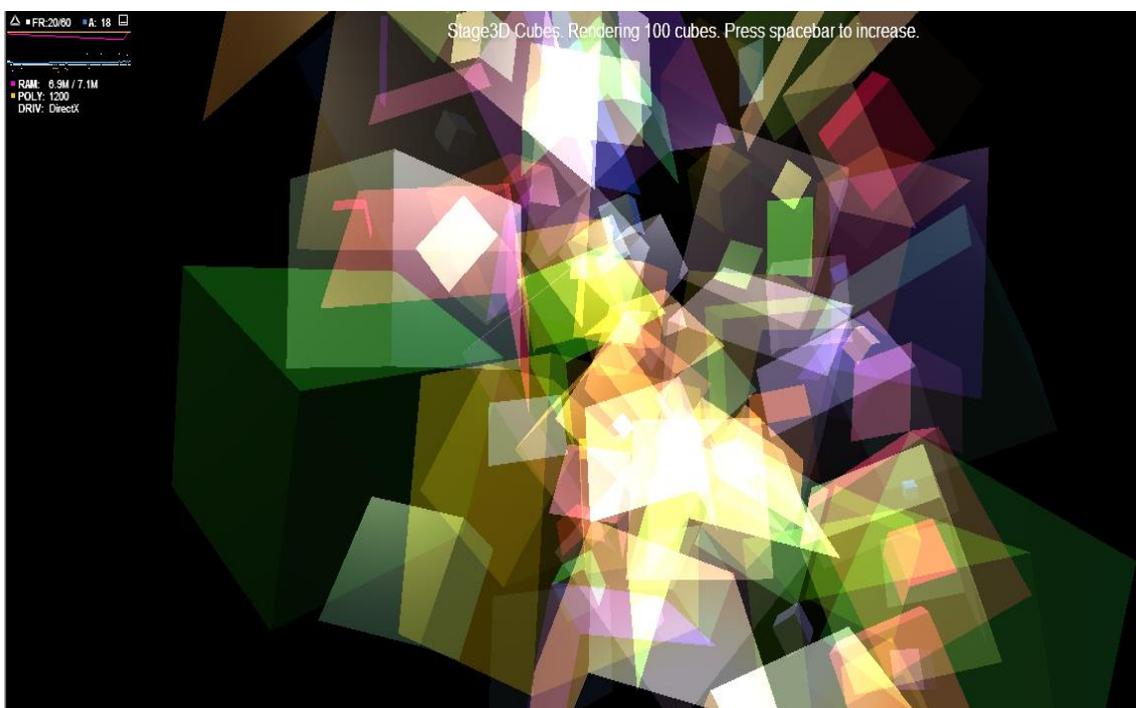


Figura No. 26 Ejemplo Cubos 3D – AWAY 3D

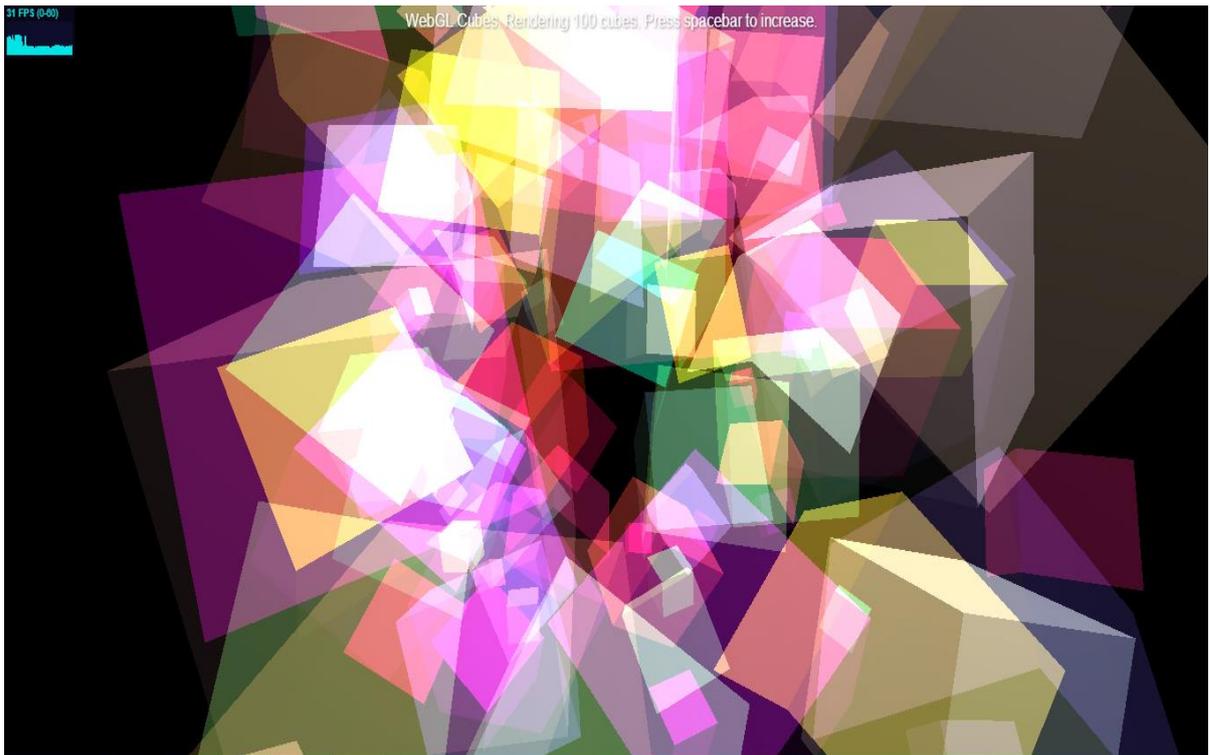


Figura No. 27 Ejemplo cubos WebGL con Three.js

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Investigar sobre la especificación estándar WebGL⁸, Canvas de HTML5² y los conceptos programables que se utilizan en las etapas de proyección esta tecnología.

Después de la lectura de esta herramienta creo que queda demostrado el conocimiento de esta tecnología. No solo cómo funciona el lenguaje y su API, sino como es interpretada por el navegador, como llega a la GPU²⁰ y como es ejecutado internamente, se realizó un ejemplo de una aplicación con esta tecnología, para realizar la comparación entre las tecnologías utilizadas.

Investigar sobre la tecnología usada por flash principal competidor de WebGL en esta rama de la tecnología.

Se realizó el estudio de la API¹⁰ de Flash Player, al igual que WebGL, se estudió todo el funcionamiento que realiza para la ejecución de aplicaciones 3D a través de su API denominada AWAY3D. Se realizó la misma aplicación de la herramienta gratuita para demostrar las diferencias en cuanto a rendimiento en las diferentes maquinas.

Demostrar a las empresas, que existen posibilidades de software gratuitas para determinadas tareas.

Se realizó un estudio de las herramientas gratuitas y pagadas en cuanto a desarrollo 3D, en donde las empresas tendrán la oportunidad de empaparse que funcionalidades cumple y cuales podrían remplazar a las que usan

actualmente para determinadas tareas disminuyendo los gastos anuales en licencias y capacitaciones.

