

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
Facultad de Ciencias Químicas e Ingeniería
Maestría y Doctorado en Ciencias e Ingeniería



Entorno para Diseño de Contenido Multimedia Reproducibile en Sistemas Embebidos

TESIS

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS

Presenta:

ADRIÁN ANTONIO CAMACHO GARCÍA

Director:

DR. LEOCUNDO AGUILAR NORIEGA

TIJUANA, BAJA CALIFORNIA, MÉXICO

JUNIO 2015

Universidad Autónoma de Baja California
FACULTAD DE CIENCIAS QUÍMICAS E INGENIERÍA
COORDINACIÓN DE POSGRADO E INVESTIGACIÓN

FOLIO No. 143

Tijuana, B. C., a 25 de mayo de 2015

C. Adrián Antonio Camacho García
Pasante de: Maestro en Ciencias
Presente

El tema de trabajo y/o tesis para su examen profesional, en la
Opción TESIS

Es propuesto, por el C. Dr. Leocundo Aguilar Noriega

Quien será el responsable de la calidad del trabajo que usted presente, referido al
tema: Entorno para Diseño de Contenido Multimedia Reproducible en Sistemas
Embebidos.

el cual deberá usted desarrollar, de acuerdo con el siguiente orden:

- I.- INTRODUCCIÓN
- II.- MARCO TEORICO
- III.- PLATAFORMA DE CONTENIDO MULTIMEDIA
- IV.-DESARROLLO DEL ENTORNO DE CONTENIDO MULTIMEDIA
- V.- CASOS DE PRUEBA
- VI.- CONCLUSIONES Y TRABAJO FUTURO



Q. Noemí Hernández Hernández
Secretario

UNIVERSIDAD AUTÓNOMA
DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS
QUÍMICAS E INGENIERÍA



Dr. Leocundo Aguilar Noriega
Director de Tesis



Dr. Luis Enrique Palafox Maestre
Director

Agradecimientos

A mi director de tesis Dr. Leocundo Aguilar, por haberme dejado entrar a su grupo de investigación siendo un extraño, y bueno, que al final hasta proyecto de vinculación me tocó. Gracias la confianza y la enseñanza. Así como también al Dr. Licea por haberme apoyado en este trayecto.

Al Dr. Juan Antonio Holgado por haberme aceptado y apoyado en la estancia, así como a su grupo de investigación, son unas excelentes personas y profesionales, aprendí mucho tanto profesionalmente como personalmente, además que vivir ese tiempo en España me dio mucha experiencia y perspectiva sobre lo que sigue.

Principalmente agradezco a mi familia (dentro de ella mi novia) por todo su apoyo no solo en la maestría, sino en la trayectoria completa.

A mis compañeros del programa, por empujarme sobretodo en esos seminarios y finalmente esta tesis.

CONACYT y UABC muchas gracias.

Resumen

Este trabajo busca colaborar con la propuesta de una plataforma tecnológica para manejo de contenido multimedia que pueda ser interactuado por eventos de movimientos físicos. Para esto, se desarrolla un entorno para diseño de contenido multimedia así como la estructura y formato de datos que lo representa y que lo puede hacer interoperable hacia un sistema embebido capaz de desplegar multimedia mediante un medio visual así como de reaccionar a eventos externos predefinidos.

La intención de este proyecto va enfocada como apoyo en el proceso de enseñanza y aprendizaje en el campo educativo del nivel preescolar, dando al educador la posibilidad de crear contenido multimedia con la capacidad de reacción a eventos físicos realizados por los niños.

Para este trabajo se manejan dos etapas principales. La primera de ellas que consiste en el desarrollo del entorno de diseño de contenido multimedia, y la segunda que consiste en la definición de una estructura de datos y un formato de datos para el contenido, que permita la interoperabilidad hacia un sistema embebido que pueda desplegar multimedia así como recibir instrucciones predefinidas desde un dispositivo externo.

Abstract

This work seeks to collaborate with the proposal of a technological platform for handling multimedia content that can be interacted with by physical motion events. For this, a multimedia content designing environment is developed as the data model and format that represents it and can make it interoperable to an embedded system capable of displaying multimedia through a visual media and capable of react to predefined external events.

The intention of this project is focused as a support for the teaching and learning process in preschool level education field, giving the teacher the possibility of creating multimedia content with the ability to react to physical events performed by children.

For this work, two main phases are handled. The first one consists in developing a multimedia content designing environment, and the second one consisting of the definition of a data structure and a content data format to allow interoperability to an embedded system that can deploy multimedia and receive predefined instructions from an external device.

Índice general

1. Introducción	1
1.1. Motivación	1
1.2. Planteamiento del problema	2
1.3. Propuesta	4
1.4. Objetivos	4
1.4.1. Objetivo General	4
1.4.2. Objetivos Particulares	5
1.5. Estructura de la Tesis	5
2. Marco teórico	7
2.1. Tecnologías de la Información y La Comunicación	7
2.2. TIC en la Educación	8
2.3. Educación Preescolar	9
2.4. TIC's Existentes en la Educación	10
2.4.1. Kinect	11

2.4.2. Wiimote	11
2.4.3. Smart Fit Park	12
2.4.4. Scratch	13
2.4.5. Mario Maker	14
2.5. Dispositivo de Hardware DAVID PEP	15
2.5.1. Introducción	16
2.5.2. Descripción de uso	17
2.5.3. Arquitectura	17
2.6. Manejo del Contenido Multimedia	22
2.6.1. Bitmap	23
2.6.2. Serialización de Datos	23
2.6.3. Formato Contenedor de Multimedia	26
3. Plataforma de Contenido Multimedia	27
3.1. Introducción	27
3.2. Requisitos Generales	30
3.3. Actualidad	31
3.4. Alcances	32
4. Desarrollo del Entorno de Contenido Multimedia	33
4.1. Diseño de Solución	33
4.2. Estructura del Contenido	35

4.3. Entorno de Diseño de Contenido	37
4.3.1. Requisitos	38
4.3.2. Calidad del Software	42
4.3.3. Metodología	43
4.3.4. Arquitectura y Diseño de Software	44
4.3.5. Patrones de Software	47
4.3.6. Diseño de Interfaz Gráfica de Usuario	47
4.3.7. Codificación	48
4.3.8. Validación y Verificación	51
4.4. Interoperabilidad	52
4.4.1. Interoperabilidad del Contenido	52
4.4.2. Interoperabilidad con DAVID PEP	54
5. Casos de Prueba	56
5.1. Caso 1	56
5.1.1. Introducción	56
5.1.2. Descripción	57
5.1.3. Resultados	57
5.2. Caso 2	58
5.2.1. Introducción	58
5.2.2. Descripción	58
5.2.3. Resultados	58

6. Conclusiones y Trabajo Futuro	60
6.1. Conclusiones	60
6.2. Trabajo Futuro	62

Índice de figuras

2.1. Etapas del desarrollo de la inteligencia según Piaget.	10
2.2. Kinect.	11
2.3. Wiimote.	12
2.4. Smart Fit Park.	13
2.5. Scratch.	14
2.6. Mario Maker.	15
2.7. Conectividad de DAVID PEP	17
2.8. Tarjeta principal de DAVID PEP	18
2.9. Tarjeta principal de DAVID PEP	19
2.10. Arquitectura multicore del microcontrolador	19
2.11. Pulsera con Nodo S de la Plataforma LiSANDRA	20
2.12. Manejo de multiplanos por DAVID PEP	21
2.13. Imagen resultante de multiplanos superpuestos	22
3.1. Diagrama General.	29

4.1. Visión del Entorno Propuesto.	34
4.2. Estructura del Contenido.	36
4.3. Estructura.	37
4.4. Plano Resultante.	38
4.5. Requisitos Funcionales.	39
4.6. Requisitos No Funcionales.	42
4.7. Flujo de la Aplicación.	45
4.8. Modelo de Clases.	46
4.9. Software Creador de Contenido.	48
4.10. Arquitectura de JavaFX.	50
4.11. Cambio en el Manejo de Gráficos.	51
4.12. Formatos de Serialización. XML, JSON, YAML.	53
4.13. Comparación de tamaño de archivo con la estructura de datos.	54
4.14. Media de tamaño de archivo con la estructura de datos.	54
4.15. Formato de contenido para DAVID PEP.	55
5.1. Entorno de diseño de Contenido.	57
5.2. Entorno de diseño de Contenido.	59
5.3. Conteido siendo Reproducido con David Pep.	59

Capítulo 1

Introducción

1.1. Motivación

La motivación de este proyecto viene de la finalidad de colaborar con la propuesta de una plataforma tecnológica (hardware y software) que permita diseñar de forma sencilla contenido multimedia que pueda reaccionar a eventos físicos predefinidos. Este proyecto nace de dos trabajos de tesis anteriores, uno en el cual se desarrolló un dispositivo de hardware de bajo costo, el cual es programable y tiene la capacidad de manejar contenido multimedia que puede desplegarse con la ayuda de un medio visual, y otro en el cual se desarrolló un dispositivo tipo pulsera que permite sensor movimientos. Estos dos trabajos pueden trabajar en conjunto, pues la pulsera se puede integrar como aditamento al dispositivo de manejo de contenido multimedia y dicha pulsera permite a un usuario interactuar con el contenido multimedia.

Como el contenido multimedia del dispositivo de hardware solo puede ser creado mediante instrucciones de programación propias, es que se busca una solución que permita el diseño de contenido multimedia de una forma sencilla y que pueda ser interactuable con el dispositivo

y a través de éste con los eventos externos.

Parte de la motivación, es el poder contribuir en la aplicación de las tecnologías de la información en la educación, pues se busca que este trabajo pueda apoyar en la educación preescolar, al brindar una plataforma que permita a los educadores diseñar contenido multimedia que pudiera ser interactuado por sus educandos.

1.2. Planteamiento del problema

En la actualidad, se ha observado que la aplicación de las tecnologías de la información en la educación han beneficiado al proceso de enseñanza y aprendizaje. En este caso particular, se ha observado que en la educación preescolar es posible utilizar figuras geométricas para explicar algún tema y que estas figuras se pueden representar a través de dibujos mediante herramientas digitales como un software de presentaciones, aunque en estos casos se minimiza la interactividad a los educandos, por lo que buscamos una plataforma tecnológica que le permita al educador diseñar el contenido multimedia y que al integrarse con los dispositivos de hardware pueda permitir interactividad al educando.

Para dicha plataforma, se cuenta ya con un dispositivo electrónico de hardware (un sistema embebido) que permite desplegar contenido multimedia, pero que solo permite crear los elementos multimedia mediante instrucciones de programación en el dispositivo, por lo que se requiere de un medio que permita diseñar un contenido de forma rápida y sencilla que permita interactuar con el dispositivo y a través del dispositivo con eventos externos, logrando tener así las partes que pueden conformar la plataforma.

Parte del problema, además de no contar con un medio para crear el contenido de una forma sencilla, es que no se cuenta con una estructura de datos que permita manejar el contenido de manera práctica ni una forma en la que pueda interoperar el contenido con el

dispositivo. Y que aunque es común que en sistemas computacionales existen mecanismos muy comunes para lograr la comunicación entre los elementos que lo integran (ya sean en sistemas homogéneos o heterogéneos) donde a través de protocolos y formatos se realiza el intercambio de datos utilizando lenguajes y formatos de intercambio de datos como XML, JSON o YAML, entre otros, por el lado de los sistemas embebidos y principalmente los de recursos muy limitados es probable que estas técnicas no sean tan eficientes al manejar gran cantidad de datos. Por lo que es una problemática con la se contará en el momento de buscar la interoperabilidad con el dispositivo de hardware, principalmente si se busca una forma sencilla de manejar la información.

De modo que por los requerimientos de este trabajo surge la necesidad del manejo de datos entre aplicaciones heterogéneas, en este caso de un sistema de computadora personal hacia un sistema embebido para poder distribuir la información de una manera sencilla y sin afectar en gran parte el rendimiento del sistema embebido (por recursos mas limitados que éstos manejan).

Por lo que se busca una forma para que un contenido diseñado por una aplicación de computadora que maneje el contenido en diferentes planos, tenga la capacidad de ser interpretado por un sistema embebido, es decir, buscar la interoperabilidad entre los datos de el que diseña el contenido (computadora personal) y quien lo utiliza (sistema embebido).

Para esto, también es requerido de la adecuación de programación en el sistema embebido para la interpretación del contenido diseñado, utilizando sus propias funciones y que, debido al continuo crecimiento de recursos y la minimización de costos en el hardware es que se busca que la forma de manejar el contenido pueda ser amigable, brindando implementaciones a futuros sistemas embebidos que puedan mejorar en recursos y costos.

Esto es con el fin de contribuir a una plataforma que formada por hardware y software permita ofrecer una herramienta que pueda apoyar en la educación.

1.3. Propuesta

Se plantea el desarrollo de un entorno de diseño multimedia que además de crear y editar el contenido, permita transformarlo para que pueda ser ejecutado en un sistema embebido el cual permite desplegar el contenido a un medio visual, generando una plataforma tecnológica con la finalidad de apoyar en el proceso de enseñanza y aprendizaje en el nivel de educación preescolar, en donde el educador pueda utilizar el entorno para diseño de contenido multimedia y haciendo uso de figuras básicas, letras y colores, pueda crear una idea y plasmarla sobre la pantalla, de forma semejante a lo que hace al preparar su material sobre cartulinas o en un software de presentaciones, a diferencia de que en este entorno se plantea la posibilidad de interactividad con los educandos. Dicha interactividad se plantea agregando elementos multimedia en la herramienta con capacidad de reacción a eventos predefinidos, así como con la interpretación y funcionalidad del sistema embebido y de un dispositivo electrónico tipo pulsera. Esta funcionalidad de interactividad se muestra en este trabajo como una avance para integrarlo en la plataforma, pero queda su completa implementación como continuación a este trabajo.

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar un entorno de diseño multimedia que permita manejar contenido multimedia desde un software de computadora hacia un sistema embebido, enfocado en obtener una plataforma tecnológica en apoyo para la educación preescolar.

1.4.2. Objetivos Particulares

- Proponer y desarrollar un entorno de diseño multimedia que permita crear, manejar y transformar contenido multimedia.
- Definir y aplicar la estructura de datos que permita el almacenamiento y portabilidad de contenido multimedia.
- Definir y aplicar un formato de datos a utilizar que sea interoperable con un sistema embebido.
- Una versión prototipo de la herramienta en conjunto con un sistema embebido.

1.5. Estructura de la Tesis

Este documento de tesis se encuentra organizado en seis capítulos que se explican brevemente a continuación.

- **Capítulo 1:** En este primer capítulo se presenta la introducción del trabajo, para esto se menciona la motivación sobre el origen del trabajo, así como la problemática encontrada en este trabajo, la propuesta, los objetivos y por último la estructura de este documento.
- **Capítulo 2:** En este capítulo se menciona el marco teórico de la investigación realizada. Se habla de lo general y lo particular en cuanto a las tecnologías de la información aplicadas en la educación, así como de los sistemas existentes que en ciertos aspectos se pueden relacionar con este trabajo. También se menciona sobre el hardware existente que se utiliza en esta propuesta, y por último generalidades sobre manejo del contenido.

- **Capítulo 3:** En este capítulo se plantea la propuesta para la integración de la plataforma, utilizando los dispositivos de hardware existentes y que propone el desarrollo para el entorno de diseño multimedia, el cual permite crear contenido multimedia a través de una aplicación de computadora y ser desplegado en un medio visual a través de un sistema embebido. Para esto se describe una introducción, los recursos actuales, los alcances de la propuesta en este trabajo y los requisitos.
- **Capítulo 4:** En este capítulo se muestra el desarrollo realizado para el entorno de diseño de contenido multimedia, así como las tecnologías utilizadas y la metodología que se ha seguido. Se muestra el diseño de la solución, la forma en la que se estructura el contenido, así como el desarrollo del software que representa el entorno de diseño de contenido multimedia y se describen la etapas principales del modelo de desarrollo. Por último se hace mención a la interoperabilidad entre los sistemas.
- **Capítulo 5:** En este capítulo se exponen casos de prueba, para el diseño de contenido multimedia y para demostrar la interoperabilidad entre los sistemas de la plataforma.
- **Capítulo 6:** En el capítulo seis se presentan las conclusiones del presente trabajo, así como trabajo futuro que se contempla como parte y continuación de este proyecto, mencionando trabajo sobre el entorno de diseño, sobre la plataforma, y finalmente sobre validación pedagógica.

Capítulo 2

Marco teórico

2.1. Tecnologías de la Información y La Comunicación

Las tecnologías de la información y la comunicación (TIC) son el conjunto de tecnologías que permiten el acceso, producción, tratamiento y comunicación de la información presentada en diferentes códigos (como texto, imagen y sonido) y éstas se desarrollan a partir de avances científicos en los campos de la informática y las telecomunicaciones [1].

En la actualidad las tecnologías de la información y la comunicación (TIC) se encuentran presentes en todos los niveles de la sociedad, desde grandes corporaciones hasta pequeñas empresas, así como en gobiernos, centros educativos, con profesionales y particulares, con tecnologías indispensables como lo son las computadoras, los teléfonos móviles, los navegadores de posicionamiento global y el internet [2] apoyando a diferentes sectores con los medios técnicos y procesos empleados que actúan sobre los datos y la información, así como con la transmisión de éstos.

Tal es el caso que en [3] se define a las TIC como “las tecnologías aplicadas a la creación, almacenamiento, selección, transformación y distribución de las diversas clases de informa-

ción, así como a la comunicación, utilizando datos digitalizados” apoyando esta definición a el párrafo anterior en la razón que las TIC se encuentran presente en todos los niveles de la sociedad, pues se utilizan en las fases del manejo de la información digital.

