



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DESARROLLO DE UN RECURSO DIDÁCTICO PARA
REFORZAR DESTREZAS MATEMÁTICAS EN NIÑOS USANDO
APLICATIVO MÓVIL Y RECONOCIMIENTO DE VOZ”**

INFORME DE MATERIA INTEGRADORA

Previo a la obtención del Título de:

INGENIEROS EN CIENCIAS COMPUTACIONALES

Joseph Iván Gallardo Pincay

Pedro Andrés Iñiguez Menoscal

GUAYAQUIL – ECUADOR

AÑO: 2016

AGRADECIMIENTOS

Quiero agradecer a Dios por ayudarme en cada paso que doy y abrirme las puertas a oportunidades que me han ayudado a crecer como persona y como profesional.

También quiero agradecer a mi Madre puesto que su gran dedicación a la familia y su incondicional apoyo han sido primordiales para poder continuar día a día mi carrera. A mi hermana por apoyar tanto a mi madre y a mi hermanito por causarme momentos de felicidad que son necesarios para seguir adelante. A mi abuelita porque siempre ha estado con nosotros desde pequeños y siempre nos ha ayudado. A mi familia en general porque siempre he contado con ellos para cualquier necesidad y hemos compartido muchos momentos felices.

Agradezco a mi grupo de Amigos que forme por el Colegio pues siempre me han brindado su apoyo y le han agregado emoción a mi vida incluso cuando había preocupación y dificultades.

También a mis Amigos de la universidad, pues hemos superado muchas adversidades juntos.

Le doy gracias al dr. Vintimilla y al dr. Romero por abrirme las puertas del CIDIS, lugar donde he tenido tantas buenas experiencias personales y profesionales y donde soy recibido con mucha calidez.

Pedro Iñiguez

Quiero agradecer primeramente a Dios, por darme la oportunidad de estar vivo y a pocos pasos de completar una de las metas de mi vida, graduarme de la universidad siguiendo una carrera que me gusta y apasiona; junto a mi familia y amigos.

Así mismo, quiero agradecer con todas mis fuerzas a la principal partícipe de que yo pueda estar hoy por culminar mis estudios universitarios, mi madre, ya que, gracias a su constante apoyo y amor incondicional, me brindo las fuerzas necesarias para seguir adelante con mis estudios, así como el apoyo económico necesario para poder desarrollar mis estudios fuera de mi provincia de origen. De la misma manera, a mi hermana quien a lo largo de los años me ha brindado su apoyo y juntos hemos pasado tantos buenos y malos momentos, pero el amor fraternal siempre ha salido victorioso.

No puedo olvidar a todos los amigos y amigas que conocí durante el trascurso de mi vida universitaria, quienes también formaron un pilar fundamental en el cual nos pudimos apoyar unos a otros para seguir adelante durante las situaciones más adversas. Estoy seguro que la mayoría de estos lazos de amistad seguirán vivos durante muchos años más, por lo que les estoy inmensamente agradecido.

Por último, un gran abrazo a todos los profesores que tuve el privilegio de ser su alumno, es gracias a ellos y su pasión por enseñar que he podido crecer como persona y como profesional. A pesar de inconvenientes que pudieron surgir durante estos 5 años, a todos los admiro, siento un gran respeto por ellos y les estaré eternamente agradecido.

Joseph Gallardo.

DEDICATORIA

El presente proyecto se lo dedico con mucho cariño y esfuerzo a Dios, mi familia y amigos.

Pedro Iñiguez

Este proyecto va dedicado con mucho esfuerzo a Dios, a mi madre, a mi hermana, a mi familia y amigos, sin los cuales no hubiera sido posible su realización.

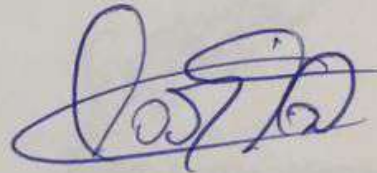
Joseph Gallardo.