Generar un documento en el cual las empresas puedan estudiar sus posibilidades, ventajas y desventajas de cambiar de tecnología.

Con el desarrollo de este proyecto las empresas tendrán la posibilidad de estudiar las características de las diferentes herramientas y tendrán la capacidad de elegir cuál de ellas podrá satisfacer sus necesidades y cumplir sus objetivos planteados a corto y largo plazo.

WebGL han demostrado sus fortalezas y potenciales en el rendimiento de aplicaciones 3D y prometedor visualización de datos bastante grandes. El resultado de la investigación es satisfactorio debido a que se observan mejoras en cuanto al ahorro de dinero por parte de las empresas al implementar WebGL y dejar de usar licencias de Adobe, en la actualidad el mundo está lleno de desarrolladores que tienen el conocimiento para aprovechar el WebGL. Sin embargo, el soporte que poseen los navegadores en la actualidad no satisfacen las necesidades de la herramienta para conseguir mejor rendimiento haciendo que WebGL sea utilizado por más desarrolladores. También como apoyo WebGL en dispositivos móviles es una gran implementación debido a que se ejecuta en todos los navegadores a comparación de Flash Player que en algunos dispositivos no está permitida su ejecución, los dispositivos móviles se están desarrollando para dar soporte a WebGL⁸ lo que tomará tiempo para que los dispositivos puedan ejecutar todas las características de tan poderosa herramienta que puede ejecutar escenas complejas en tiempo real dando como resultado una visualización agradable y suave.