Algunas de las características que pueden representar a las TIC se mencionan en [4] y de las que principalmente se apoya este trabajo son:

- *Interactividad* – Posiblemente una de las características mas importantes de las TIC en la aplicación a el campo educativo, pues realiza el intercambio de la información entre el usuario y la computadora, adaptando los recursos a las necesidades y características de los sujetos, en función de la interacción concreta del sujeto con la computadora.
- *Innovación*– Ya que Las TIC están produciendo un cambio constante en todos los ámbitos sociales y tecnológicos.

2.2. TIC en la Educación

Las TIC en la educación han pasado a ser un punto clave, pues representan oportunidades beneficiosas en el proceso de enseñanza y aprendizaje ya que permiten nuevas formas de acceder, generar y transmitir información y conocimientos ofreciendo contenido educativos mas interactivos, atractivos y variados, dándole la oportunidad a los profesores de generar contenidos educativos de acuerdo con los intereses o las particularidades de sus alumnos y de su contexto educativo, así como también reforzar con materiales existentes [5].

En cuanto a las aulas o ambientes de aprendizaje que intentan incorporar tecnología digital sería ideal que ofrezcan la oportunidad al educando de elegir con libertad el material para aprender, como con niños menores de seis años, en que el principio de singularidad maximiza las características que hacen a un humano único e irrepetible, pues en esa etapa

se define la personalidad, por lo que se dice que la tecnología y los avances tecnológicos se deben de adaptar a los niños y no en forma contraria [6].

Por lo que con la aplicación de las TIC en la educación se pueden alcanzar mejores servicios educativos y con la generación de contenido multimedia interactiva, así como con el diseño de nuevas estrategias educativas por parte de los profesores, alcanzar una mejor oferta educativa [7].

2.3. Educación Preescolar

En México, la educación preescolar se pretende para la población de tres a cinco años y se reconoce como un servicio que puede contribuir a mejorar las oportunidades académicas de los niños a lo largo de su vida, siendo sus beneficios posibles cuando ésta es de calidad, creando las condiciones para ello tanto de recursos materiales, humanos y organizativos, como de procesos adecuados a las necesidades de aprendizaje y desarrollo de los alumnos [8] [9] .

En investigaciones recientes se demuestra que en los primeros cinco años de edad la capacidad de aprendizaje de los niños es mas intensa, logrando un gran desarrollo de las capacidades intelectuales, emocionales, lingüísticas y sociales, por lo que la educación preescolar es de suma importancia pues se encuentra en la etapa en la que los niños tienen disposición natural para aprender y aprovechar de las interacciones que promueva el personal docente para apoyar en el desarrollo y aprendizaje [8].

Y en cuanto al desarrollo del pensamiento infantil y la adquisición de los conocimientos según Piaget se dividen en cuatro etapas (ver Figura 2.1), en donde la segunda que abarca de los dos a los siete años es conocida como etapa preoperacional y es el periodo en el que abarca la educación infantil y es cuando el niño adquiere las funciones simbólicas de representación de la realidad como el lenguaje, el juego simbólico, el dibujo y la escritura [10].

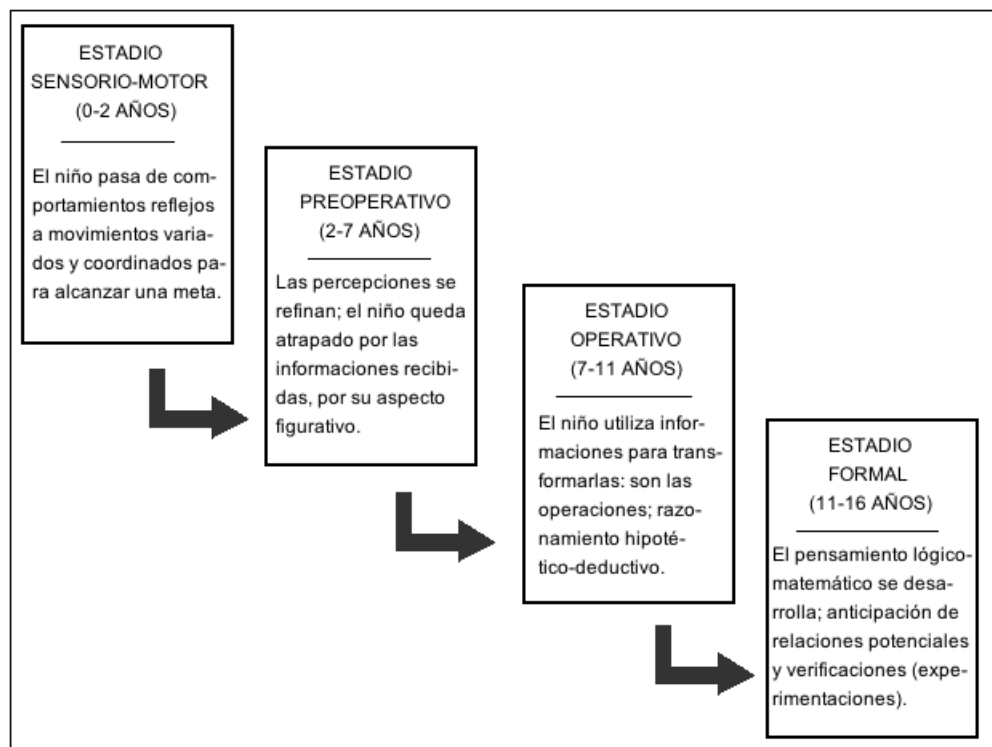


Figura 2.1 Etapas del desarrollo de la inteligencia según Piaget.

2.4. TIC's Existentes en la Educación

En la actualidad existen sistemas y entornos tecnológicos que pueden apoyar en la aplicación de las tecnologías de la información hacia los ambientes educativos en los cuales se pueden encontrar algunos puntos con cierta similaridad para este trabajo (como es el utilizar la interacción física como controlador, el diseño personalizado del contenido), aunque cabe mencionar que en lo que sí difieren es la visión, el contexto y los objetivos (pues ninguno está enfocado a la educación en nivel preescolar, y no permiten al educador que de una forma sencilla y sin conocimientos de programación pueda crear contenido personalizado para sus alumnos). Entre estos sistemas y entornos tecnológicos se pueden encontrar Kinect, Wiimote, Smart Fit Park, Scratch y Mario Maker y se mencionan a continuación.

2.4.1. Kinect

El Kinect es un controlador de juego libre desarrollado por Microsoft para las consolas Xbox 360 y Xbox One, así como para Windows (ver Figura 2.2). Este dispositivo permite a sus usuarios controlar e interactuar como un control de mando tradicional, pero sin éste, pues el control es a través de reconocimiento de movimientos, gestos y voz, pues Kinect cuenta con cámara y sensores. Y en Windows se cuenta con un SDK que permite desarrollar aplicaciones de Kinect para Windows [11].

En cuestión educativa, Kinect también puede participar, al brindar la conexión físico y mental con el videojuego “Body and Brain Connection”, así como apoyar la educación física con el videojuego “Kinect Sports” y fomentar el desarrollo social y la colaboración con el videojuego “Kinect adventures” [12].



Figura 2.2 Kinect.

Esta plataforma ha presentado muchos avances en cuestión de tecnología y muchos de sus juegos pueden apoyar en el área educativa, pero si un educador quisiera crear algún videojuego o contenido en base a sus necesidades sería de alta complejidad.

2.4.2. Wiimote

El Wiimote (Wii Remote) es el principal medio de comandos para la consola de Nintendo Wii. Éste es un control de mando inalámbrico, similar a un control de televisión, que además de botones posee detección de movimiento (ver Figura 2.3).

De igual forma que Kinect, esta plataforma ha presentado muchos avances en cuestión de



Figura 2.3 Wiimote.

tecnología y algunos de sus juegos podrían apoyar en el area educativa, pero si un educador quisiera crear algún videojuego o contenido en base a sus necesidades, también sería de alta complejidad.

2.4.3. Smart Fit Park

Smart Playground es un dispositivo para televisión tipo “plug and play” que acerca a los niños a mantenerse físicamente activos mientras aprenden (ver Figura 2.4). Los niños pueden caminar, correr, saltar y pisar mientras juegan hasta diez encuentros de aprendizajes, juegos y carreras. El Smart Playground enseña letras, números, formas y colores, entre otras, a través de la actividad con el juego. El niño controla la actividad en pantalla (mientras más rápido corren, su personaje se moverá mas rápido) [13].

Este producto se asemeja a nuestra propuesta en la forma de proponer la interacción física con el niño para aprender, aunque la diferencia radica en que el contenido son videojuegos creados por expertos en dicha área, y en nuestra propuesta, es proporcionar al educador la posibilidad de crear su propio contenido de una forma sencilla.



Figura 2.4 Smart Fit Park.

2.4.4. Scratch

Scratch es un entorno y una comunidad en línea para que niños puedan programar y compartir medios interactivos como cuentos, juegos y animaciones con personas de todo el mundo (ver Figura 2.5). Con Scratch los niños aprenden a pensar de una forma creativa, trabajar en colaboración, y razonar sistemáticamente. Scratch es diseñado y mantenido por el grupo Lifelong Kindergarten del MIT Media Lab [14].

Este producto refleja bastante potencial al brindar herramientas para crear contenido propio, aunque no de forma sencilla, pues es requerido tener conocimientos de programación, esto, por su enfoque que es orientado al desarrollo de las habilidades de programación a los niños. Por esto es diferente en cuanto a la propuesta de este trabajo, pues va enfocada a proporcionar las herramientas al educador para crear de forma sencilla contenido educativo para sus alumnos.

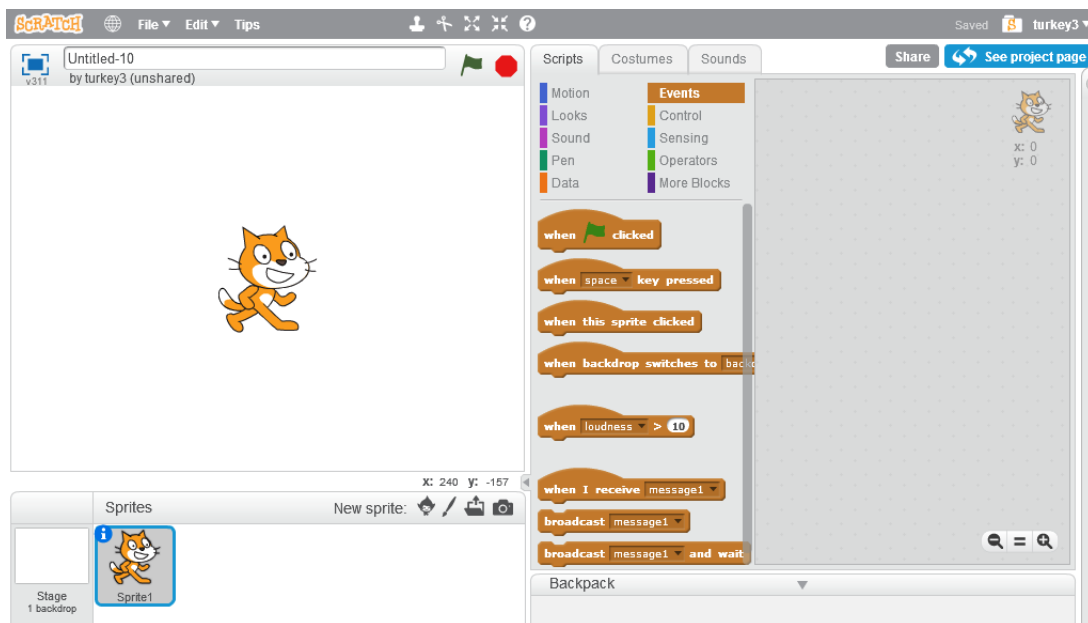


Figura 2.5 Scratch.

2.4.5. Mario Maker

Este es una plataforma que permite al usuario crear niveles del videojuego de Super Mario Bros (ver Figura 2.6). Provee de funciones que permiten agregar objetos del videojuego y así ir creando los niveles. Este producto se podrá usar el Wii U y su fecha de lanzamiento es para el 2015 [15].

Por lo que en este producto encontramos la similaridad de dar al usuario la herramientas para que el cree su contenido y que incluso se menciona en [15] que el único limite es la imaginación. Buscamos que con nuestro entorno se puedan brindar a los educadores las herramientas necesarias para crear los contenidos.



Figura 2.6 Mario Maker.

2.5. Dispositivo de Hardware DAVID PEP

En cuestión de hardware para este proyecto, se cuenta con uno como base que permite la continuidad hacia este proyecto, y en esta sección se le describe. Dicho hardware a utilizar se le conoce como DAVID PEP y una de las ventajas de seleccionar este dispositivo de hardware es que puede utilizar como aditamento a un Dispositivo de Detección de Movimientos (ambos fueron propuestas por tesistas que pertenecieron en el mismo grupo de investigación y se realizaron pruebas de comunicación entre estos dispositivos de hardware) y que se plantea en trabajo futuro la completa integración de la interactividad con el Dispositivo de Detección de Movimientos a través de DAVID PEP. Cabe mencionar que aunque se cuenta con un hardware específico, también se busca que la comunicación de los datos que se le envíen de contenido multimedia queden de una forma con bajo acoplamiento, buscando utilizar técnicas que permitan fácil interoperabilidad hacia otros dispositivos de hardware que tengan los recursos necesarios y que pudieran brindar mejores resultados al seguir mejorando en costos y recursos de hardware.

2.5.1. Introducción

DAVID PEP significa Dispositivo de Audio y Video Interactivo Digital Para Educación Preescolar, y es un dispositivo que buscaría dar soporte como una herramienta en la que los educadores de nivel preescolar puedan reproducir contenido multimedia, pues permite desplegar video a través de un medio visual.

La idea básica para su operación consiste en que el educador pueda generar un contenido multimedia mediante una computadora y este contenido sea descargado a una tarjeta de memoria SD, misma que es introducida en DAVID PEP para que reproduzca el contenido generado.

Es importante señalar que DAVID PEP no reemplaza ningún otro medio tecnológico existente en las aulas y que tampoco es una consola de videojuegos. DAVID PEP se puede ver como una introducción sutil a las tecnologías de la información, brindando la reproducción de contenido multimedia interactivo, que ayude a incentivar a los alumnos a participar y a generarles interés por el aprendizaje de una forma que les puede resultar divertida. DAVID PEP puede utilizarse de muchas formas, pero la cuestión más básica es la de hacer que el contenido sea interactivo (pues puede utilizar como aditamento un dispositivo para detección de movimiento), aunque podría utilizarse para algunas tareas pasivas, por ejemplo para contar cuentos de manera digital desplegando una serie de gráficos que representen elementos dentro de una historia. Sin embargo, el propósito principal de DAVID PEP es el de crear un ambiente ubicuo [16] en donde el dispositivo no interfiera con la presentación del contenido en el aula.

2.5.2. Descripción de uso

Una manera simple de explicar la forma de uso es describiéndolo como un proceso donde se comienza en la computadora personal del educador, específicamente con un entorno de diseño de contenidos (con el cual DAVID PEP no cuenta) donde se genere el contenido y posteriormente se almacene en la tarjeta SD la cual se introduce en DAVID PEP, el cual lee el contenido y lo interpreta, para así generar y presentar el contenido en un monitor, TV o proyector. El contenido puede variar de acuerdo a los comandos recibidos por los dispositivos de entrada del DAVID PEP, los cuales pueden ser una pulsera inalámbrica, botones gigantes, un tapete de botones etc. La Figura 2.7 muestra como el dispositivo recibe información de las diferentes fuentes de entrada y presente un salida a un dispositivo de desplegado de imagen.

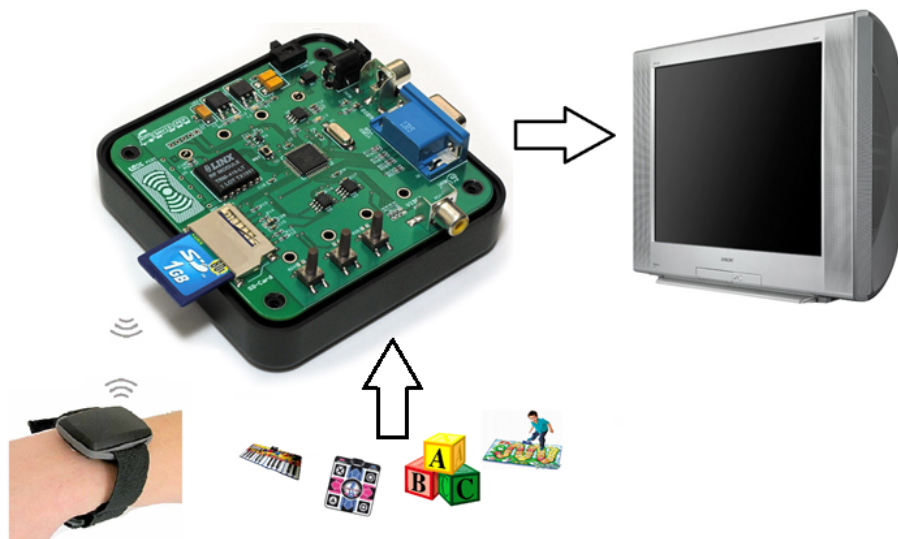


Figura 2.7 Conectividad de DAVID PEP

2.5.3. Arquitectura

DAVID PEP es un sistema embebido portátil, de bajo costo y bajo consumo de energía. En la Figura 2.8 se muestra un prototipo del dispositivo en su carcasa con la tarjeta principal

expuesta, pero los componentes de la tarjeta principal se muestra con mayor detalle en la Figura 2.9 donde puede observarse el microcontrolador como elemento principal, la memoria externa del tipo RAM y FLASH, un módulo de radiofrecuencia para la comunicación inalámbrica con los dispositivos de entrada, un conector para memoria SD, así como la salida de audio monoaural y las salidas de video VGA (Video Graphics Array) y video compuesto tipo NTSC (National Television System Committee).