TRIBUNAL DE EVALUACIÓN



Ph. D. Dennis Romero

PROFESOR EVALUADOR

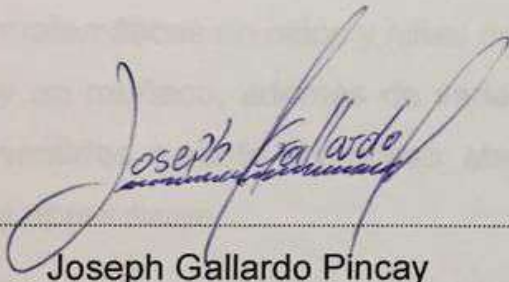


Ph. D. Dennys Paillacho

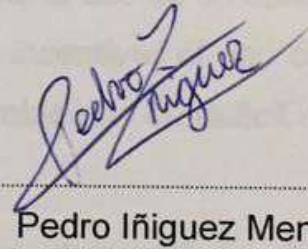
PROFESOR EVALUADOR

DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, nos corresponde exclusivamente; y damos nuestro consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"



Joseph Gallardo Pincay



Pedro Iñiguez Menoscal

RESUMEN

Durante muchos años, se han analizado los diferentes problemas de aprendizaje que pueden ser responsables del bajo desempeño académico de ciertos grupos de niños y niñas. Uno de los enfoques implementados en los últimos años para atacar estos problemas, es la inclusión de tecnologías móviles dentro del aula de clases, y así aprovechar la versatilidad de estos dispositivos para atacar varios de los problemas de aprendizaje a ser mencionados en este trabajo.

En el presente trabajo, se plantea una solución para ayudar a mejorar destrezas matemáticas en niños y niñas de edad escolar, mediante el uso de un aplicativo móvil y un muñeco, además de varias características para incentivar el uso de múltiples sentidos durante el proceso; atacando varios de los problemas de déficit de atención y aprendizaje.

El presente proyecto, al ser llevado a aulas de clase, demostró su efectividad al momento de capturar la atención de los niños y ayudar al maestro a identificar temas en los que fallaban los niños y así poder llevar un currículo de clases de acuerdo a las necesidades de los niños.

El presente documento está estructurado de la siguiente forma: Análisis del problema, Solución Tecnológica implementada, Descripción de funcionalidades de juego, pruebas y resultados; y por ultimo conclusiones y recomendaciones.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA	iv
TRIBUNAL DE EVALUACIÓN	v
DECLARACIÓN EXPRESA	vi
RESUMEN.....	vii
ÍNDICE GENERAL.....	viii
CAPÍTULO 1	1
1. ANALISIS DEL PROBLEMA.	1
1.1 Antecedentes.....	2
1.2 Estado del Arte	3
CAPÍTULO 2.....	5
2. SOLUCIÓN TECNOLÓGICA IMPLEMENTADA.....	5
2.1 Módulos de funcionalidad de la solución	6
CAPÍTULO 3.....	13
3. DESCRIPCION DE FUNCIONALIDADES DE JUEGO.	13
3.1 Módulo 1 Jugador.....	13
3.1.1 Sintetizador de Voz	13
3.1.2 Reconocimiento de Voz	14
3.1.3 Generación de Retos Matemáticos	16
3.1.4 Configuración Raspberry Pi	17
3.1.5 Reconocimiento de Rostro	18
3.1.6 Manipulación del Motor de Paso	20
3.1.7 Integración del Motor con la Cámara	23
3.2 Módulo 2 Jugadores	31
3.2.1 El Servidor.....	33
3.2.2 Estructuras Dentro del Servidor	34
3.2.3 Eventos	36

3.2.4	Estructura Principal del Juego.....	38
3.2.5	Estructuras del Cliente	40
3.2.6	Animaciones.....	42
3.2.7	Lazo del Juego.....	43
3.2.8	Manejo de las Criaturas	44
CAPITULO 4.....		46
4.	PRUEBAS Y RESULTADOS.....	46
4.1	Pruebas al módulo de 1 Jugador.....	46
4.2	Pruebas al módulo multijugador	48
CONCLUSIONES Y RECOMENDACIONES		51
BIBLIOGRAFÍA.....		53
ANEXOS.....		55

CAPÍTULO 1

1. ANALISIS DEL PROBLEMA.

La inclusión de nuevas tecnologías en el aula de clases está ampliando los recursos disponibles para el aprendizaje desde primaria, secundaria, hasta la universidad. A pesar de la disponibilidad de estas tecnologías, no siempre son integradas de una manera efectiva [1].