RECOMENDACIONES

En cuanto a los datos estudiados obtuvimos un buen resultado en cuanto a ahorros en la empresa, pero se tiene que tomar en cuenta que la herramienta gratuita está orientada a entorno web, por lo que esta herramienta es incapaz de remplazar a software estrictamente para entornos de escritorio, por lo que las empresas deberán seguir con los mismo sistemas u optar por las tecnologías recomendadas para entornos de escritorio como Blender herramienta en la cual se han generado infinidad de proyectos y ha sido utilizada para películas 3D, lo que no habla de la capacidad que tiene esta herramienta para generar dichos elementos y las características de la misma para ser tomada en cuenta por parte de empresas extranjeras para el diseño de estas animaciones y aplicaciones.

BIBLIOGRAFÍA

Catherine Leung and Andor Salga. Enabling webgl. In Proceedings of the 19th international conference on World wide web, WWW '10, pages 1369{1370, New York, NY, USA, 2010. ACM.

[WebGL] WebGL - OpenGL ES 2.0 para la web. Sitio web de Khronos.
URL: <http://www.khronos.org/webgl/> - Retrieved Octubre 10, 2012

OpenGL Evolution & JOGL. Documentos JOGAMP.
URL: <http://jogamp.org/jogl/doc/Overview-OpenGL-Evolution-And-JOGL.html> -
Obtenido Octubre 15, 2012

Creating the Context. Khronos WebGL Wiki.
URL: http://www.khronos.org/webgl/wiki/Tutorial#Creating_the_Context -
Obtenido Octubre 18, 2012

Stage3D. Sitio Web de Adobe.
URL: <http://www.adobe.com/devnet/flashplayer/stage3d.html> - Obtenido
Octubre 20, 2012

Seguridad de WebGL. Khronos News, May 9, 2011. URL:
<http://www.khronos.org/news/permalink/>
WebGL-security - Obtenido Octubre 20, 2012

Suite de programas Autodesk, Sitio Web de Autodesk
URL: www.autodesk.es – Obtenido Octubre 15, 2012

API Away3D Flash, Sitio web de Away
URL: <http://away3d.com> – Obtenido Octubre 15, 2012

Herramienta de desarrollo Gratis Google, Sitio web de Google Sketchup
URL: <http://sketchup.google.com> – Obtenido Octubre 16, 2012

API WebGL Three.js, Sitio Web de Three
URL: <http://mrdoob.github.com/three.js/> - Obtenido Octubre 17, 2012

Experimentos WebGL, Sitio Web Experimentos Chrome
URL: <http://www.chromeexperiments.com/webgl/> - Obtenido Octubre 18, 2012

ANEXOS

ANEXO 1: ABREVIACIONES Y ANGLICISMOS

HTML	HyperText Markup Language
WebGL	Web Graphics Library
OpenGL ES	Open Graphics Library Embedded Systems
GPU	Graphics Processing Unit
CPU	Central Processing Unit
ANGLE	Almost Native Graphics Layer Engine
IPC	Inter Process Communication
API	Application Programming Interface
JS	JavaScript
FPS	Frames Per Second

*Nota: Se ha abusado de muchos anglicismos en palabras técnicas provenientes del inglés.

Si los tradujéramos perderán rigor o precisión ya que su origen no es el castellano, lengua de este proyecto.

ANEXO 2: COMPARACIÓN DE CÓDIGOS

Código Stage3D para Flash

```

package
{
import away3d.containers.ObjectContainer3D;
import away3d.containers.View3D;
import away3d.core.base.Geometry;
import away3d.debug.AwayStats;
import away3d.entities.Mesh;
import away3d.entities.Sprite3D;
import away3d.filters.BloomFilter3D;
import away3d.filters.BlurFilter3D;
import away3d.filters.DepthOfFieldFilter3D;
import away3d.lights.DirectionallLight;
import away3d.lights.LightBase;
import away3d.lights.PointLight;
import away3d.materials.BitmapMaterial;
import away3d.materials.ColorMaterial;
import away3d.materials.methods.FogMethod;
import away3d.primitives.Cube;
import away3d.primitives.Plane;
import away3d.primitives.Sphere;
import flash.display.BitmapData;
import flash.display.BlendMode;
import flash.display.Sprite;
import flash.display.StageAlign;
import flash.display.StageScaleMode;
import flash.events.Event;
import flash.events.KeyboardEvent;
import flash.events.MouseEvent;
import flash.filters.DropShadowFilter;
import flash.geom.ColorTransform;
import flash.geom.Vector3D;
import flash.text.TextField;
import flash.text.TextFieldAutoSize;
import flash.text.TextFormat;
import flash.text.TextFormatAlign;
import flash.utils.getTimer;
/**
 * Cubes Stage3D Demostracion
 */
[SWF(backgroundcolor='0x333333', framerate='60', width='800',
height='600')]
public class Main extends Sprite
{
private var MATERIAL_COUNT:int = 30;
private var CUBE_SIZE:int = 300;
private var CUBE_INCREMENT:int = 100;
private var SPREAD:int = 1000;

private var view : View3D;
private var cubes:Array = [];