Figura 2.8 Tarjeta principal de DAVID PEP

- *MICROCONTROLADOR MULTIPROCESADOR* – Debido a que la generación y manipulación de gráficos demanda alto poder de procesamiento, DAVID PEP hace uso como procesador principal un microcontrolador multiprocesador o multicore llamado Propeller de la compañía Parallax. La Figura 2.10 muestra un diagrama de la estructura interna del microcontrolador Propeller que cuenta con 8 unidades de ejecución, llamados “COGs”, y donde todos ellos pueden acceder a las terminales de entrada/salida de manera paralela. El acceso a la memoria HUB se realiza brindando en la modalidad de exclusión mutua del recurso, esto al estilo “round robin”, es decir, dándole el turno al cog 0 para utilizar la memoria después al cog 1, luego al cog 2 y así sucesivamente hasta el cog 7 y de nuevo empezar con el cog 0, luego el cog 1, y así sucesivamente.

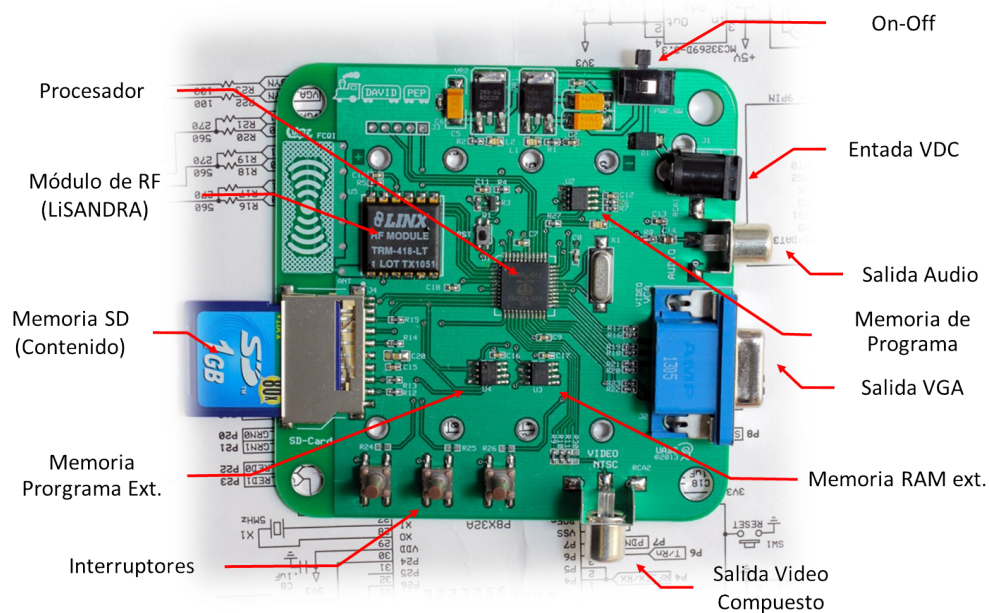


Figura 2.9 Tarjeta principal de DAVID PEP

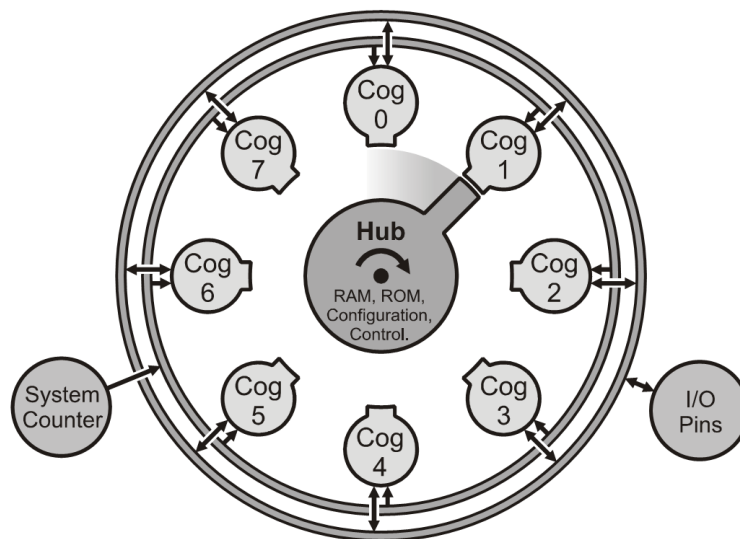


Figura 2.10 Arquitectura multicore del microcontrolador

- *PLATAFORMA LiSANDRA* – La tarjeta principal de DAVID PEP puede comunicarse inalámbricamente con los dispositivos de entrada y para ello hace uso de la plataforma LiSANDRA [17]. En la Figura 2.11 se muestra uno de los módulos que forma parte de

la plataforma LiSANDRA que cuenta con la comunicación inalámbrica, así como un acelerómetro de tres ejes y que es parte de una dispositivo tipo pulsera. Este módulo está programado con un algoritmo para detectar cierto contexto de la persona que porte la pulsera con el fin de transmitirlo como un evento a DAVID PEP para que sea procesado y se realicen los cambios en los gráficos de acuerdo a los programado previamente en el contenido que esté produciendo [18].



Figura 2.11 Pulsera con Nodo S de la Plataforma LiSANDRA

- *MÁQUINA DE GRÁFICOS DE DAVID PEP* – Una de las principales características de la máquina de gráficos de DAVID PEP, es la simplicidad algorítmica, esto con el fin de evitar requerir de alto poder de cómputo. Otra característica importante es el manejo de los gráficos mediante un esquema de multiplanos en la cual la imagen principal se conforma por la unión de planos superpuestos. La Figura 2.12 nos muestra una representación gráfica burda de como la imagen principal estaría formada por el conjunto de elementos en los planos posteriores. Las líneas rojas muestran la cantidad de operaciones que se realizan para obtener cada uno de los pixeles de la imagen principal,

en otras palabras, la cantidad de localidades de memoria, según la posición dada, se leen en las diferentes matrices que representan los planos, para obtener el dato o pixel. A pesar de que la Figura 2.12 es una representación burda, la idea principal es que de cada plano contenga ciertos elementos, por ejemplo, el plano que esta más al fondo podría contener el cielo, nubes y sol, un plano mas con montañas, un último plano árboles, y finalmente todos estos planos superpuestos forman una escena o imagen principal como se muestra en la Figura 2.13.

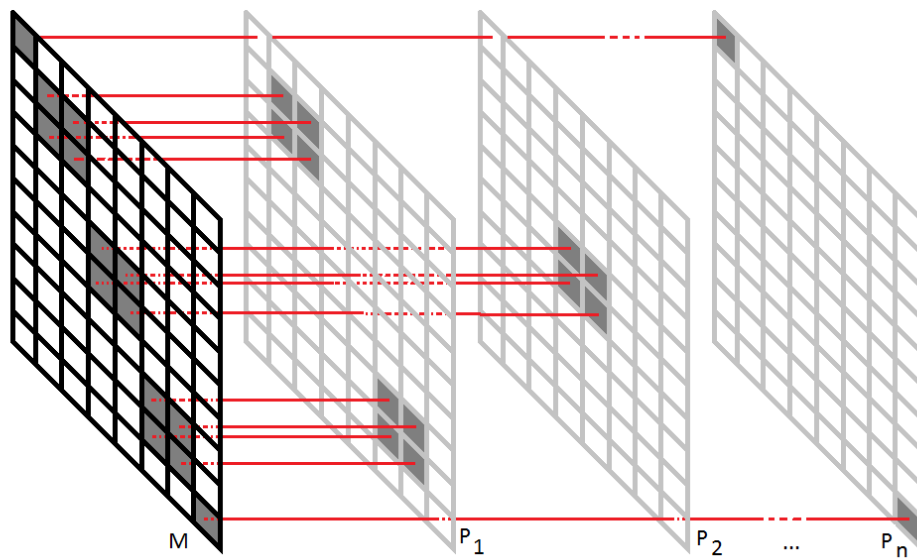


Figura 2.12 Manejo de multiplanos por DAVID PEP

Otra de las características de la máquina de gráficos de DAVID PEP es su limitada resolución de la imagen y cantidad de colores a usar debido a la capacidad gráfica con que se cuenta en el microcontrolador de Propeller, pero por su bajo costo y capacidad lo hacen ideoneo en cuanto costo beneficio para usarse en DAVID PEP. Sin embargo, algo a favor de esta limitante es que en la etapa preescolar, el uso de figuras o gráficos burdos ayudan a desarrollar ciertas habilidades importantes. Por ejemplo un objeto dibujado de manera muy burda con primitivas simples como líneas o círculos, con ciertos colores primarios, representa un reto cognitivo, ya que el cerebro necesita hacer

una comparación con los objetos que tiene registrados o que conoce y tiene que hacer una abstracción de los elementos más importantes que constituyen a ese objeto para poder compararlo con el que se está observando e identificarlo como un dibujo que representa a dicho objeto.

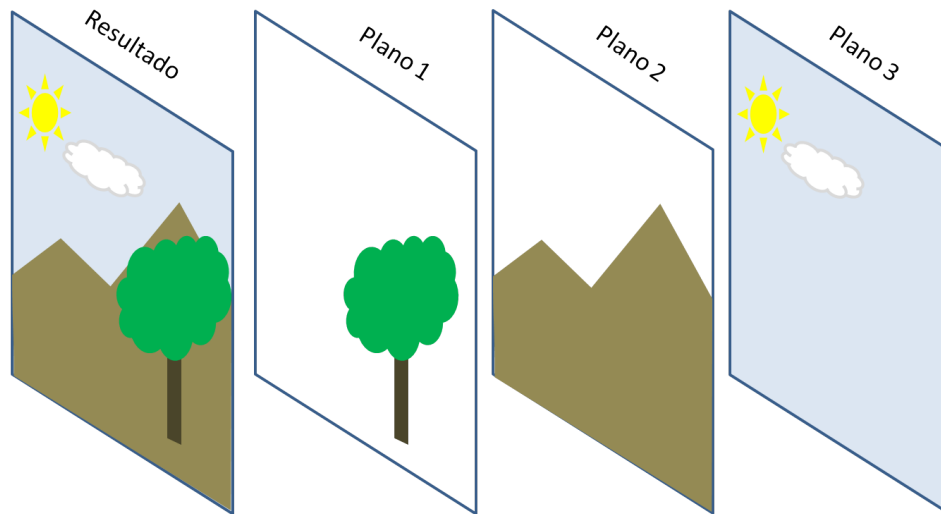


Figura 2.13 Imagen resultante de multiplanos superpuestos

2.6. Manejo del Contenido Multimedia

En la actualidad gran parte de los sistemas de información que se desarrollan buscan una mayor interoperabilidad entre plataformas [19], como en el caso de los servicios web que se pueden definir de una forma sencilla como un conjunto de tecnologías estándar de software para el intercambio de datos entre aplicaciones [20], buscando una mejor integración entre diferentes aplicaciones y lenguajes.

Aunque por los recursos limitados que actualmente maneja DAVID PEP es que no sería recomendable el completo uso de tecnologías que consuman una cantidad considerable de recursos, por lo que se busca un manejo de datos mas simple para la matriz de colores y

se busca la forma de combinar con serialización de datos en cuestión de eventos o casos específicos.

2.6.1. Bitmap

Es un formato de imagen que también se conoce como matriz de puntos, pues este se compone por puntos independientes (píxeles) que identifican uno por uno a la imagen [21]. Por lo que este formato o algo similar es buscado para el contenido gráfico, de forma que DAVID PEP pueda manejar de forma sencilla el dibujo del contenido.

2.6.2. Serialización de Datos

La serialización de datos es el proceso de convertir un objeto (o estructura de datos) en una secuencia de bytes para almacenar el objeto o transmitirlo a un archivo, base de datos o en memoria, guardando el estado del objeto para poder crearlo de nuevo cuando este se necesite, y al proceso inverso se le conoce como deserialización[22].

Entre algunos de los formatos de serialización mas utilizados se encuentran los siguientes, y se podrían utilizar uno ligero para el manejo de las acciones que indiquen al contenido multimedia que hacer, es decir, lo interactivo del contenido mediante las acciones físicas en el sistema embebido, y por el lado en la herramienta que crea el contenido multimedia, se pueden utilizar para representar el contenido multimedia completo (pues sería en una computadora con gran cantidad de recursos).

- *XML* – XML (Extensible Markup Language) es un estándar aceptado por la industria y que ha sido desarrollado por el World Wide Web Consortium. Este estándar provee un mecanismo para estructurar datos en jerarquía a través de marcas (tags) parecido como lo hace HTML, a diferencia de que XML es ajustable y extensible, además de

contar con un lenguaje de esquema llamado XSD (XML Schema) el cual describe la estructura y restricciones de los contenidos de los documentos XML [23].

Este estándar ha sido utilizado en ambientes de escritorio y de servidores desde 1998 pues además de ser flexible su sintaxis es amistosa (human-readable). Xml dispone de varios analizadores sintácticos (parsers), según sea lo mas conveniente, entre los cuales destacan SAX (Simple API for XML), que generalmente es utilizado para rápida lectura (por operar secuencialmente por cada sección, liberando memoria), así como DOM (Document Object Model) el cual opera y representa todo el contenido en forma de árbol (que puede ser actualizado en tiempo real) [24].

- *JSON* – JSON (JavaScript Object Notation) es un formato ligero para el intercambio de datos. Su lectura y escritura es amigable para sus usuarios, así como es sencilla su conversión y lectura para las computadoras. JSON es un formato de texto independiente del lenguaje aunque usa convenciones que pueden ser familiares a programadores de lenguajes tipo C [25].
- *YAML* – YAML (YAML Ain't Markup Language) es un lenguaje de serialización de datos basado en Unicode y diseñado para ser amistoso. para todos los lenguajes de programación. YAML puede ser visto como un superconjunto de JSON, que ofrece mejor legibilidad humana y un modelo de información más completa, por lo que cualquier archivo JSON es valido para YAML. Aunque YAML y JSON pretenden ser formatos legibles (human readable) de intercambio de datos, éstos tienen distintas metas, JSON hace a sus datos mas sencillos de procesar por los lenguajes de programación reduciendo la legibilidad a los desarrolladores, mientras que YAML mejora la legibilidad (human readable) aunque es mas complejo su procesamiento [26].
- *BSON* – BSON (Binary JSON) es un formato binario de serialización de documentos tipo JSON. BSON contiene extensiones que no son parte de la especificación de JSON

como las fechas y tipos de datos binarios. BSON se puede comparar con otros formatos binarios como Protocol Buffers, aunque BSON no cuenta con esquema [27].

- *MessagePack*– MessagePack es un formato binario eficiente de serialización de datos. Permite el intercambio de datos de lenguajes como JSON, aunque mas rápido y de menor tamaño [28].

La Tabla 2.1 presenta una comparativa entre los lenguajes de serialización de datos.

Tabla 2.1 Formatos de Serialización de Datos

Nombre	Estandarizada	Legible	Fácil de Usar	Binario
XML	Si	Si	Si	No
JSON	Si	Si	Si	No
YAML	Si	Si	Si	No
BSON	Si	No	Si	Si
MessagePack	Si	No	Si	Si

Estos son lenguajes son utilizados para representar datos similares:

- *DrawingML* – Es un formato de archivo para gráficos en dos dimensiones que utiliza un lenguaje de marcas XML que contiene un extenso conjunto de especificaciones para diferentes componentes visuales en los documentos Office que permiten definir objetos gráficos como imágenes, figuras, gráficas y diagramas en OOXML (Office Open XML). DrawingML no es un lenguaje independiente, sino que apoya y aparece dentro de documentos WordprocessingML, SpreadsheetML y PresentationML[29].

- *SVG* – SVG (Scalable Vector Graphics) es un lenguaje que describe gráficos vectoriales en dos dimensiones utilizando XML [30].
- *SMIL* – SMIL (Synchronized Multimedia Integration Language) es un estándar de la W3C que permite integrar contenido multimedia (audio, video, imágenes, texto). SMIL es un lenguaje fácil de aprender, donde a través de XML se permite definir comportamiento a los elementos. [31]

Por lo que en este proyecto se busca que los elementos a animar e interactuar se pueda complementar su definición en alguna estructura como las anteriores o similar, donde describan los datos que permitan a DAVID PEP realizar alguna acción determinada.

2.6.3. Formato Contenedor de Multimedia

Un formato contenedor es un tipo de archivo que describe como se almacena información, estos son comúnmente usados en aplicaciones multimedia. Este archivo puede identificar diferentes tipos de datos, algunos pueden soportar formatos de audio, video, subtítulos, entre otros, ejemplos de estos son AVI, MPEG, MP4 [32]. Aunque en este trabajo no se esta buscando un formato de contenido, si se busca una forma para manejar el contenido multimedia, que pueden ser los archivos que contengan sprites, figuras, audio, o algún tipo de multimedia.

Capítulo 3

Plataforma de Contenido Multimedia

En este capítulo se plantea la contribución a una plataforma que a través de un entorno permita diseñar contenido multimedia y que dicho contenido pueda ser desplegado en un medio visual así como poder ser interactuado por eventos externos predefinidos, a través de un dispositivo de hardware, y se describe la propuesta del entorno de diseño de contenido multimedia. El dispositivo de hardware es quien tiene la capacidad de reproducir el contenido y enviarlo a medio visual, así como de recibir eventos de un dispositivo tipo pulsera, con capacidad de sensar movimientos.