Muchas veces, el bajo desempeño académico de los estudiantes es atribuido a la falta de contenido o actividades que logren captar la atención durante las sesiones de clase. Al 2013, según el Instituto nacional de estadísticas y censos, un 7.95% del total de la población mayores de 5 años posee celular o dispositivos inteligentes [7]. Esto hace evidente que, la tecnología forma parte de las actividades diarias de la población general. Sin embargo, existen muchas consideraciones al momento de incluir dichos dispositivos en el deseado proceso de aprendizaje [2], tales como la capacidad de los profesores para usar tecnologías, como herramientas didácticas para las clases hacia las necesidades de los estudiantes, o si se proveerá una retroalimentación hacia estudiantes y profesores.

No obstante, el aprendizaje de las matemáticas tiende a ser una dificultad para muchos niños hasta el día de hoy y aunque son muchos los factores que originan esto, como problemas en el desarrollo cognitivo, falta de memoria, trastorno de concentración, dislexia, entre otras. Pocas son las medidas que se toman para incentivar el ejercicio de destrezas matemáticas.

El presente trabajo busca ser un recurso para el aprendizaje de las matemáticas en niños entre 5 y 10 años brindándoles retos de cálculo y lógica matemática simples que les ayudaran a reforzar lo aprendido en clase o incluso a tomar iniciativa y adquirir nuevo conocimiento, tanto en el aula de clase como fuera de la misma.

Debido a la gran facilidad de adquirir dispositivos móviles pocos de bajo costo, este proyecto busca integrar diferentes dispositivos por medio de la implementación en hardware y software de tal manera que pueda brindar una experiencia completa al

niño para poder captar su interés y mejorar su desempeño. Adicionalmente, el proyecto presenta una interfaz intuitiva para el niño y el profesor, el docente no tendrá problemas en ajustarlo a las necesidades del niño debido a que se requiere una mínima interacción por parte del profesor para que el sistema continúe de manera autónoma.

Al utilizar múltiples dispositivos se cubre diferentes enfoques que permitirán que el proyecto sea utilizado también fuera de las aulas de clase a las aulas de clases, debido a que la aplicación principal va a ser desarrollada para dispositivos android, muy populares para el uso en cualquier lugar. Partiendo del objetivo principal, este trabajo busca incorporar nuevas tecnologías a los entornos escolares y además hacerlo de manera que los estudiantes puedan aprovecharlo al máximo tanto en la clase como en el hogar.

El presente trabajo abarca dos modos de aprendizaje sumamente importantes. El aprendizaje activo, donde el niño participará de manera directa en el aula de clase, y el aprendizaje cooperativo, donde todos los niños forman parte de la actividad. [3]

Ya habiendo identificado varios aspectos del proyecto, detallaremos más a fondo los inconvenientes de un escaso control del aprendizaje y como afecta esto a los estudiantes.

1.1 Antecedentes

El término “Math Disability” o discapacidad matemática es el término que se usa para referirse a los problemas en el aprendizaje de matemáticas encontrados en las personas, los efectos varían según el problema y pueden derivarse de problemas de memoria, problemas de desarrollo cognitivo o un problema visual-espacial, recayendo principalmente en la realización de operaciones matemáticas que podemos dividir en 4 dificultades: [4]

- i) Realizar cálculos: puede darse debido dificultades de memorización o por dificultades en el desarrollo cognitivo. Aquellos que poseen esta discapacidad suelen tener muy poco progreso en el desarrollo de operaciones matemáticas a lo largo del tiempo.
- ii) Resolver problemas: en este caso nos referimos a resolver aquellos problemas matemáticos que se encuentran en pequeños textos.