```

```

private var materials:Array = [];
private var cubeHolder : ObjectContainer3D;
private var rotSpeed:Number = .3;
private var cubeCount:int = 0;
private var windowHalfX:int;
private var windowHalfY:int;
private var mX:int = 0;
private var mY:int = 0;
private var infoText:TextField;
private var geometry:Geometry;

public function Main(){
super();

stage.scaleMode = StageScaleMode.NO_SCALE;
stage.align = StageAlign.TOP_LEFT;

//init 3D world
view = new View3D();
view.camera.z = -1000;
addChild(view);
view.addEventListener(Event.ADDED_TO_STAGE,onViewAdded);

//init object to hold cubes and rotate
cubeHolder = new ObjectContainer3D();
view.scene.addChild(cubeHolder);

//add info text
infoText = new TextField();
infoText.autoSize = TextFieldAutoSize.LEFT;
infoText.y = 5;
infoText.filters = [new
DropShadowFilter(2.5,90,0x000000,.3,3,3,.5,1)];
var format:TextFormat = new TextFormat();
format.font = "Helvetica,Arial,_sans";
format.size = 16;
format.color = 0xFFFFFFFF;
infoText.defaultTextFormat = format;
addChild(infoText);

//add lights
var light:PointLight = new PointLight();
light.position = new Vector3D(-1000,1000,-1000);
light.color = 0xffeeaa;
view.scene.addChild(light);

var light2:PointLight = new PointLight();
light2.position = new Vector3D(1000,1000,1000);
light2.color = 0xFFFFFFFF;
view.scene.addChild(light2);

//init materials
for (var i:int = 0 ; i < MATERIAL_COUNT; i ++ ){

```

```

var material : ColorMaterial = new
ColorMaterial(Math.random()*0xFFFFFF,.5);
material.blendMode = BlendMode.ADD;
material.lights = [light,light2];
materials.push(material);
}

//init cubes
Geometry =
newCube(materials[0],CUBE_SIZE,CUBE_SIZE,CUBE_SIZE).geometry;
addCubes();

//add stats
addChild(new AwayStats(view));

this.addEventListener(Event.ENTER_FRAME, onEnterFrame);
stage.addEventListener(KeyboardEvent.KEY_DOWN,onKeyDown);
stage.addEventListener(MouseEvent.MOUSE_MOVE,onMouseMove);
stage.addEventListener(Event.RESIZE, onStageResize);
onStageResize(null);

}

private function addCubes():void {
cubeCount += CUBE_INCREMENT;

for (var j:int = 0 ; j < CUBE_INCREMENT; j ++ ){

var cube:Mesh = new Mesh(materials[j % MATERIAL_COUNT], geometry);
//randomize size
cube.scaleX = cube.scaleY = cube.scaleZ = Math.random() + .1;

cubeHolder.addChild(cube);
cubes.push(cube);

cube.x = Math.random()* SPREAD - SPREAD/2;
cube.y = Math.random()* SPREAD - SPREAD/2;
cube.z = Math.random()* SPREAD - SPREAD/2;

cube.rotationX = Math.random()*360 - 180;
cube.rotationY = Math.random()*360 - 180;
cube.rotationZ = Math.random()*360 - 180;

}

//make more cubes less opaque
for(var i:int = 0; i < MATERIAL_COUNT; i++) {
materials[i].alpha = 50 / cubeCount;
}

//Texto con la informacion para crear mas cubos.
infoText.text = "Stage3D Cubes. Rendering " + cubeCount + " cubes.
Press spacebar to increase.";