3.1. Introducción

Como continuación a los trabajos de tesis en los que se desarrollaron DAVID PEP y el Dispositivo de Detección de Movimientos se plantea la contribución a esta plataforma con el entorno de diseño que permite crear contenido multimedia mediante una aplicación de software donde dicho contenido pueda ser utilizado en el sistema embebido de DAVID PEP, y que a través de DAVID PEP pueda ser controlado mediante eventos físicos a través de un

dispositivo de detección de movimientos.

La idea de la plataforma es proporcionar al educador, el cual es aquella persona que tiene el conocimiento sobre los temas o habilidades a desarrollar así como la experiencia para enseñar y sobretodo que tiende a conocer cuales son las necesidades de sus alumnos, una entorno de diseño que le permita crear contenido multimedia y que con el apoyo de un sistema embeido, pueda ser desplegado en un medio visual y recibir instrucciones a través de eventos externos predefinidos que sean efectuados mediante el alumno o el educador como se muestra en la Figura 3.1.

En cuanto a la interacción de los eventos externos, se busca que el educador los pueda definir en el entorno en el momento de diseñar el contenido multimedia, los cuales sean eventos predefinidos que el sistema embebido pueda reconocer.

De forma que el desarrollo e integración a dicha plataforma pueda apoyar en la educación preescolar, buscando beneficiar en el proceso de enseñanza y aprendizaje.

En la Figura 3.1 en la parte izquierda se muestra la parte de la tecnología que requiere esta plataforma, y se divide en la parte de software y de hardware.

La plataforma se compone de un software que permite al educador diseñar contenido multimedia de una forma sencilla en donde tenga la posibilidad de agregar animaciones y sprites, así como eventos que hagan posible animar o realizar alguna acción en el contenido, según un evento externo predefinido. De igual forma se busca que el software (entorno de diseño multimedia) permita exportar el contenido en un formato que pueda ser reproducido en sistemas embebidos como DAVID PEP (el cual es un sistema embebido de recursos limitados) o algún otro que proporcione los requisitos necesarios.

En cuestión del manejo de contenido, se busca que sea una estructura sencilla para el educador y los educandos, algo asemejable, por lo que se ha llegado a que sea basado en un teatro, de manera que se puedan agregar escenarios, escenografías, elementos y actores que

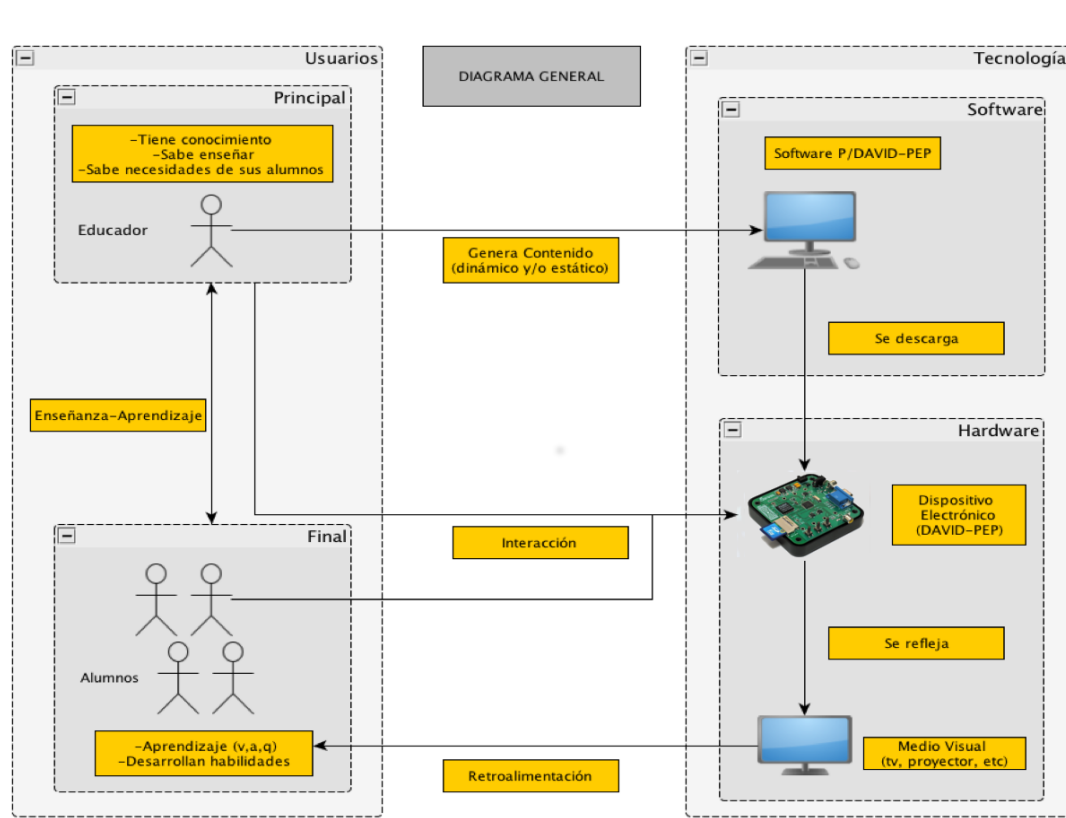


Figura 3.1 Diagrama General.

puedan seguir un guión.

Respecto al hardware, se encuentra la parte del sistema embebido, el cual debe tener la capacidad de desplegar video y ser programable, de forma que se le adapte la capacidad de reconocer la estructura y formato de datos, y por parte de la detección de eventos, se busca un dispositivo que pueda sensar movimientos y enviar los datos hacia el sistema embebido.

De forma que el dispositivo de detección de eventos, permita al usuario realizar una acción, de forma que el dispositivo lo detecte y envíe una acción al sistema embebido, el cual realice según el contenido multimedia, la acción correspondiente, es decir, seguir el guión del actor, según lo creado en el entorno de diseño.

3.2. Requisitos Generales

Por lo que en cuanto a la tecnología para esta propuesta se busca por parte del hardware que procesa el contenido que:

- Permita desplegar contenido multimedia a través de medios visuales.
- Permita implementar módulos de programación.
- Permita leer archivos.
- Permita recibir señales de otro dispositivo que pueda detectar eventos externos.
- Permita recibir señales del dispositivo que detecte eventos externos.

En cuanto al dispositivo de detección de movimientos se pretende que:

- Permita detectar eventos físicos predeterminados.
- Permita enviar datos al dispositivo que procesa el contenido.

Y por la parte del software se pretende que:

- Permita crear presentaciones.
- Permita agregar diapositivas.
- Permita manejar multiples capas por diapositivas.
- Permita diseñar elementos (figuras).
- Permita crear sprites.
- Permita animar elementos.

- Permita importar contenido.
- Permita exportar contenido.

De tal forma que queden las partes que conforman la plataforma con las capacidades necesarias para poder ser integradas.

3.3. Actualidad

En cuanto al hardware, para la detección de eventos, se cuenta ya con un dispositivo electrónico que permite detección de movimientos. Este dispositivo se utiliza en forma de pulsera lo que permite su fácil uso.

En cuanto al dispositivo embebido programable que permita desplegar video, también se cuenta con uno que cubre las expectativas necesarias (que puede ser otro dispositivo embebido, siempre y cuando cuente con los requisitos y programación necesaria) y que además tiene la capacidad de recibir datos del dispositivo de detección de movimientos, pues este dispositivo embebido lo utiliza como un aditamento, y como se mencionaba anteriormente, ambos fueron desarrollados en el grupo de trabajo y fueron conectados entre si.

En cuanto al software, aunque si existen distintos entornos para diseñar contenido multimedia, no se ha encontrado uno de código abierto que permita los elementos buscados para este entorno. Por lo que se ha optado por desarrollar uno a la medida en el cual se le pueda seguir agregando funcionalidad y mejoras según las evaluaciones que en un futuro se vayan realizando.

Por lo que también es necesario una forma de comunicación entre el software de computadora y el sistema embebido, así como agregar la programación necesaria al sistema embebido para reconocer el contenido y saber como reaccionar mediante los eventos externos.

3.4. Alcances

Los alcances principales en este trabajo son proponer y desarrollar el entorno de diseño multimedia (el software para educadores) que se enfoca en el diseño de contenido multimedia, así como brindar una versión prototipo que permita llevar a una validación pedagógica y a la continuación del desarrollo de la plataforma. Por lo que se analiza, diseña, desarrolla y prueba la versión prototipo del entorno de diseño multimedia, así como pruebas generales en la comunicación con el hardware DAVID PEP.

Capítulo 4

Desarrollo del Entorno de Contenido Multimedia

En este capítulo se plantea el desarrollo del trabajo, así como las tecnologías utilizadas y la metodología que se ha seguido.

Como parte de la plataforma se toma como base el sistema embebido DAVID PEP, el cual cuenta con un aditamento integrable que es un dispositivo electrónico de detección de movimientos, por tener los requisitos necesarios para la plataforma planteada.

Por lo que en este capítulo se centra en la parte que corresponde al entorno de diseño de contenido multimedia, que crea, maneja y transforma el contenido multimedia para que sea procesado en DAVID PEP, teniendo así las partes que conforman el entorno.

4.1. Diseño de Solución

Teniendo en cuenta los recursos de hardware, para completar la plataforma, hace falta de un entorno que diseñe el contenido multimedia, en este caso, se habla de un software con base

en los requisitos y que considere a sus usuarios (educador y alumno), además de que aplique la estructura que permite manejar el contenido y el formato que hace al contenido interoperable con DAVID PEP, es decir, de poder transformar el contenido creado en el software de alto nivel a una forma en la que sea comprensible y manejable en DAVID PEP (por sus recursos limitados), para que el dispositivo que detecta movimientos pueda ejercer una acción en el contenido a través de DAVID PEP. Para esto es necesario de métodos que permitan a DAVID PEP manejar el contenido de una forma eficiente así como implementación en trabajo futuro para la completa integración con el dispositivo que detecta movimientos.

En la Figura 4.1 se muestran todas las partes que componen la plataforma. Se encuentra el entorno de diseño de contenido multimedia el cual lo comparte con el dispositivo DAVID PEP para que pueda ser el contenido desplegado en un medio visual y pueda recibir acciones mediante el dispositivo que detecta movimientos.

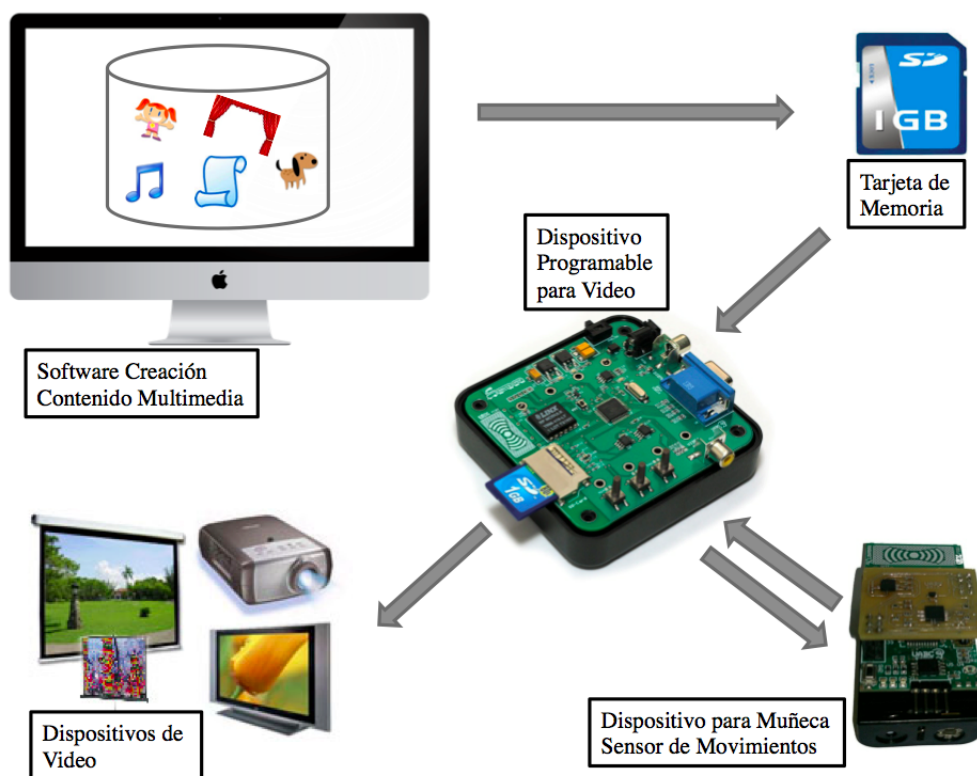


Figura 4.1 Visión del Entorno Propuesto.

Se ha buscado que el proceso de visualización, manejo del contenido y de eventos, sea ejecutado en este dispositivo embebido, por el bajo costo que representan en comparación de una computadora y principalmente para que los dispositivos sean ubicuos en el area de clases buscando minimizar la distracción al formar parte del ambiente de aprendizaje.

4.2. Estructura del Contenido

Para manejar la información del contenido se busca una forma sencilla y ligera de poder expresarlo, por lo que se utiliza una estructura basada en un escenario de teatro, de forma de poder tener un escenario el cual maneje escenografía, objetos y actores, y que pueda ser mapeado hacia elementos que componen a al software de presentación.

En la Figura 4.2 se muestra la estructura que representa los datos del contenido. Para esta estructura se busca la semejanza a los elementos básicos de un teatro, por lo que se cuenta con un elemento principal de tipo escenario. Este elemento es representado mediante un “Slide” y puede haber multiples escenarios. Cada escenario puede también tener sus eventos, los cuales al ser accionados representan una función en el escenario. Cada escenario cuenta con su respectiva escenografía, la cual puede ser dividida en 4 capas, esto para poder representar profundidad y es representado por los elementos de tipo “Layer”. Cada elemento de tipo “Layer” contiene los “Elements” que representan a los actores y objetos, los cuales pueden ser animados y reaccionar a eventos. Estos actores y objetos (elementos) pueden ser representados por “Element” que son elementos simples, que podrían ser como una figura geométrica. Los elementos de tipo “Composite” se proponen a figuras compuestas. Y el último tipo de elemento es el “Sprite” el cual se diseña con varios actores secuenciales (en base a los elementos de dibujo), y estos actores van cambiando en el tiempo según la secuencia de dibujo. Cada Sprite tiene la posibilidad de ser animado, esto es mediante el elemento “FrameAnimation” el cual indica las propiedades de movimiento (de que punto, hacia que

punto y a que velocidad)(los términos para representar la estructura se han escrito en ingles para coincidir con el idioma de la implementación).

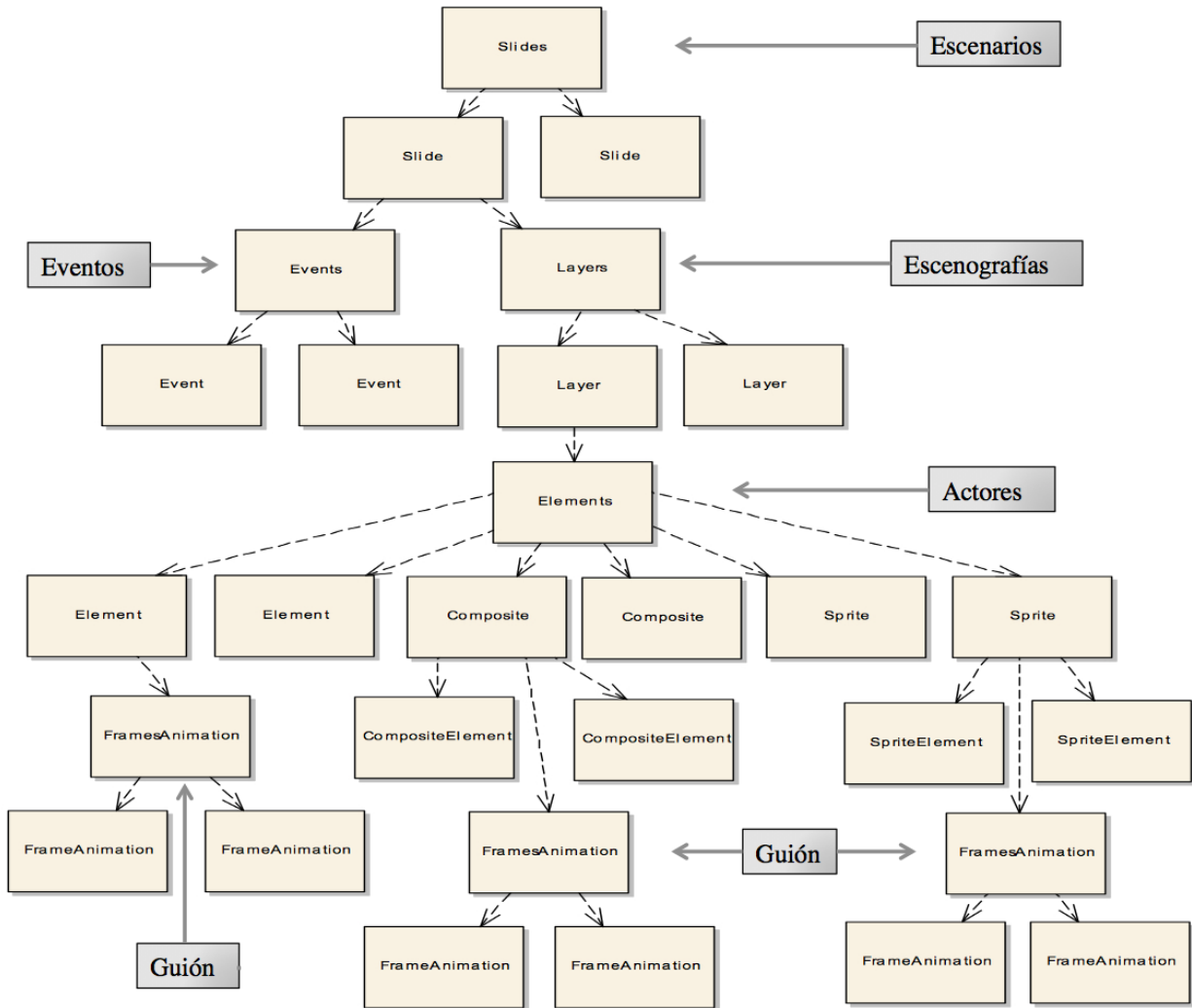


Figura 4.2 Estructura del Contenido.