- iii) Reglas y procedimientos: Se debe a un retraso en el entendimiento de las reglas y procedimientos matemáticos.
- iv) Lenguaje: este problema implica que el niño no es capaz de darle significado a las operaciones matemáticas que se le presenta.

Sin embargo, las discapacidades en el aprendizaje no solo están presentes en las matemáticas. El instituto de neurociencias de la junta de beneficencia de Guayaquil en el 2014 aclaró que existen 6 problemas principales en el aprendizaje de niños y adolescentes. El déficit de atención con hiperactividad, dislexia, inmadurez viso motriz, trastorno mixto, trastorno espacial en lectura y déficit de atención y concentración. [5]

Fuera de las matemáticas y, enfocándose solamente en el proceso de aprendizaje, se ha comprobado que mecanismos de enseñanza que incluyen múltiples estímulos (multisensorial) funcionan de manera más efectiva (dígase mejora de retentiva o captura de atención), frente a mecanismos enfocados solamente al uso de estímulo único [8]. Esto debido a la evolución del cerebro humano hacia aprender y operar en ambientes naturales multisensoriales. De esta manera, la implementación de mecanismos multisensoriales puede llegar a influir de manera positiva en personas con problemas de aprendizaje, en comparación con los métodos de enseñanza tradicionales.

1.2 Estado del Arte

Debido a que la tecnología se desarrolla rápidamente, cada vez contamos con más estudios que buscan probar que podemos utilizar nuevos recursos de hardware y software como una herramienta para mejorar los resultados de aprendizaje. No solo estudios sino también múltiples prototipos han sido desarrollados para el ámbito académico.

En el campo de las matemáticas diferentes estudios demuestran que al utilizar recursos multimedia y software podemos aumentar la atención de los niños durante el aprendizaje [6]. Además, podemos generar soluciones personalizadas en dispositivos móviles que no solo se enfoquen a las aulas de clases sino también a los niños con dificultades particulares, como por ejemplo el autismo.

Una solución propuesta para el aprendizaje de inglés, en la cual luego de un amplio estudio y pruebas de técnicas de aprendizaje, fue implementada en una aplicación móvil conectada a un robot, dando muy buenos resultados en el aprendizaje.

Como se muestra en la figura 1.1, actualmente existen aplicaciones móviles que se enfocan en repasar matemáticas de una forma simple y con pocos recursos didácticos, como imágenes de animales o frutas, que se limitan en solamente poner al niño a practicar sin alguna inmersión adicional.

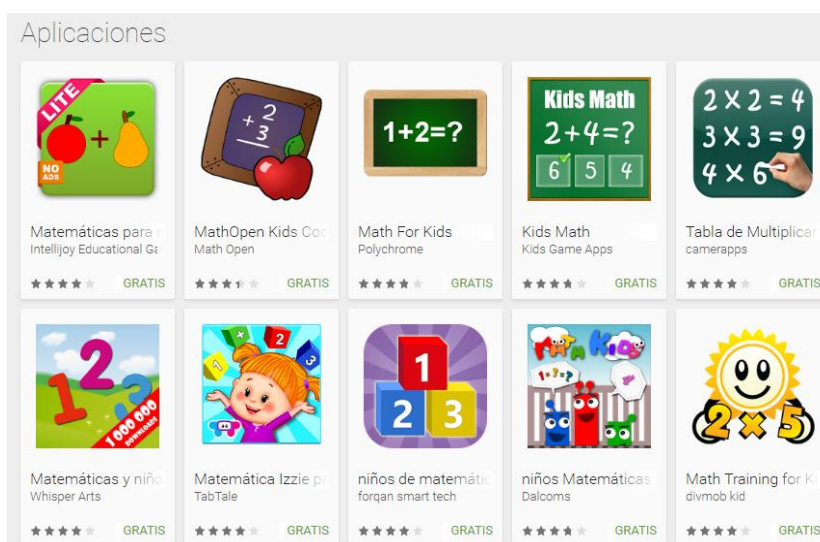


Figura 1.1 Aplicaciones de juegos matemáticos en Play Store

CAPÍTULO 2

2. SOLUCIÓN TECNOLÓGICA IMPLEMENTADA.

El presente proyecto es una solución implementada por medio de hardware y software, con el objetivo de atacar los problemas de aprendizaje relacionados a operaciones matemáticas, mencionados en el capítulo anterior, presentados en niños de 5 a 10 años.