```

```

}

private function onKeyDown(event:KeyboardEvent):void {
if(event.keyCode == 32) {
addCubes();
}
}

private function onMouseMove(event:MouseEvent):void {
mX = mouseX - windowHalfX;
mY = mouseY - windowHalfY;
}

private function onStageResize(event : Event) : void{
view.width = stage.stageWidth;
view.height = stage.stageHeight;

windowHalfX = stage.stageWidth/2;
windowHalfY = stage.stageHeight/2;

//center info text
infoText.x = (stage.stageWidth - infoText.textWidth)/2
}

private function onViewAdded(event : Event) : void{
// Mejorar el rendimiento mediante la desactivación de la comprobación
de //errores
view.stage3DProxy.context3D.enableErrorChecking = false
}

private function onEnterFrame(ev : Event) : void{

view.camera.x += (mX - view.camera.position.x) * 0.1;
view.camera.y += (-mY - view.camera.position.y) * 0.1;
//always look at center
view.camera.lookAt(cubeHolder.position);

cubeHolder.rotationY += rotSpeed;
for (var i:int = 0 ; i < cubeCount; i ++ ){
cubes[i].rotationX += rotSpeed;
}
view.render();
}
}
}
}

```

Código JavaScript Cubos usando Three.js

```

<script type="text/javascript">
//WebGL Cubos Demostración

if(!Detector.webgl)
    Detector.addGetWebGLMessage();

    var MATERIAL_COUNT = 30;
    var CUBE_SIZE = 300;
    var CUBE_INCREMENT = 100;
    var SPREAD = 1000;

    var scene;
    var renderer;
    var camera;
    var stats;
    var cubes = [];
    var materials = [];
    var cubeHolder;
    var rotSpeed;
    var cubeCount = 0;
    var geometry;
    var windowHalfX;
    var windowHalfY;
    var mouseX = 0;
    var mouseY = 0;

    init();

function init() {

    document.onselectstart = function() {
        return false;
    };
    //Inicio del Mundo 3D
    var container = document.createElement('div');
    document.body.appendChild(container);
    renderer = new THREE.WebGLRenderer({antialias : false,
    clearAlpha : 1});
    container.appendChild(renderer.domElement);
    camera = new THREE.PerspectiveCamera(55, window.innerWidth /
    window.innerHeight, 20, 3000);
    camera.position.z = 1000;
    scene = new THREE.Scene();
    scene.add(camera);
    //init crear objeto para la rotación
    cubeHolder = new THREE.Object3D();
    scene.add(cubeHolder);

    //Agregar luces
    var light = new THREE.PointLight(0xffeeaa, 1);
    light.position = new THREE.Vector3(-1000, 1000, -1000);

```

```

scene.add(light);

var light2 = new THREE.PointLight(0xFFFFFF, 1);
light2.position = new THREE.Vector3(1000, 1000, 1000);
scene.add(light2);

//init materiales
for(var i = 0; i < MATERIAL_COUNT; i++) {
    var material = new THREE.MeshLambertMaterial({
        opacity : 0.5,
        blending : THREE.AdditiveBlending,
        depthTest : false,
        transparent : true
    });
    material.color= new THREE.Color(0xFFFFFF * Math.random());
    materials.push(material)
}

//init cubos
geometry=new THREE.CubeGeometry(CUBE_SIZE, CUBE_SIZE, CUBE_SIZE);
addCubes();
//agregar estados
stats = new Stats();
stats.domElement.style.position = 'absolute';
stats.domElement.style.top = '0px';
container.appendChild(stats.domElement);
//Igualar la velocidad con la versión Stage3D en grados
rotSpeed = .3 / 360 * Math.PI * 2;
window.addEventListener('resize', onWindowResize, false);
window.addEventListener('mousemove', onMouseMove, false);
window.addEventListener('keydown', onKeyDown, false);
onWindowResize(null);
animate();
}

function addCubes() {
    cubeCount += CUBE_INCREMENT;

    //init cubos
    for(var j = 0; j < CUBE_INCREMENT; j++) {
        var cube = new THREE.Mesh(geometry, materials[j %
MATERIAL_COUNT]);

        //Crear tamaños aleatoriamente
        cube.scale.x=cube.scale.y=cube.scale.z = Math.random()+.1;

        cubeHolder.add(cube);
        cubes.push(cube);

        cube.position.x = Math.random() * SPREAD - SPREAD / 2;
        cube.position.y = Math.random() * SPREAD - SPREAD / 2;
        cube.position.z = Math.random() * SPREAD - SPREAD / 2;

        cube.rotation.x = Math.random() * 2 * Math.PI - Math.PI;