De forma que se busca que la estructura sea viable si para cada evento que le pertenezca a un escenario existe por lo menos un actor en una escenografía dentro del escenario al cual se le ha asignado un guión. Esto pues al ser detectado un evento es porque ha sido necesario una animación a la cual el actor deba reaccionar, y busca ser representado bajo la siguiente condición de viabilidad.

Donde:

e = Escenario, s = Escenografía, a = Actor, v = Evento, g = Guión.

Sea : $(\forall v)(v \subset e \implies (\exists a)(a \subset s \subset e)(g \mid g \subset a))$

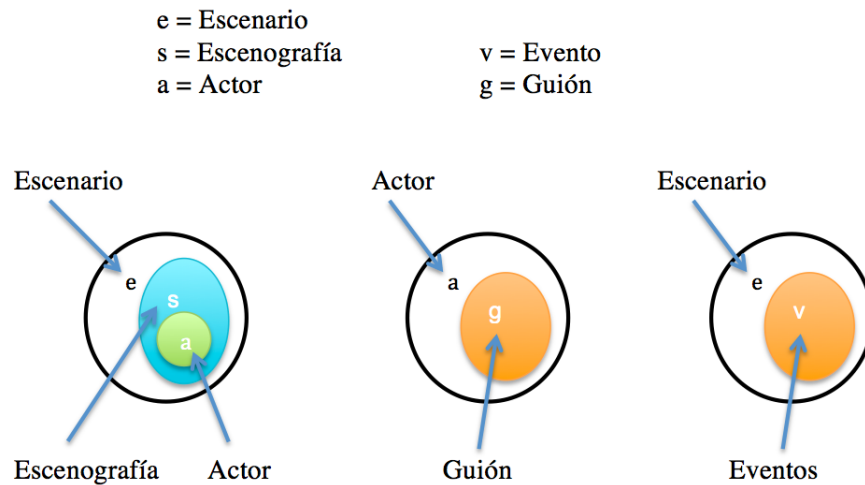


Figura 4.3 Estructura.

4.3. Entorno de Diseño de Contenido

El entorno de diseño de contenido multimedia se representa mediante un software, el cual es el creador y manejador de contenido. Es el que permite al educador crear contenido multimedia como son figuras básicas, sprites y animaciones sencillas en las cuales se puedan agregar eventos predefinidos, los cuales pueden ser accionados desde los eventos externos, una vez que el contenido sea procesado en el sistema embebido y tenga la funcionalidad requerida para la interacción con el contenido al detectar un evento externo.

Esta aplicación busca ser estándar en cuanto a un software de presentación, es decir, contar con dispositivos, contar con una forma de crear figuras, así como también cuidar el enfoque propuesto para los usuarios (educador), de manejar el contenido sencillo y amigable.

Al basarnos sobre la plataforma DAVID PEP hemos visto que la forma en que maneja

las figuras geométricas permite que sean desplegadas en cuatro planos, lo cual puede ser vista como profundidad, por lo que para la creación del contenido multimedia en el software, será enfocada en 4 planos, dando mejor integración en el momento de crear el contenido y pasarlo a la plataforma, aunque, se busca que esto pueda ser configurable en la aplicación de software por cambios futuros que puedan suceder.

El manejo de planos que se ha planteado se muestra en la Figura 4.4, por lo que es posible usar cada uno para representar profundidad.

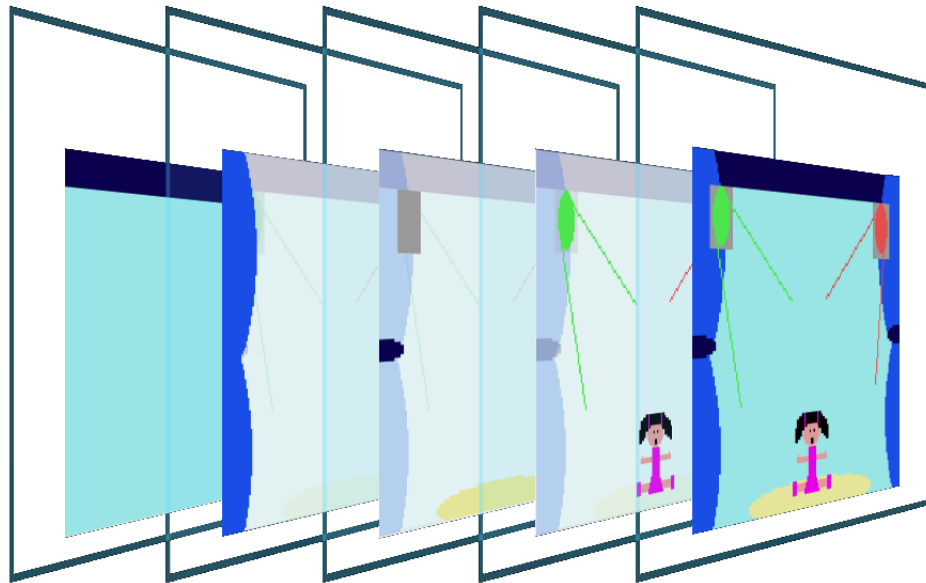


Figura 4.4 Plano Resultante.

4.3.1. Requisitos

Se define a un requisito como una condición o capacidad necesaria por un usuario para solucionar un problema o lograr un objetivo [33].

En base al objetivo principal que busca facilitar una forma para que el educador pueda diseñar un contenido multimedia basado en alguna idea o tema y que el contenido pueda

ser controlado desde un dispositivo electrónico que permita desplegarlo así como ser accionado desde eventos externos, es que se plantean los siguientes requisitos funcionales y no funcionales.

- **Requisitos Funcionales** Los requisitos funcionales son aquellos que especifican una funcionalidad que un sistema o componente del sistema debe ser capaz de realizar [33].

En la Figura 4.5 se listan los principales requisitos a implementar.

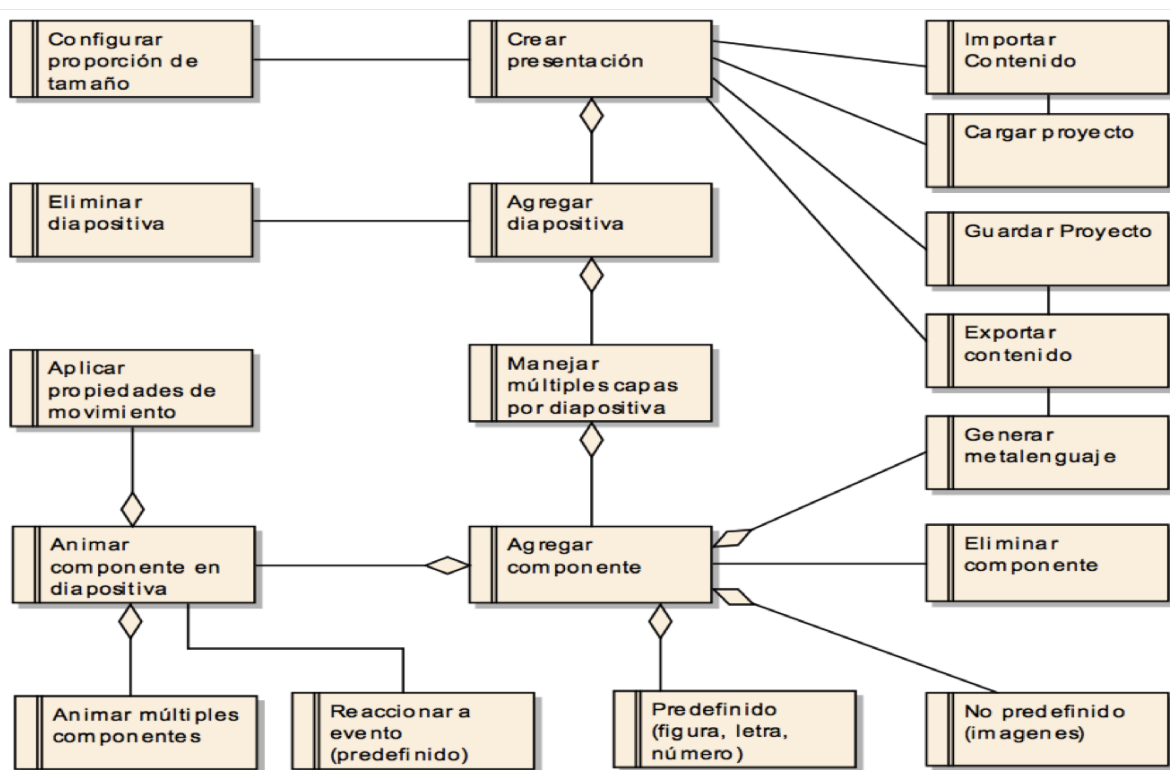


Figura 4.5 Requisitos Funcionales.

- **Requisitos No Funcionales**– Los requisitos no funcionales también son conocidos como cualidades de un sistema, y en otros términos son “restricciones”, “atributos de calidad”, “objetivos de calidad”, “calidad de servicio” [34], y especifican como debe ser la operación del sistema, de modo diferente a los requisitos funcionales, los cuales especifican una funcionalidad que el sistema o componente del sistema debe ser capaz

de realizar.

Para este desarrollo se han seleccionado los siguientes requisitos no funcionales.

- **Usabilidad** La Usabilidad se conoce como un atributo de la calidad que mide que tan fácil son de usar las interfaces de usuario, así como también se les conoce a métodos para mejorar la facilidad de uso durante el proceso del diseño y se le define por cinco componentes de la calidad, los cuales son facilidad de aprendizaje, eficiencia, memorabilidad, errores y satisfacción [35].

Para el diseño de este software se tomaron tres componentes como prioridad.

–El primero de ellos es la Facilidad de Aprendizaje, pues se busca que la aplicación sea sencilla de usar, de modo que las tareas básicas se puedan aprender en unos cuantos pasos. Esto en parte se logra a un diseño amigable que maneja cierta similitud que se encuentran en otras aplicaciones de presentaciones y dibujo.

–En el segundo componente se toma la Eficiencia, que se refiere a que tan rápido se puede realizar una tarea una vez que se ha aprendido el diseño, lo cual es necesario en esta aplicación, para minimizar los tiempos que el educador invierta creando su contenido. Para esto se crea un módulo que permite guardar figuras creadas, con el fin de reutilizarlas en diferentes ocasiones.

–Y el tercero que se toma es la Memorabilidad el cual busca la facilidad para que los usuarios que han utilizado la aplicación después de un período de no usarlo, adquieran de nuevo los conocimientos que ya tenían. En este caso puede ser posible que entre diferentes prácticas y dinámicas que manejan los educadores o en periodos vacacionales de los niños, no se utilice esta aplicación, por lo que se ha manejado una aplicación sencilla en que la interfaz guíe al educador cuando vuelva a utilizarla, a través de la división de la interfaz, que consiste en las diapositivas,

las figuras a seleccionar, los colores, y un plano central para dibujar, tal como lo haría si no fuera una aplicación, es decir, si fuera un dibujo físico, en donde se selecciona la hoja, el color y la figura a dibujar.

- **Interoperabilidad** La interoperabilidad se define como la habilidad de que dos o mas componentes o sistemas intercambien información y utilicen la información intercambiada [33].

Por lo que este fue un atributo necesario, pues se buscaba la forma de que el sistema embebido con recursos limitados utilizara la información brindada por el entorno de diseño de contenido, la cual viene desde un sistema y aplicación de alto nivel.

Para esto se ha buscado y adaptado un mapa de bits que permite enviar el contenido al sistema embebido y que permita interactuar con dicha información.

- **Rendimiento** Éste se define como el grado en que un sistema o componente realiza sus funciones designadas dentro de las limitaciones dadas, como la velocidad, la precisión, o el uso de memoria [33], y este atributo fue importante hacia la estructura y formato del contenido que crea el software, pues siendo sencillos y eficientes se obtiene mejor rendimiento en el sistema embebido.
- **Portabilidad** La portabilidad es la facilidad con la que un sistema o componente puede ser transferido de un hardware o de un software a otro [33].

Para lograr la portabilidad en el contenido, se ha utilizando una estructura de datos basada en un escenario y en atributos basados en los que maneja el lenguaje de SMIL, de tal forma que se represente todo el contenido en esta estructura, la cual puede ser interpretada por el mismo sistema, por otra implementación del entorno (por ejemplo en tableta), en un sistema embebido(con mayor cantidad de recursos), o cualquier sistema con las capacidades y con la adaptación de la programación necesaria para leer esta estructura y su formato de datos. De igual forma como los

recursos de memoria DAVID PEP son limitados, es que se realizan adaptaciones para el manejo, incluyendo datos a través de archivos tipo bitmap y el contenido mediante un formato de datos que pueda contener solamente instrucciones para las acciones de los eventos.

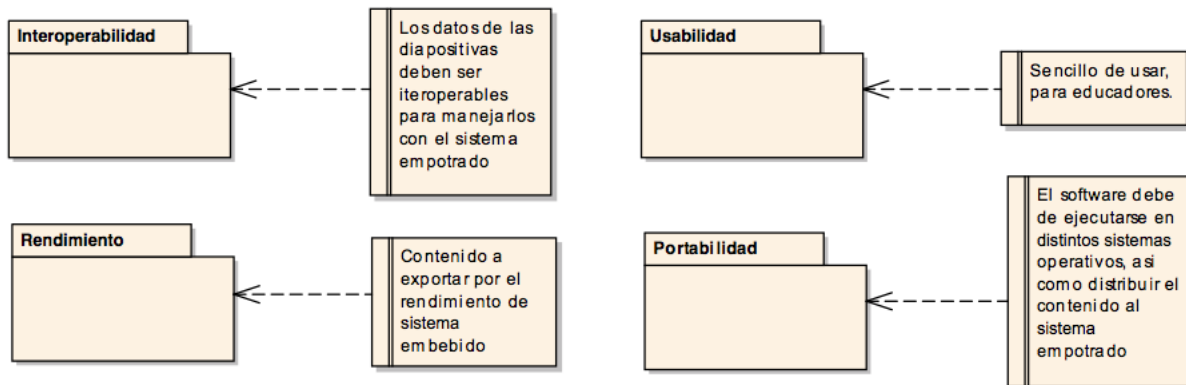


Figura 4.6 Requisitos No Funcionales.

4.3.2. Calidad del Software

En [33] se define como el grado en que un sistema, componente o proceso cumple con los requisitos especificados y con las necesidades o expectativas del cliente o usuario.

También se menciona en [34] que en una percepción más común de la calidad del software, se le suele reconocer como ausencia de fallos en el software, ya que el producto con fallos, difícilmente cubriría los requisitos y a estos comúnmente se expresan como ratio de defectos (numero de defectos en algún área) y en fiabilidad (numero de fallos en cierto tiempo de funcionamiento, o la probabilidad de no tenerlos en cierto tiempo).

En lo que para este desarrollo se ha enfocado en cumplir los requisitos para una versión prototipo en funcionamiento básico, minimizando a través de pruebas de funcionalidad la cantidad de defectos y maximizando la fiabilidad, buscando en el diseño y la implementación

mejoras en la usabilidad(que tan fácil son de usar las interfaces de usuario), mantenibilidad(esfuerzo para localizar y arreglar un error) y adaptabilidad(que tan fácil el programa puede ser modificado para adecuarse a distintas limitaciones).

En cuestión de métricas de calidad, se han pensado en aplicar las métricas de mantenimiento, esto cuando el producto se encuentre en una versión mas estable y sea probado por mas usuarios, de forma que se corrijan posibles errores detectados en la brevedad posible, así como aplicar métricas de satisfacción del usuario.

4.3.3. Metodología

Una vez iniciado el desarrollo de software se optó por tomar como base el modelo de prototipo y posteriormente se siguió con bases de la metodología scrum, de forma que fuera permitiendo facilidad en el cambio de requisitos y en fácil adaptación, esto en parte por ser una propuesta de innovación que en el transcurso se ha seguido madurando.

La metodología de scrum es un framework iterativo e incremental para proyectos y productos o desarrollo de aplicaciones que estructura el desarrollo en ciclos de trabajo no mayores a un mes que son llamados sprints y que son secuenciales y sin pausa. Al inicio del sprint el equipo selecciona los elementos (requisitos) de una lista priorizada y el sprint debe finalizar en la fecha especificada. El equipo debe completar los elementos del sprint, y durante este no deben cambiar los elementos elegidos. Al final del sprint se muestra el desarrollo. Los principios ágiles se enfocan en generación de software que funcione y pueda ser utilizado rápidamente, en vez de enfocar gran cantidad de tiempo en la escritura de documentación [36]. Una de las ventajas de Scrum, es que debido a las entregas periódicas con funcionalidad, se podía conocer el estado del proyecto y reafirmar o modificar los requisitos [37].