Tomando como referencia un proyecto integrador presentado anteriormente [9], el cual desarrolló una aplicación móvil para el sistema operativo Android para recuperar agilidad matemática en estudiantes universitarios; se utiliza el mismo enfoque hacia el desarrollo de una aplicación móvil para atacar problemas de aprendizaje en niños; agregando módulos de reconocimiento y síntesis de voz para aumentar la efectividad del aprendizaje [8], además, con la finalidad de facilitar la interacción y aumentar la captura de atención de los niños, se plantea la implementación de un anexo en forma de un muñeco, que además de servir como soporte para el dispositivo móvil que ejecuta la aplicación, es capaz de conectarse a la aplicación por medio de bluetooth y tendrá la capacidad de reconocer y seguir el rostro de la persona que se encuentre en frente de él. Se muestra un esquema general de la solución en la figura 2.1.

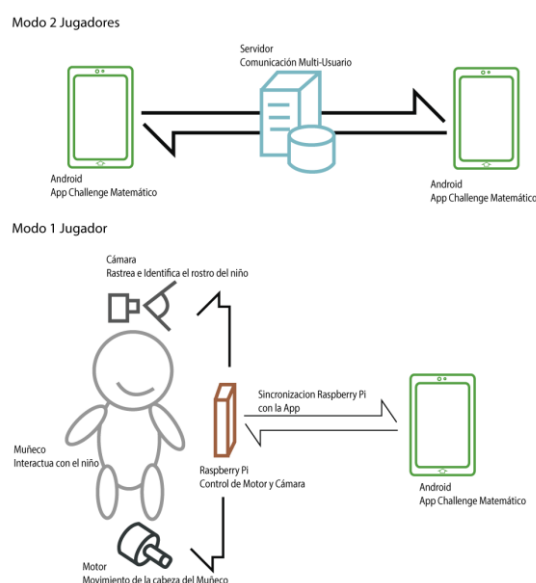


Figura 2.1 Diseño de la Solución

2.1 Módulos de funcionalidad de la solución

2.1.1 Reconocimiento de voz

Google, entre las características que ofrece el sistema operativo Android, se encuentra el dictado automático por voz o también llamado reconocimiento de voz. En apoyo a los desarrolladores, Android permite usar esta característica desde el API del sistema operativo en desarrollo de aplicaciones, sin embargo hasta hace unos años esta característica necesitaba una conexión activa a internet ya que el procesamiento de la onda sonora se realizaba en la nube de google, sin embargo los dispositivos móviles actuales tienen la capacidad computacional para realizar reconocimiento de voz sin la necesidad de una conexión a internet, esta característica está disponible en dispositivos que cuentan con el sistema operativo android desde la versión de API número 16 (a.k.a jellybean). Para usar esta característica es necesario ingresar a la configuración de idioma del dispositivo, luego en la opción de dictado por voz e ir hacia la opción "Reconocimiento de voz sin conexión" y descargar el idioma a utilizar en el reconocimiento.

Esta característica del sistema operativo android resulta de mucha utilidad ya que permite una implementación fácil y efectiva de una las características claves del proyecto, al permitir la interacción con el usuario final de una manera más efectiva y ayuda a reforzar el aprendizaje [8], como fue mencionado en el capítulo anterior.

2.1.2 Sintetización de voz

De forma similar el sistema operativo Android provee herramientas muy valiosas en la implementación del proyecto, ocurre así con la implementación del módulo de sintonización de voz, del cual existe una gran variedad de implementaciones de terceros, con la desventaja de ser propietarias; El sintetizador de voz de google presenta varias ventajas:

- Nativa del sistema operativo.
- Variedad de lenguajes disponibles.

- Fluidez.
- Posibilidad de uso offline.