```

```

        cube.rotation.y = Math.random() * 2 * Math.PI - Math.PI;
        cube.rotation.z = Math.random() * 2 * Math.PI - Math.PI;
    }

    //Crear cubos opacos
    for(var i = 0; i < MATERIAL_COUNT; i++) {
        materials[i].opacity = 50 / cubeCount;
    }

    //Texto de información
    document.getElementById("info").innerText = "Cubos WebGL.
    Renderizando " + cubeCount + " cubos. Presione la barra
    espaciadora para agregar más cubos.";
}

function onKeyDown(event) {
    if(event.keyCode == '32') {
        addCubes();
    }
}

function onMouseMove(event) {
    mouseX = event.clientX - windowHalfX;
    mouseY = event.clientY - windowHalfY;
}

function onWindowResize(event) {
    windowHalfX = window.innerWidth / 2;
    windowHalfY = window.innerHeight / 2;
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize(window.innerWidth, window.innerHeight);
}

function animate() {
    requestAnimationFrame(animate);
    render();
    stats.update();
}

function render() {
    camera.position.x += (mouseX - camera.position.x) * 0.1;
    camera.position.y += (-mouseY - camera.position.y) * 0.1;

    camera.lookAt(cubeHolder.position);

    cubeHolder.rotation.y -= rotSpeed;
    for(var i = 0; i < cubeCount; i++) {
        cubes[i].rotation.x += rotSpeed;
    }
    renderer.render(scene, camera);
}
</script>

```

ANEXO 2: ANÁLISIS FINANCIERO

Las empresas en los últimos años no han tenido la posibilidad de estudiar los pro y los contra de los diferentes sistemas de diseño 3D existentes en el mercado debido a que la mayoría de estos necesitaban de licencias las cuales eran demasiado costosas para ser adquiridas por empresas que recién estaban iniciando en el mundo de diseño y animaciones 3D.

En la actualidad las tecnologías están en constante desarrollo tanto es así que los desarrolladores han logrado crear etiquetas en el caso de HTML5 para poder ejecutar animaciones 2d y 3D en el navegador esto ha sido aprovechado por tecnologías de terceros creando librerías JavaScript capaces de crear iluminaciones, cámaras en ambiente 3D sin la necesidad de pagar por licencias, la desventaja de esto es que el personal de las empresas no tienen un conocimiento extenso del tema debido a que la tecnología es nueva, por lo que se requiere de capacitaciones y más tiempo invertido en las animaciones.

Algunas empresas mantuvieron en reserva la información acerca del pago de licencias y sueldo a empleados por lo que se utilizaran valores similares a las de las demás empresas, para no alejarnos demasiado de la realidad en cuanto a valores entre las distintas instituciones.

A continuación estudiaremos los gastos que realiza una empresa de desarrollo 3D, que utiliza software pagado y realizaremos un análisis de cómo cambiarían sus gastos en el caso de implementar la tecnología gratuita a su suite de desarrollo.

Encuesta StudioFX

En la encuesta realizada a la empresa StudioFX se registraron los siguientes resultados.

En cuanto a programas la empresa realiza una inversión inicial de 21081,18 debido a que las licencias son adquiridas por primeras, las mismas que podrán ser utilizadas el tiempo que desee la persona con esa versión de software, en caso de que exista una actualización o versión del software el empresario necesitara pagar para hacer uso de dicha actualización o versión.

Programas				
Programas	Número de Copias	P. Unit. Lic. Nueva	Precio Actualización	Precio Total
3DS Max	2	5.706,18	2.853,77	11412,36
AutoCAD	1	6.988,50	3.512,36	6988,5
Flash	2	699,00	199,00	1398
Total	5	13393,68	6.565,13	19798,86

Tabla 8 Tabla de precios de licencias adquiridas por primera vez

Con el uso de HTML5², y sin el pago de ninguna actualización los gastos en software disminuyen radicalmente.