El modelo de prototipo busca que el sistema o componente del sistema se construya

rápidamente para entender con mas facilidad lo que se busca minimizando el riesgo del desarrollo, y este modelo es común que se utilice cuando se conoce la idea del producto pero aún falta delimitar y detallar los requisitos.

En la primera etapa del desarrollo se utilizó el modelo de prototipo utilizando el modelo con características esenciales, el cual permite evaluar algunos aspectos del diseño y características muy elementales, y en una segunda etapa seguimos en la versión actual utilizando características de la metodología scrum.

4.3.4. Arquitectura y Diseño de Software

La arquitectura de software de un programa o sistema informático, es la estructura o estructuras del sistema, que incluyen elementos de software, las propiedades externamente visibles de esos elementos, y las relaciones entre ellos [38].

Por lo que teniendo teniendo la arquitectura se puede visualizar la forma en la que la aplicación se debe comportar desde el principio, de forma que en base a la propuesta y principalmente a los requisitos se modeló en el inicio el flujo para el desarrollo, el cual se muestra en la Figura 4.7.

De igual forma se conformó un diseño inicial para el prototipo, el cual se mejoró y se implementó en el desarrollo actual, y que en continuación de las iteraciones se logró llegar al modelo de clases actual, el cual que se observa en la Figura 4.8.

Se ha buscado que el agregar elementos de tipo figura sea de forma sencilla a través de la clase “ElementFigure” la cual brinda el comportamiento principal de la aplicación o si fuera de cualquier otro tipo se heredaría a partir de la clase “Element”. La interacción de la aplicación se da a través de la clase “Controller” y se llama a un archivo de tipo vista la cual desliga el contenido. Los pixeles que se manejan son simulados en la clase “Pixel”, para

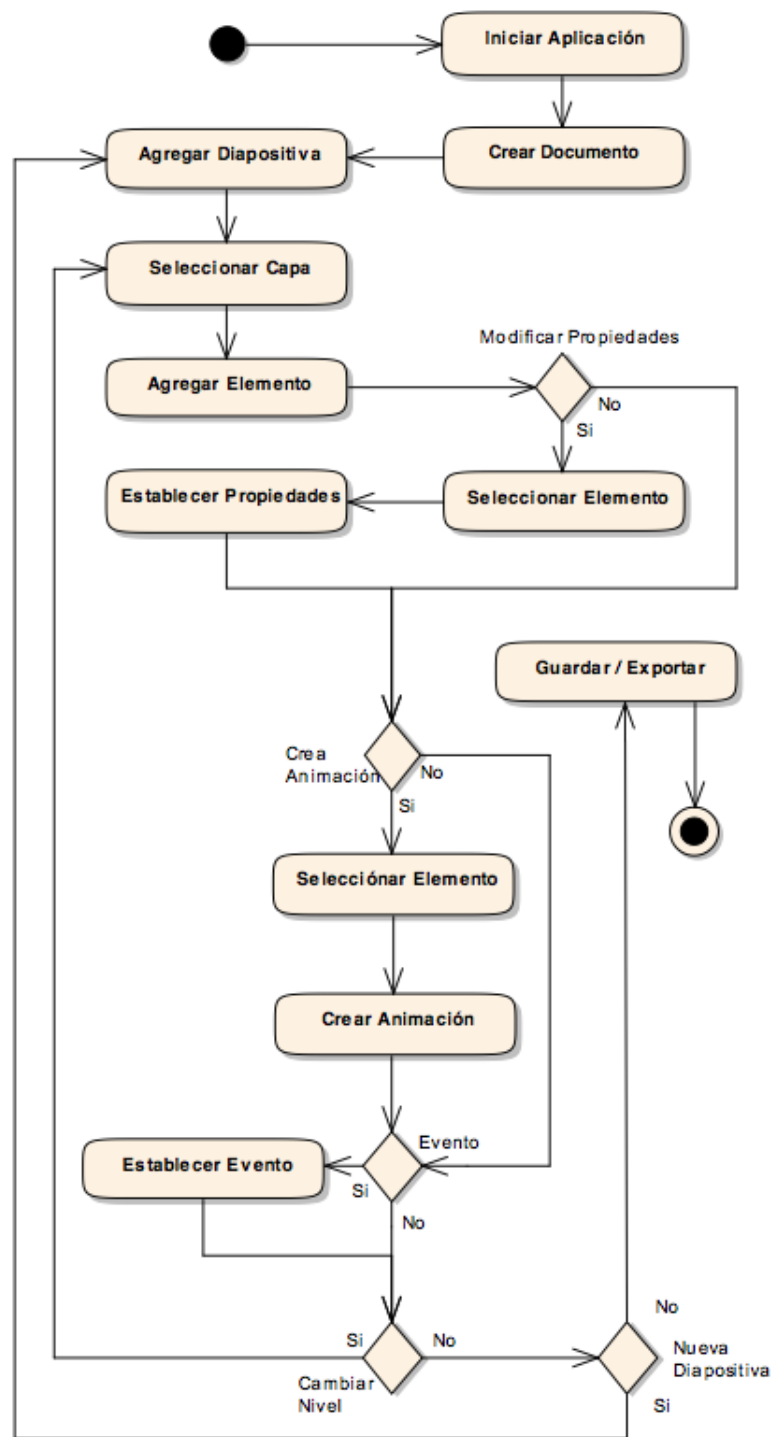


Figura 4.7 Flujo de la Aplicación.

acercar el diseño y resolución a la del dispositivo embebido, actualmente ésta resolución de pixeles simulada si puede ser configurable, pero a través de código. Para gráficos se extiende de clases de java como son “Pane” para los paneles y “Canvas” para dibujo. En cuanto a las figuras se implementaron algoritmos propios que pudieran utilizar a la clase “Pixel” controlando los datos exactos de pintado, que pueden ser enviados como mapa de bits al sistema embebido.

Para el manejo de la arquitectura se utilizó el patrón de MVC.

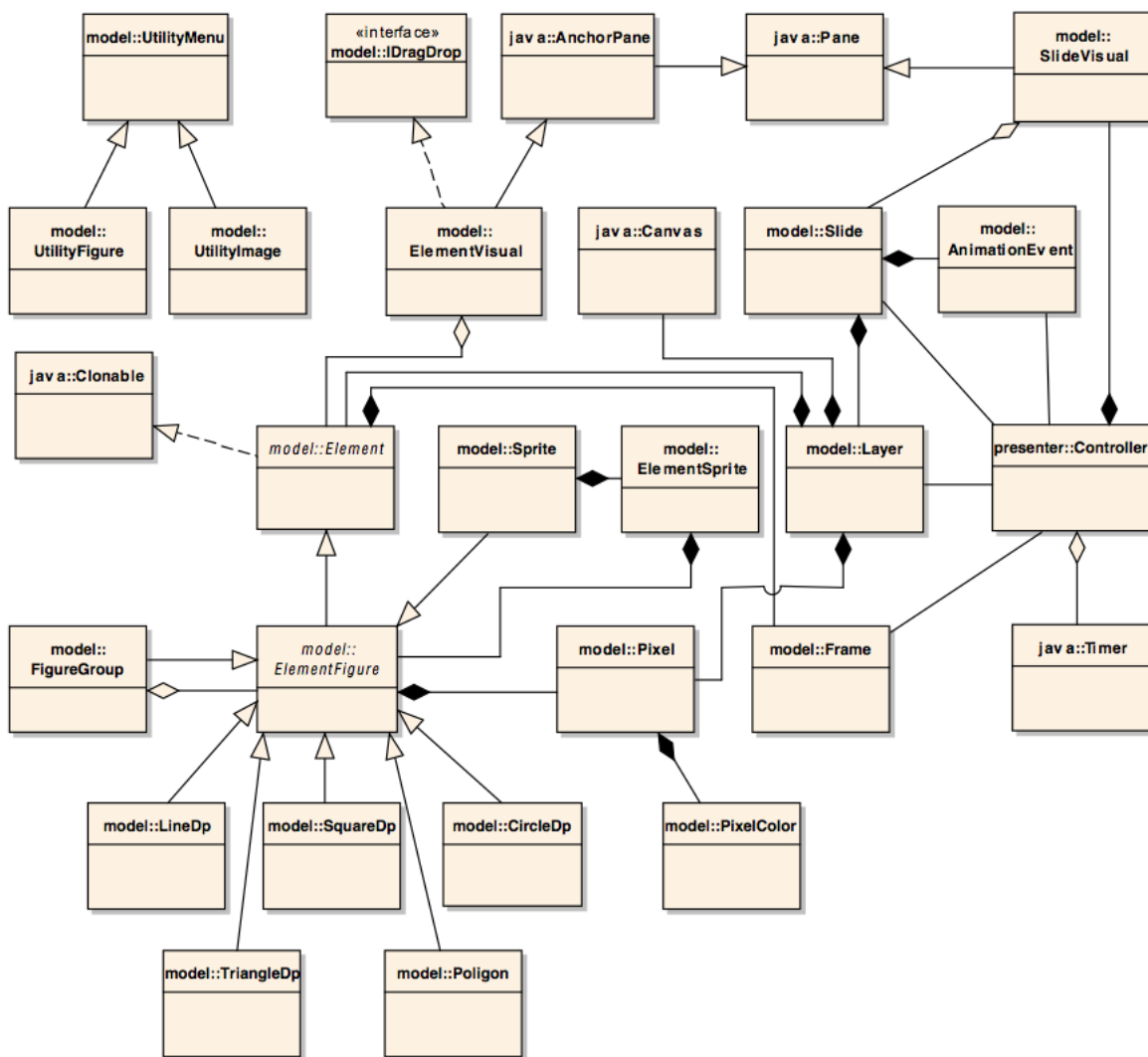


Figura 4.8 Modelo de Clases.

4.3.5. Patrones de Software

Los patrones de diseño son estructuras de las soluciones probadas y documentadas a problemas de desarrollo de software que están sujetos a contextos similares y existen patrones creacionales, estructurales y de comportamiento [39] . Los principales patrones que se aplican en este software son :

- **Observador** Este patrón notifica cambios en los objetos. En esta aplicación se utiliza en las animaciones de los elementos, ya que el que pinta es un solo thread, por lo tanto cada objeto puede en su respectivo thread cambiar alguna propiedad de posición, y a través del observador notifican su cambio, y el que dibuja ya lo tiene en cuenta para desplegarlo.
- **Fábrica Simple** Crea nuevas instancias de objetos. Este patrón se utiliza en la creación de figuras.
- **Composite** Este patrón sirve para construir objetos complejos a partir de objetos simples, y este proyecto este patrón es aplicable al crear un sprite (cada elemento del sprite se puede componer de varios elementos) o al crear una figura compuesta.
- **MVC** Este patrón separa en tres capas: modelo, vista y controlador. El modelo que representa el dominio de datos, el controlador que define el comportamiento de la interacción del usuario, y la vista que muestra la interfaz gráfica de usuario. Este patrón se utiliza en la estructura de la aplicación.

4.3.6. Diseño de Interfaz Gráfica de Usuario

El diseño de la GUI (Graphics User Interface) se ha manejado de una forma sencilla, solo con el objetivo de facilitar al usuario la interacción con la funcionalidad de la aplicación,

aplicando algunos de los puntos de las heurísticas de Nielsen. Convendría aplicar de forma completa la usabilidad así como el diseño gráfico y que sería aún mejor con la aplicación de conceptos sobre UX (experiencia de usuario).

En la Figura 4.9 se muestra el layout básico del entorno de diseño de contenido multimedia.

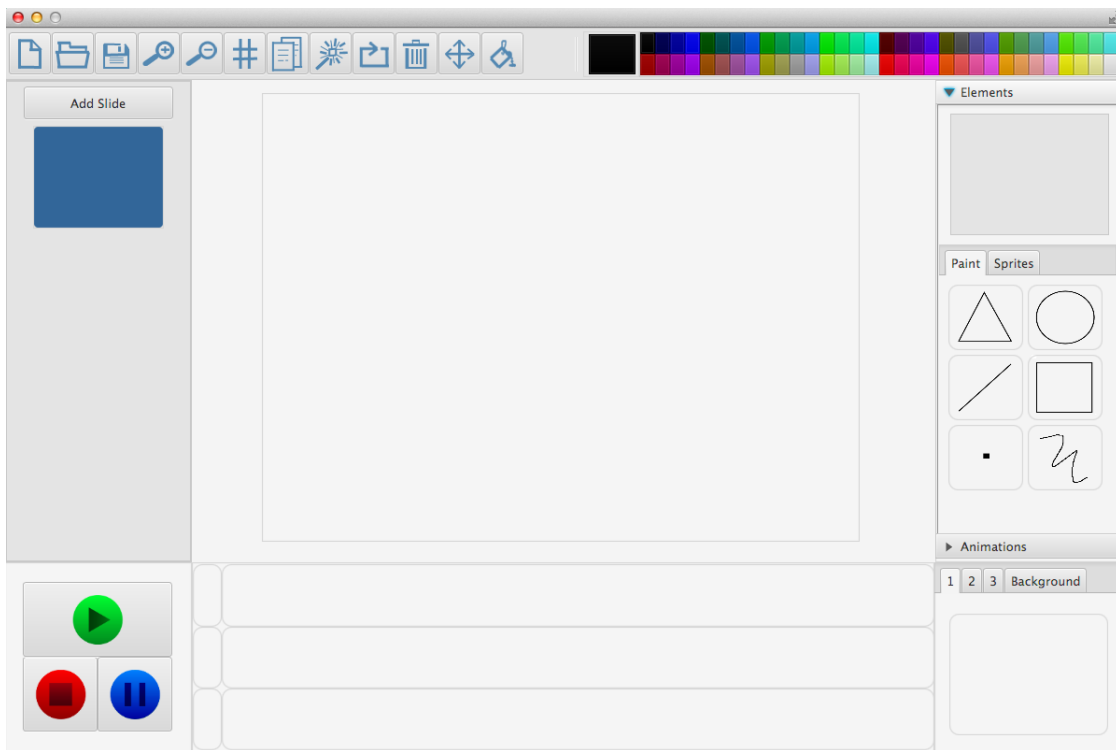


Figura 4.9 Software Creador de Contenido.

4.3.7. Codificación

La programación ha sido iterativa en base a la prioridad de los requisitos. Se han aplicado las técnicas de la programación orientada a objetos en base al diseño y arquitectura y se ha programado bajo el lenguaje de java principalmente por ser un lenguaje sencillo y confiable que permite ser utilizado en multiples plataformas y la librería de JavaFX por su amplia librería de gráficos.

- *Lenguaje Java*– Java es un lenguaje de programación rápido, seguro y confiable liberado por Sun Microsystems en 1995 [40]. Java es un lenguaje orientado a objetos e independiente pues sus programas son traducidos a un lenguaje intermedio llamado bytecode el cual se ejecuta en cualquier sistema operativo, software o dispositivo que tenga un interprete Java (máquina virtual Java) [41].
- *JavaFX*– Para este proyecto se utiliza la API (Application Programming Interface) de JavaFX. JavaFX es un conjunto de librerías gráficas y multimedia que permiten el diseño, creación, pruebas, depuración e implementación de aplicaciones de cliente enriquecida (Rich Client Applications). La librería de JavaFx es escrita en Java por lo que puede referenciar a cualquiera de sus APIs. Para la apariencia y estilo (“look and feel”) de la interfaz gráfica de usuario (GUI) se pueden utilizar hojas de cascada (CSS) separándoles de la implementación. También es posible separar la interfaz de usuario a través de un lenguaje de marcas basado en XML (FXML) o diseñarlo a través de la herramienta Scene Builder. Las APIs de JavaFX están totalmente integradas al Java SE Runtime Environment (JRE) y a el Java Development Kit (JDK) y debido a que el JDK está disponible para las principales plataformas de escritorio (Windows, Mac OS X y Linux). Las aplicaciones JavaFX compilados para el JDK 7 y posteriores se ejecutan también en las principales plataformas de escritorio. El apoyo a las plataformas ARM también ha puesto a disposición con JavaFX 8. El JDK para ARM incluye la base de gráficos y componentes de controles de JavaFX [42]. La arquitectura de JavaFx se muestra en la Figura 4.10.

En cuanto al diseño de la arquitectura basada en MVC (Modelo Vista Controlador) se ha implementado a través de JavaFX y el uso de FXML (XML que define la interfaz gráfica de usuario de JavaFX), siendo el “modelo” los objetos de dominio de la aplicación, la “vista” que consiste del FXML y el “controlador” que es el código que define el comportamiento de

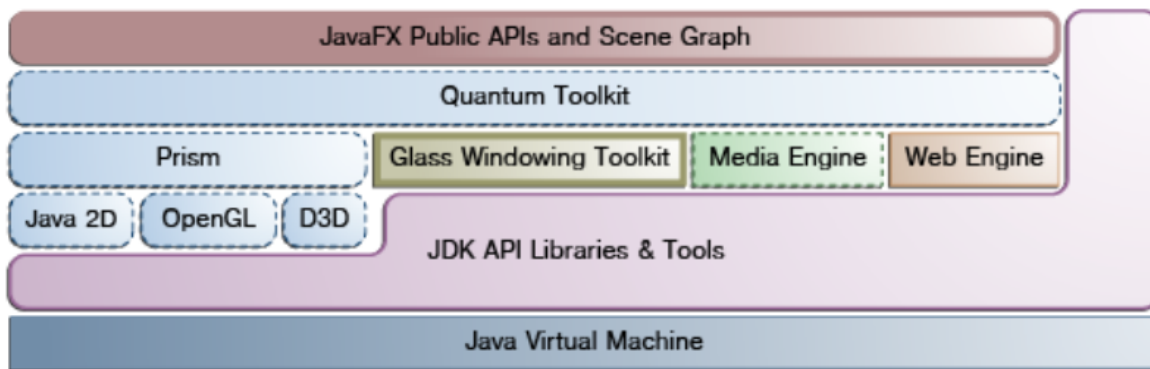


Figura 4.10 Arquitectura de JavaFX.

la interfaz gráfica de usuario para interactuar con el usuario [43].

Una de las ventajas de que la arquitectura ha sido evolutiva, es que al estar desarrollando los gráficos con las librerías de java, observamos que la resolución en la aplicación se mantenía distante de la plataforma DAVID PEP y no resultó idóneo para lo que buscábamos, pues buscamos que la experiencia al crear el contenido sea muy asemejada a la que se desplegará en la plataforma, por lo que se modificó la arquitectura para no utilizar la librería de JavaFX en las figuras y extendiendo y cambiarlas a figuras propias basadas en algoritmos de gráficas computacionales, dando a las figuras el poder ser manejadas totalmente por la aplicación, adecuando una simulación de los pixeles como lo mostraría DAVID PEP.

La diferencia del cambio se ve en la Figura 4.11 y la resolución puede ser ajustable para otros sistemas embebidos, pues se utilizan pixeles simulados, es decir, se unen varios para representar uno, según la configuración.

Por lo que al cambiar los gráficos de JavaFX, también se cambió la forma en la que se animaban las figuras, y para el trazo de la ruta a seguir, por el momento se utiliza el algoritmo de Bresenham.

En cuanto al formato para guardar el contenido creado y recuperarlo en el mismo software, se ha utilizado la misma estructura, basada en XML, esto por ser un estándar común y

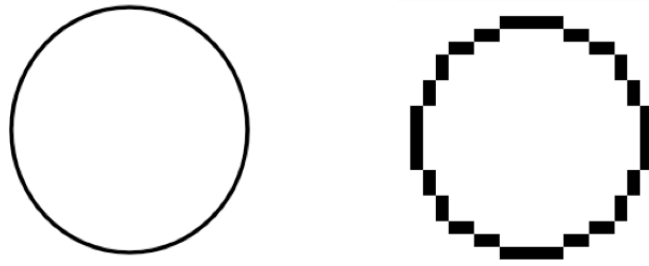


Figura 4.11 Cambio en el Manejo de Gráficos.

las diferencias entre el rendimiento de XML contra otros formatos no presenta algún inconveniente.

4.3.8. Validación y Verificación

La validación del software busca, mediante una serie de actividades, demostrar que se cumplen con los requisitos, por otro lado, la verificación es el conjunto de actividades que aseguran que el software implemente correctamente una función específica [44]. Por lo que en este caso conforme avanzaban las iteraciones se realizaban pruebas que validaran que los requisitos funcionales se cumplieran, principalmente utilizando pruebas de caja negra, las cuales examinan aspectos funcionales, y de forma sencilla se aplicaban pruebas de caja blanca para examinar la estructura interna de la aplicación, identificando y buscando la corrección de los desvíos que se presentaban durante el proceso. De forma parecida, con la validación se ha buscado que se cumplen con los requisitos en base a las metas y los objetivos del proyecto.

4.4. Interoperabilidad

Para la finalización de la propuesta se habla sobre la interoperabilidad del contenido multimedia, que consiste en la estructura para representar el contenido multimedia a través de los elementos básicos de un teatro (anteriormente mencionado) y en un formato de datos.

4.4.1. Interoperabilidad del Contenido

Aunque es conocido que java provee portabilidad a través de la JVM (WORA, o "write once, run anywhere") existen dispositivos que no permiten su ejecución de una forma nativa, como son los dispositivos móviles con IOS, incluso dispositivos con Android, aunque hay aplicaciones que permiten ejecutar un programa escrito en JavaFX, aunque en este caso resultó ser lento (en comparación de la pc), además de que la experiencia de usuario se puede mejorar bastante al diseñar una aplicación nativa en cualquiera de esos dispositivos, que tratar de adaptar la versión de escritorio ya desarrollada, por lo que se busca interoperabilidad del contenido multimedia a través de un lenguaje de serialización de datos, para que el contenido pueda ser creado y manejado en cualquier otro dispositivo (la actual desarrollada para escritorio, o alguna otra que se pueda desarrollar).

Para esta sección del proyecto se han revisado entre los lenguajes de XML, JSON y YAML y se ha optado por utilizar XML, que en cuanto ligereza resultó ser el intermedio entre estos tres en las pruebas efectuadas con esta estructura de datos, pero que resulta ser un estándar bastante utilizado y permite la validación a través de un esquema de datos, y según el analizador sintáctico resulta ser eficiente, por lo que en este software la forma en la que almacena y maneja el contenido también se ha realizado mediante XML y en etapas del desarrollo se ha validado mediante un archivo XSD. Aún así la aplicación de software permite exportar el contenido en XML, JSON y YAML, Figura 4.12.



Figura 4.12 Formatos de Serialización. XML, JSON, YAML.

En la Figura 4.13 se muestra la comparativa realizada entre seis diferentes contenidos creados con la aplicación creadora de contenido y bajo la estructura propuesta. En estos casos el formato de JSON mostró ser el de menor tamaño en el archivo, aunque YAML provee una mejor legibilidad, quedando XML como un posible intermedio entre estos y contando con el esquema de datos XSD que permitiría validar implementaciones en otros dispositivos.

En la Figura 4.14 se observa que bajo esta estructura de datos, en promedio, JSON fue casi 20 por ciento mas ligero en tamaño que XML y casi 50 por ciento que YAML.

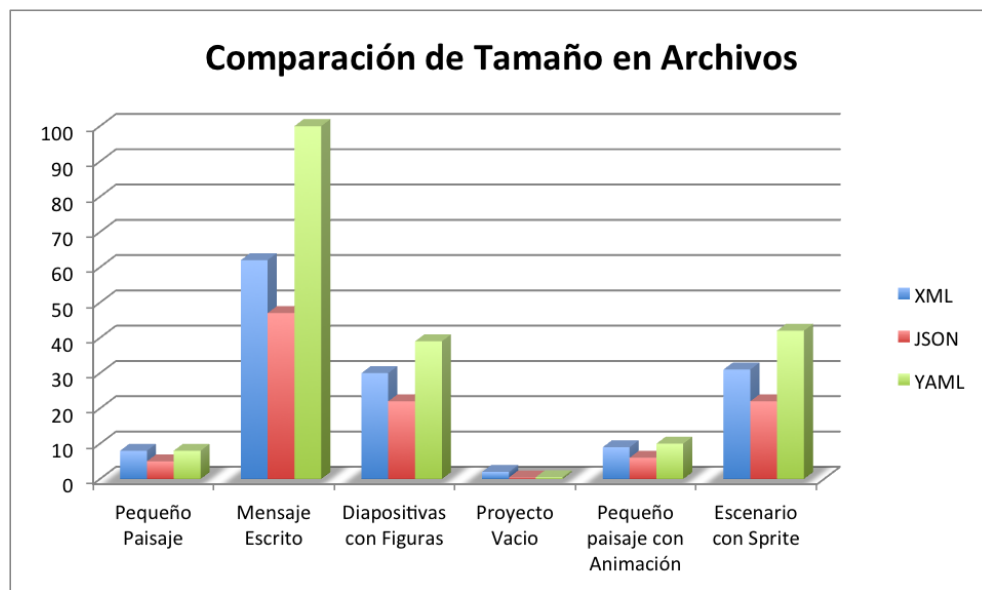


Figura 4.13 Comparación de tamaño de archivo con la estructura de datos.

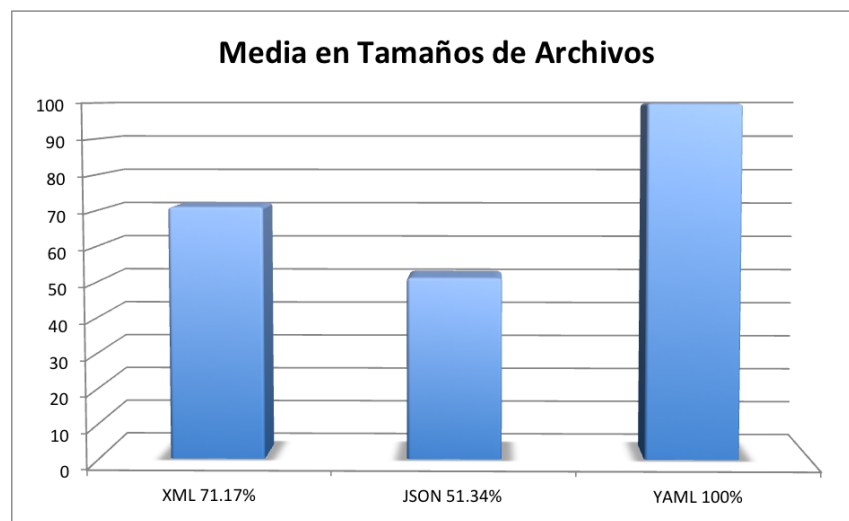


Figura 4.14 Media de tamaño de archivo con la estructura de datos.

4.4.2. Interoperabilidad con DAVID PEP

Mencionando que los resultados anteriores sobre el manejo de datos en XML pueden ser aplicado a un dispositivo con recursos suficientes, pues en DAVID PEP por los recursos limitados con los que actualmente cuenta se ha optado por enviar el contenido mediante un mapa de bits y en cuanto a los eventos y animaciones se planea seguir la estructura sencilla

para su representación (la estructura con los valores de animación y eventos de los sprites y/o elementos que se tienen como archivos separados, y en en DAVID PEP se manejan e incrustan en los planos estáticos), dejando en trabajo futuro el seguir con la investigación de formatos binarios (como BSON, MessagePack o alguno personalizado) para mejorar la eficiencia al utilizar recursos limitados, o seguir con la estructura bajo un formato de datos como JSON al aumentar los recursos en el hardware.

Por lo que, por los recursos limitados que maneja DAVID PEP se manejan los datos de los planos y sprites en archivos separados y bajo un formato tipo bitmap, enviando por cada bit un color que representa uno de los 64 colores que maneja DAVID PEP. En la Figura 4.15 se muestra un ejemplo de como se crean y almacenan los valores de los gráficos.

```
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFC0000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00FCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFC00FC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFC00FC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC FCFCFCFC
```

Figura 4.15 Formato de contenido para DAVID PEP.

Capítulo 5

Casos de Prueba

En el capítulo anterior se describió el diseño y el desarrollo del entorno de diseño de contenido que logra completar una parte de la plataforma propuesta para manejo de contenido multimedia creado por una aplicación de computadora que sea manejado por el sistema embebido DAVID PEP.

En este capítulo se muestra casos de pruebas para el diseño del contenido multimedia y despliegue del contenido en DAVID PEP.

5.1. Caso 1

5.1.1. Introducción

En este caso de prueba se diseña y crea un contenido que pretende demostrar el uso del entorno. Para esto se usan diferentes figuras geométricas y colores.

5.1.2. Descripción

Se crea un contenido multimedia a través del entorno de diseño de contenido. Este contenido se crea en las cuatro capas y se hace uso de un sprite, de forma que se pueda representar un escenario mediante la estructura del contenido basada en un teatro.

5.1.3. Resultados

Se muestra la validación del contenido y se validan los requisitos de software.

En la Figura 5.1 se muestra un escenario con una bailarina creado con el entorno (la figura de la bailarina fue basada en una figura del libro [45]).

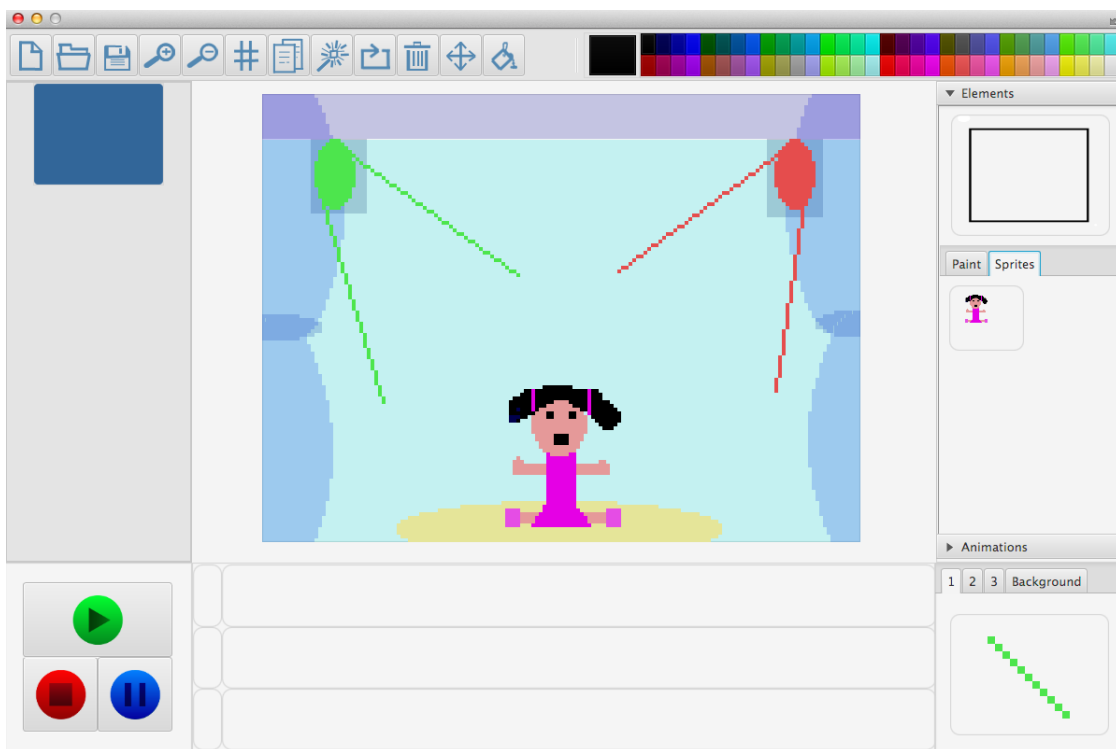


Figura 5.1 Entorno de diseño de Contenido.

5.2. Caso 2

5.2.1. Introducción

En este caso de prueba se diseña y crea un contenido con multiples sprites (la mororía con 2 elementos) y se demuestra la interoperabilidad del contenido con DAVID PEP, utilizando la matriz de pixeles generada por el software.

5.2.2. Descripción

Para la integración con DAVID PEP se ha desarrollado un módulo que permite la comprensión de los archivos enviados que representan el contenido multimedia.

De igual forma en DAVID PEP se ha ajustado la programación que permite desplegar el contenido a través de la matriz de pixeles.

5.2.3. Resultados

El dispositivo cargó el contenido recibido y cambiaba los elementos de los sprites cada segundo, en lo que se demostró el cumplimiento de los requisitos y una versión de prueba del entorno de diseño y de la plataforma.

En la Figura 5.2 se muestra las primeras series de elementos tipo sprite creadas con el entorno y en la Figura 5.3 se muestra la primer serie de sprites diseñados en el entorno y siendo reproducido en un medio visual mediante David Pep (las figuras fueron basadas en figuras del libro [45]).

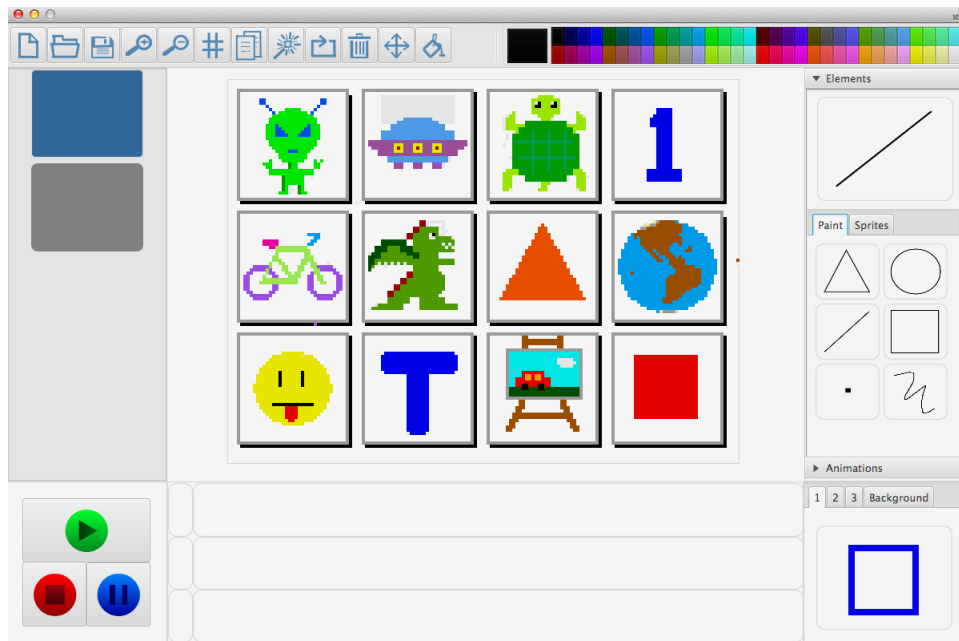


Figura 5.2 Entorno de diseño de Contenido.