Programas				
Programas	Número de Copias	P. Unit	Precio Actualización	Precio Total
3DS Max	1	5.706,18	2.853,77	5706,18
AutoCAD	1	6.988,50	3.512,36	6988,5
HTML5				0
Total	2	12694,68	6.366,13	12694,68

Tabla 9 Tabla de precios de licencias de actualizaciones

Sueldos persona desarrollo y diseño

En cuanto a gastos que realiza la empresa, nos facilitaron únicamente datos acerca de las personas que cumplen las tareas de desarrollo y diseño. Dando como gasto mensual 1800\$

Sueldos Diseñadores/Programadores				
Puesto	Número de personas	Sueldo		
Programadores	2	300	Mensual	600
Diseñadores	4	300	Mensual	1200
Total	6	7200	anual	21600

Tabla 10 Pagos de sueldos empleados

Ingresos de la empresa

En cuanto a ingresos las empresas fueron muy reservadas, debido a que mantienen en reserva estos datos nos facilitaron algunos datos con los cuales tratamos de enfocarlos a los más real posible, teniendo en cuenta valores de licencias pagos a empleados, etc. Que son factores que influyen en los precios de los productos generados.

Ingresos				
Diseños/Animaciones	Valor	Cant. aprox. Pedidos		Valores totales
Páginas Web	550	10	anuales	5500
Diseños 3D	100	15	anuales	1500
Diseños 2D	50	20	mensual	12000
Animaciones 2D	250	25	anuales	6250
Animaciones 3D	500	10	anuales	5000
Total	1450	80		30250

Tabla 11 Tabla de ingresos de la empresa StudioFX

Gastos Capacitaciones

En cuanto a capacitaciones la empresa trata de mantener actualizado a su personal, por lo que las capacitaciones son pagadas sin ningún problema, pero con se lleva a cabo la firma de un contrato de honestidad y fidelidad para evitar que la persona tome el curso y al siguiente mes presenta la renuncia para cambiarse de empresa.

Gastos Capacitaciones			
Cursos	Número de personas	Costo	Gastos
3DS Max	2	700	1400
AutoCAD	2	900	1800
Flash	2	1100	2200
Total	6	2700	5400

Tabla 12 Pago capacitaciones al personal

En cuanto a ganancias percibidas por parte de la empresa, teniendo en cuenta que los datos brindados son solo los que tienen que ver con animaciones y diseños 2D y 3D y además usando software pagados. Son los siguientes.

Ganancias sin HTML5		
Ganancias Soft. Lic. Nuevas	Ganancias Soft. Lic. Act.	Ganancias Soft. Gratuito
-16548,86	-3.315,13	3.250,00

Tabla 13 Ganancias de la empresa sin utilizar HTML5

Una vez la empresa deja de usar un programa con licencias y pasa a usar WebGL⁸ y HTML5² las ganancias suben.

Ganancias con HTML5		
Ganancias Soft. Lic. Nuevas	Ganancias Soft. Lic. Act.	Ganancias Soft. Gratuito
-8644,68	-2.316,13	4.050,00

Tabla 14 Ganancias de la empresa utilizando HTML5

En cuanto a diferencia en los ingresos anuales, entre las distintas tecnologías notamos grandes cambios, teniendo en cuenta que los ingresos tomados son solo a lo referente con diseño de páginas y animaciones 2D, 3D. Es decir que aparte de estos ingresos las empresas perciben otros, los mismos valores que se mantuvieron en reserva en la investigación, por lo que tenemos que en los inicios de la empresa se perciben pérdidas de 8644.68 debido a que las licencias son compradas por primera vez, en el caso de una actualización se mantienen pérdidas de 2.316.13, pero en el momento que no se realizan actualizaciones de

licencias las ganancias suben a 4050, lo mismo que se lleva con software gratuito.

Encuesta Equilibrio

Gastos e Ingresos sin Software gratuito

En el caso de la segunda empresa encuestada, realiza un gasto anual de 14087.12\$ en la primera compra de licencias, en caso de realizar únicamente actualizaciones de licencias el gasto disminuye a 3711.36\$.

Programas				
Programas	Número de Copias	Precio Unitario	Precio Actualización	Precio Total
AutoCAD	1	6.988,50	3.512,36	6988,5
Flash	2	699,00	199,00	1398
Maya	1	5.700,62		5700,62
Total	4	13.388,12	3.711,36	14087,12

Tabla 15 Costos de programas con licencias – Equilibrio.

En cuanto a gastos realizados por pagos a los empleados a cargo del desarrollo y diseño la empresa realiza los siguientes gastos:

Sueldos Diseñadores/Programadores				
Puesto	Número de personas	Sueldo		
Programadores	3	500	Anual	18000
Diseñadores	4	450	Anual	21600
Total	7	950	Anual	39600

Tabla 16 Sueldos a empleados de la empresa.

La empresa **Equilibrio** lucha por mantenerse a la vanguardia de la tecnologías por lo que mantiene permanentemente capacitando al personal en nuevas tecnologías que puedan explotar y con esto beneficiar sus ingresos anualmente por lo que mantiene los siguientes gastos:

Gastos Capacitaciones			
Cursos	Número de personas	Costo	Gastos
AutoCAD	1	900	900
Flash	2	1100	2200
Maya	2	1500	3000
Total	5	3500	6100

Tabla 17 Pagos capacitaciones al personal de la empresa – Equilibrio.

Los ingresos anuales de la empresa utilizando software pagado o actualizando el uso de licencias son los siguientes:

Ingresos			
Diseños/Animaciones	Valor	Cantidad aprox. Pedidos	Valores totales
Páginas Web	650	20 Mensual	13000
Diseños 3D	150	15 Anual	27000
Diseños 2D	75	25 Anual	1875
Animaciones 2D	300	15 Anual	4500
Animaciones 3D	600	15 Anual	9000
Total	1775	90	55375

Tabla 18 Ingresos anuales de la empresa equilibrio utilizando licencias.

Gastos e Ingresos con Software gratuito

Pero en el caso de que no se realicen actualizaciones de ningún tipo y se utilice la herramienta gratuita como es WebGL⁸ o Blender los gastos disminuyen a 12688.57\$.

Programas				
Programas	Número de Copias	Precio Unitario	Precio Actualización	Precio Total
AutoCAD	1	6.988,50	3.512,36	6988,5
HTML5				
Maya	1	5.700,07		5700,07
Total	2	12688,57	3512,36	12688,57

Tabla 19 Pago licencias anuales empresa, usando HTML5.

Como lo mencionamos anteriormente la empresa siempre busca estar a la vanguardia con la tecnología, por lo que no le importa pagar por capacitaciones a su personal, con software pagado las capacitaciones son costosas por el marca, en el caso de software gratuito las capacitaciones

serán un poco más baratas lo que es un ahorro para la empresa el mismo que lo puede invertir en otros negocios o en mejorar el hardware o el software de la empresa.

Los gastos de capacitaciones con software gratuito son los siguientes:

Gastos Capacitaciones			
Cursos	Número de personas	Costo	Gastos
AutoCAD	1	900	900
WebGL	2	700	1400
Maya	2	1500	3000
Total	5	3100	5300

Tabla 20 Pagos capacitaciones al personal HTML5

En comparación con las ganancias que se obtienen anualmente, se tienen diferencias extremadamente grandes teniendo en cuenta que estos ingresos son solo en cuanto a productos que tienen que ver con diseño y animaciones 2D y 3D, la empresa también percibe ganancias por otros tipos de productos.

En cuanto a ganancias adquiriendo licencias completamente nuevas se perciben pérdidas, teniendo en cuenta que la compra se realiza de todas las licencias en un solo año.

Pagando actualizaciones de las licencias y utilizando HTML5, teniendo en cuenta que se realiza la actualización de todas las licencias en un mismo año, sabiendo que las actualizaciones de los programas pueden tardar de uno a 2 años en salir.

Cuando la empresa deja de pagar por licencias y actualizaciones las ganancias por parte de la aumentan casi en un 40%.

Comparación Ganancias sin HTML5		
Ganancias Soft. Nuevas	Ganancias Soft. Act.	Ganancias Soft. Gratuito
-4412,12	3.105,86	9.675,00

Tabla 21 Ingresos de la empresa utilizando HTML5.

En cuanto a ganancias que percibe la empresa anualmente con el uso de HTML5² y eliminando algunas herramientas las cuales puede reemplazar HTML sin problema.

Comprando licencias completamente nuevas se tienen pérdidas de 2213.57\$.

Realizando actualizaciones de los programas utilizados y si el uso de herramientas que reemplaza HTML5 se perciben ganancias por lo que la empresa ya tiene la posibilidad de pagar por actualizaciones lo que con el uso de herramientas pagadas se mantenían perdidas.

Utilizando HTML y sin pagar ningún tipo de actualizaciones las ganancias suben en un 40%, ganancias que el gerente puede invertir en otro tipo de negocios o herramientas para mejor el desempeño de la empresa.

Comparación Ganancias con HTML5		
Ganancias Soft. Nuevas	Ganancias Soft. Act.	Ganancias Soft. Gratuito
-2213,57	6.962,64	10.475,00

Tabla 22 Comparación Ganancias entre software pagado y gratuito.