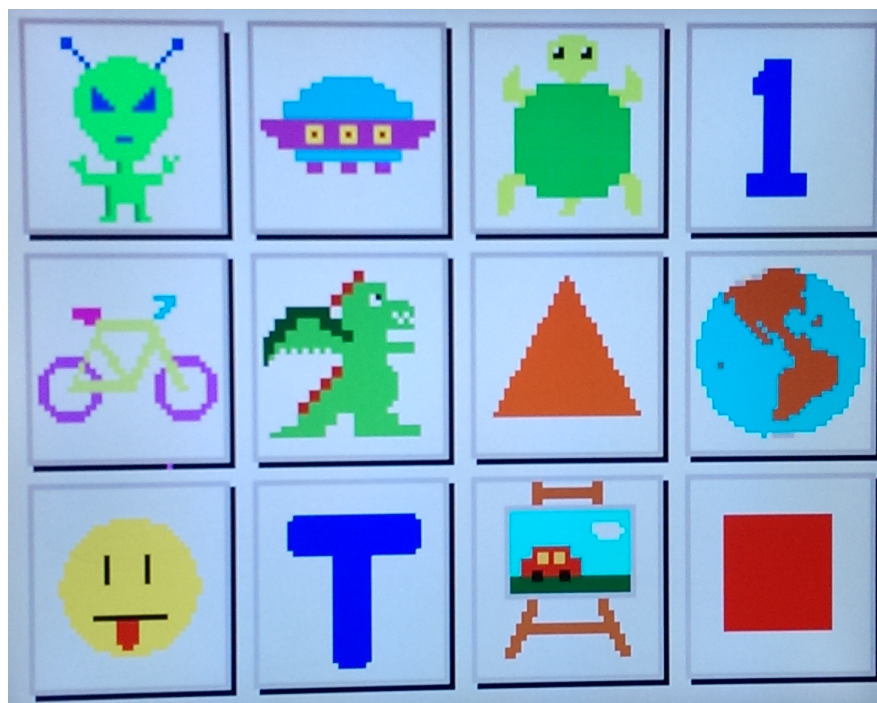


Figura 5.3 Conteido siendo Reproducido con David Pep.

Capítulo 6

Conclusiones y Trabajo Futuro

6.1. Conclusiones

Se concluye que si es posible obtener un entorno de diseño de contenido extendible que permita crear, manejar y transformar contenido multimedia desde una computadora y enviarlo hacia un sistema embebido de recursos limitados que tiene la posibilidad de manejar el contenido según eventos físicos, formando una plataforma que pueda ser utilizado con fines educativos. Aunque también se ha observado que por los recursos limitados que sistemas embebidos como DAVID PEP maneja, es recomendable seguir buscando y aplicando técnicas que permitan una mejor interoperabilidad con animaciones y eventos.

En cuanto a la aplicación de las TIC en la educación, en mi punto de vista me parece que si es elemental el seguir innovando y buscando áreas que permitan apoyar y mejorar las capacidades de enseñanza y aprendizaje, pero que de igual forma fomenten y permitan al alumno trabajar y poner de su parte, es decir, que la tecnología no lo haga todo por el alumno, como por ejemplo si fuera un aplicación para aprender a leer, que la tecnología no solo lea, sino que permita al alumno leer, y después la tecnología lo ayude, o como la propuesta de

este trabajo, que permite al alumno interactuar con el contenido, no solamente viéndolo, y viéndolo desde punto de vista tecnológico, pero que para fines de enseñanza y aprendizaje debería aún ser fundamentado por el área pedagógica.

En cuanto a los sistemas existentes, son muy buenas herramientas, y que en algunas de ellas, es posible la integración del software de este proyecto, o si no de este, de algún otro desarrollo, que pueda permitir mas interacción en el momento de crear algo enfocado a educadores, que sea sencillo y permita al educador crear algún contenido educativo interactivo.

Respecto al hardware utilizado de DAVID PEP, los recursos son suficientes para esta propuesta, pero como mencionaba, al ser limitados hay ciertas mejoras que serían costosas en cuanto a sus recursos, pero se ha visto que es posible una nueva versión de DAVID PEP con mas recursos de hardware, lo cual permitiría al software nuevas funcionalidades (aunque lo que se ha planteado en esta primera versión se ha permitido en el hardware actual).

Por parte del desarrollo del software (del entorno de diseño), aún han quedado puntos que pueden ser aplicados, pero se ha creado como base para la propuesta y para seguir en iteraciones que permitan su mejora.

En cuanto a la integración del software con el hardware, fue de lo mas complejo, por el hecho de utilizar un hardware que también está en fase de pruebas y que sigue en proceso de madurez, aunque si logró mostrar los resultados esperados para la propuesta de este trabajo.

En cuanto a la experiencia personal en este proyecto de tesis ha sido bastante, aunque enfocado principalmente en el software creador de contenido, así como el modelo y la selección del formato de serialización, el buscar las forma de integrarlo a un sistema con menos recursos lo hace muy interesante por las limitantes que pueden existir con la memoria y el procesamiento. De igual forma la experiencia en cuanto a la metodología para seguir este trabajo y redactarlo mediante esta tesis ha sido algo muy representativo, pues anterior a estos estudios no había realizado alguna tesis ni trabajo de investigación y desarrollo, por

lo que, aunque hayan quedado varias mejoras, se logra con el objetivo de brindar un aporte e ingresar en esta área y poder así, seguir en mejora con la experiencia de éste y futuros trabajos.

6.2. Trabajo Futuro

Al ser esta una propuesta inicial sobre el entorno de diseño quedan pendientes otros puntos que pudieran beneficiar este trabajo.

Entorno de Diseño de Contenido Multimedia (software)

- Realizar pruebas con formatos binarios como pueden ser message pack y bson para ver posibles mejoras en el rendimiento con el manejo de la serialización de los datos en los sistemas embebidos.
- Actualmente los eventos se agregan a la diapositiva y en mejora se podrían agregar a los elementos.
- Los eventos son iniciados por una linea del tiempo por lo que podrían ser independientes al tiempo.
- Aplicar diseño gráfico y mejoras en cuanto a experiencia de usuario.

Propuesta General

- En cuanto a la estructura de datos, se han usado nombres largos que buscan describir para lo que son, pero para mejor eficiencia al serializar los datos completos, se podrían cambiar los nombres largos por nombres cortos. Como en el caso de las animaciones, para indicar el inicio en la coordenada X se utiliza el nombre de “pointInitX” el cual

podría ser reemplazado por solo “x0” y el nombre de “pointFinishX” por solo “x1” mejorando en el tamaño del archivo y velocidad de lectura.

- Aunque se han realizado algunas pruebas de integración con DAVID PEP, la completa integración se ha planteado como trabajo futuro, pues es requerida mas investigación sobre las mejores técnicas e implementación para manejar el contenido multimedia con la propuesta de la estructura de datos en DAVID PEP, principalmente para las animaciones y la detección de eventos debido a los recursos de memoria que actualmente DAVID PEP maneja, así como también es posible buscar el incremento de los recursos de DAVID PEP. Siguiendo con la limitación de los recursos, se contempla la utilización de un formato sencillo para la animación y ejecución de eventos predefinidos. En cuanto al formato de animación SMIL, se presenta un ejemplo a continuación, pero que por ser un formato xml no se presenta como opción para DAVID PEP.

```
<svg width=""300px" height="100px">
<rect x="0" y="0" width="300" height="100" stroke="black" stroke-width="1"/>
<circle cx="0" cy="50" r="15" fill="blue" stroke="black" stroke-width="1">
<animate attributeName="cx" from="0" to="100" dur="5s" repeatCount="1" />
</circle>
</svg>
```

En cuanto a la herramienta de creación de contenido, en el emulador, se almacena en XML y se presenta para los eventos de la siguiente forma (dentro de la marca de slide):

```
<events >
<event AtMilli="5000" EventType="click" ActionType="continue"/>
</events >
```

Y para las animaciones se presentan de la siguiente forma, dentro del layer:

```
<elements>
```

```

<elementFigure slide="0" layer="layer1" type="CircleDP" name="circledp1"
x="55.0" y="42.0" color="0x000000ff" isFill="false" width="15.0" height="9.0">
<framesAnimation>
<frameAnimation TimeInit="2.0" FrameDuration="5.0" xInit="55.0" yInit="42.0"
xFinish="150.0" yFinish="150.0"/>
</framesAnimation>
</elementFigure>
</elements>

```

Por lo que para DAVID PEP se puede utilizar un formato que haga referencia al archivo de bits que contiene a la figura, como el siguiente:

```

load circledp1 55 42
move circledp1 55 42 150 150 2 5

```

Indicando mover el mapa de bits de circledp1 de x=55 y=42 a x=150 y=150 en el segundo 2 con una duración de 5 segundos.

Y para el evento algo como lo siguiente:

```
event 5 s c
```

Que representa un evento, en donde en el segundo 5 detenga sus animaciones y al recibir el evento predefinido para continuar, continúe con su contenido.

Basada esta plantilla en animaciones y eventos por tiempo, pero como se menciona, se busca adecuarlo para funcionalidad por eventos, mas que por una linea de tiempo.

- Realizar pruebas y buscar otras posibles aplicaciones con otros dispositivos embebidos con comunicación a dispositivos que puedan reconocer eventos externos predefinidos.

En experimentación que se realizó en este trabajo, se diseñó y creó un contenido para probar la interoperabilidad a través de la serialización de los datos del contenido (mediante la estructura de datos que representan el contenido multimedia, generados

a través de la aplicación de software) con un dispositivo embebido con recursos de hardware no tan limitados como es el dispositivo Raspberry Pi.

Para la integración con este dispositivo se desarrolló un módulo que permite comprender los archivos que representan el contenido, utilizando los nombres en el formato actual del modelo que describen a los elementos.

Este módulo se programó con C utilizando la librería Jansson para la manipulación de JSON y la librería de OpenVG para el manejo de gráficos.

La aplicación dentro del dispositivo validó la estructura y el formato, permitiendo la animación al detectar un evento externo.

El dispositivo Raspberry Pi cuenta con recursos muy amplios, al ser una computadora de tipo SBC(single board computer) de bajo costo, contando el modelo B con 512 MB de memoria ram, un CPU de 700 MHz y un procesador gráfico, así como puertos usb, un puerto ethernet, salida de video HDMI y salida de audio de 3.5 mm, y con con una ranura de tarjeta SD para almacenamiento [46] [47]. Por lo que es posible permitir otros usos con base en este trabajo, así como también poder enfocar este software de manejo de contenido a nivel primaria con otras opciones y mejor manejo de gráficos al permitir utilizar librerías soportadas en este dispositivo.

De igual forma se ve la posibilidad de utilizar otros medios que puedan realizar la detección de movimientos, como puede ser el Kinect, y hasta posibilidad de crear toda una plataforma basada en dicha tecnología.

Validación Pedagógica

- Esta es una propuesta que busca apoyar en el ambiente de educación preescolar, por lo que se ha encontrado validación principalmente a la literatura, pero falta una validación ante pedagogos que permitan una retroalimentación de esta versión prototipo.

Bibliografía

- [1] Consuelo Belloch Ortí. Las tecnologías de la información y la comunicación (t.i.c.). *Unidad de Tecnología Educativa. Universidad de Valencia.*
- [2] Ramón Carlos Suárez y Alonso. *Tecnologías de la Información Y la Comunicación, Introducción a los Sistemas de Información y de Telecomunicación.* Number 2. Ideas Propias, 2010.
- [3] Herminia Azinián. *Las tecnologías de la información y la comunicación en las prácticas pedagógicas.* Novedades Educativas, 2009.
- [4] Julio Cabero Almenara. Impacto de las nuevas tecnologías de la información y la comunicación en las organizaciones educativas. *Enfoques en la organización y dirección de instituciones educativas formales y no formales*, pages 197–206, 1998.
- [5] Antonia María Moya Martínez. Las nuevas tecnologías en la educación. *Innovación y experiencias educativas*, Guadix, Granada. 2009.
- [6] Victoria González García. María Marta Camacho Álvarez. Principios para la incorporación de tecnología digital en espacios preescolares. *Inter Sedes.*, IX(47-58), 2009.
- [7] *Aplicaciones Educativas de Las Tecnologías de la Información y la Comunicación.* Ministerio de Educación y Ciencia, Secretaría General de Educación, 2005.

-
- [8] Luis Horacio Ruiz Cuéllar Guadalupe López García Alma Yadhira Pérez Martínez, María Guadalupe Pedroza Zúñiga. La educación preescolar en México. condiciones para la enseñanza y el aprendizaje. *Instituto Nacional para la Evaluación de la Educación*, 2010.
- [9] Andrea Rolla y Mercedes Rivadeneira. Por qué es importante y cómo es una educación preescolar de calidad. *En Foco*, 2006.
- [10] Robert Rigal. Educación motriz y educación psicomotriz en preescolar y primaria. pages 53–57, 2006.
- [11] Kinect, <http://en.wikipedia.org/wiki/kinect>, consultado el 15-02-2015.
- [12] Kinect educational, <http://www.microsoft.com/en-us/education/products/xbox-kinect/>, consultado el 15-02-2015.
- [13] Fisher-Price. Smart fit park, <http://service.mattel.com/us/productpopup.aspx?prodno=p4494&siteid=27>, revisada el 15-12-2014. 2009.
- [14] MIT Media Lab. Lifelong Kindergarten. Scratch, <http://scratch.mit.edu/about/>, consultado el 20-12-14. 2014.
- [15] Nintendo. Mario maker, <https://www.nintendo.es/juegos/wii-u/mario-maker-titulo-provisional-892704.html>, consultado el 19-12-2014. 2014.
- [16] M. Weiser. Ubiquitous computing, acm conference on computer science, 1994, p. 418.
- [17] G. Licea L. Aguilar and J. A. Garcia-Macias. Lisandra: A low-cost experimental wireless sensor network. *In Proceedings of the 2010 IEEE Electronics, Robotics and Automotive Mechanics Conference, Washington, DC, USA, 2010, pp. 259-264*.

- [18] Jesús Adán García López. Detección de patrones de movimiento mediante mediciones de inercia en base a un estimador difuso. 2013.
- [19] Laureano Felipe Gómez Dueñas. Interoperabilidad en los sistemas de información documental. Technical report, 2007.
- [20] Carlos Andrés Morales Machuca. Estado del arte: Servicios web. 2010.
- [21] Cen lun-lie Zhang Ya-Mei. Research on method of transformation from bitmap to vector graphics based on adobe illustrator cs4. 3rd international conference on advanced computer theory and engineering (icacte). 2010.
- [22] Microsoft. Serialization, <http://msdn.microsoft.com/es-es/library/ms233843.aspx>, consultado el 21-12-14. 2013.
- [23] M.F.L. Hufkens. Xml for embedded systems. the role of xml in distributed embedded networks. *Master's Thesis.*, 2003.
- [24] Richard Jones. Xml provides extra help for embedded systems. *IEE Computing Control Engineering*, 2005.
- [25] Json. Introducing json, <http://json.org>, consultado el 21-12-2014.
- [26] YAML. Yaml ain't markup language (yaml™) version 1.2, <http://www.yaml.org/spec/1.2/spec.html>, consultado el 21-12-2014. 2009.
- [27] Bson. Bson, <http://bsonspec.org>, consultado el 21-12-2014.
- [28] MessagePack. Messagepack, <http://msgpack.org>, consultado el 21-12-2014. 2013.
- [29] Office Open XML. Drawingml, <http://officeopenxml.com/drwoverview.php>, consultado el 22-12-2014. 2014.

- [30] SVG. Svg, <http://www.w3schools.com/svg/>, consultado el 23-12-2014. 2014.
- [31] w3c. Smil, <http://www.w3.org/audiovideo/>, consultado el 22-12-2014.
- [32] Format container, http://en.wikipedia.org/wiki/digital_container_format, consultado el 16-03-2015. 2014.
- [33] The Institute of Electrical and Electronics Engineers. Ieee standard glossary of software engineering terminology. 1990.
- [34] Md. Ismail Jabiullah Md. Alamgir Kabir Sagar, Md. Mijanur Rahman. Model for identifying the security of a system: A case study of point of sale system. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 2013.
- [35] Jakob Nielsen. Usability 101: Introduction to usability, <http://www.nngroup.com/articles/usability-101-introduction-to-usability>, consultado el 26-12-2014. 2014.
- [36] Inc Scrum. The scrum papers: Nut, bolts, and origins of an agile framework. 2012.
- [37] Sebastián Priolo. *Métodos ágiles*. USERS, 2009.
- [38] Germán Harvey Alférez Edwin Rafael Mago. El papel de la arquitectura de software en scrum. *Software Guru #30*, 2010.
- [39] Nicolás Tedeschi. ¿qué es un patrón de diseño?, <http://msdn.microsoft.com/es-es/library/bb972240.aspx>, consultado el 02-01-2015.
- [40] Oracle. What is java, https://www.java.com/en/download/faq/whatis_java.xml, consultado el 15-12-2014.
- [41] Ignacio Zahonero Martinez Luis Joyanes Aguilar. *Programación en C, C++, JAVA y UML*. Number 743-745. McGraw-Hill, 2010.

-
- [42] Oracle. Javafx overview, <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm>, consultado el 15-12-2014. 2014.
- [43] Oracle. Best practice: Enforce model-view-controller (mvc) with fxml, http://docs.oracle.com/javafx/2/best_practices/jfxpub-best_practices.htm, consultado el 01-01-2015. 2014.
- [44] Roger Pressman. *Ingeniería del Software. Un enfoque práctico. Sexta edición*. McGraw-Hill / Interamericana de México, 2005.
- [45] *Help your kids with computer coding*. Dorling Kindersley Limited, 2014.
- [46] Raspberry pi, http://es.wikipedia.org/wiki/raspberry_pi.
- [47] Raspberry pi model b, <http://www.raspberrypi.org/products/model-b/>.