Debido a estas ventajas, se utiliza el sintetizador de voz desde el API de Android para implementar otro de los puntos clave del proyecto hacia la mejora del aprendizaje [8].

2.1.3 Detección de Rostro

Una de las características más importantes de un proyecto de interacción Hombre-Máquina es la inmersión y captura de atención de los usuarios finales, en este caso orientado hacia la mejora del proceso de aprendizaje. Para esto, se agrega la funcionalidad de detección de rostro, la cual facilita una retroalimentación al usuario final, detectando su rostro y realizando seguimiento del mismo mediante movimientos de la cámara, dando la sensación de enfoque y capturando mejor la atención.

El algoritmo de detección de rostro es implementado usando OpenCv 2.4.9 para Python 2.7. Este algoritmo usa 'cascade classifiers' [10] para la detección de rostros, opencv provee por defecto "haar cascade classifiers", que detecta rostros con gran precisión, aunque con un elevado costo computacional; debido a esto han sido reemplazados en una versión posterior por LBP (Local Binary Patterns) cascade, ya que estos utilizan enteros en sus operaciones a diferencia de haar cascade que utiliza flotantes, dando como resultado un procesamiento más rápido, pero perdiendo alrededor de un 10% a 20% en precisión.

2.1.4 Raspberry Pi

Dada la necesidad de procesamiento computacional que conlleva el reconocimiento facial, además de la variedad de facilidades que ofrece, se incluye una Raspberry Pi 3 para manejar los módulos de detección de rostro, movimiento del muñeco y conexión hacia la aplicación móvil.

La Raspberry pi o “RasPi” es una SBC (Single Board Computer) desarrollada en Reino Unido por la fundación Raspberry Pi; este mini computador es muy versátil debido a su tamaño un poco mayor a una tarjeta de crédito y prácticamente todas las funcionalidades de un computador de escritorio. Para la solución implementada se usa una Raspberry Pi 3 como la que se presenta en la figura 2.2.

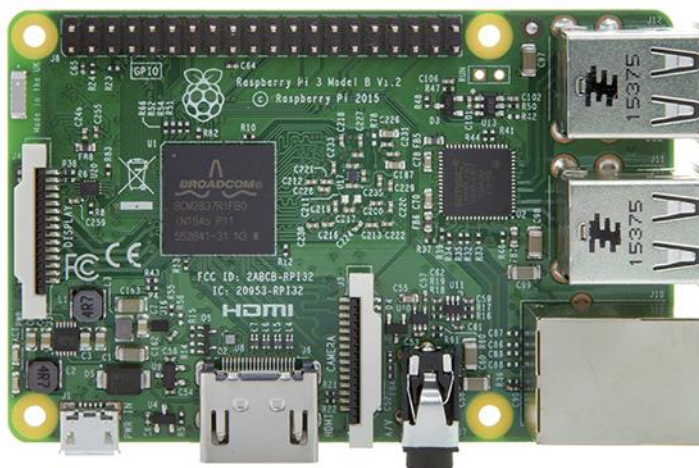


Figura 2.2 Mini Computador Raspberry Pi 3

A la Raspberry Pi se le instaló el sistema operativo Linux, gracias a esto es posible ejecutar la implementación del módulo de detección de rostros mencionado en la sección anterior, además de controlar otros módulos presentes en la RasPi como los pines GPIO o el módulo de conexión Bluetooth que han sido igualmente utilizados en el desarrollo de la solución.

2.1.5 Motor de Pasos

Los motores de paso son ideales para la construcción de mecanismos que requieren movimientos precisos, su característica principal es que pueden moverse paso a paso por cada pulso que se aplique. Gracias a esta característica el motor resulta ideal para la implementación del seguimiento del rostro del usuario final.

El motor de paso es controlado a través de la Raspberry Pi mediante los pines GPIO que se conectan a una pequeña placa de control, interpretando las señales recibidas y transformándolas en movimiento.

Para que el motor de pasos funcione correctamente, es necesario cumplir con ciertos requerimientos en cuanto a la alimentación de corriente; se necesita un voltaje entre 5V – 12V y una intensidad de corriente entre 1.2A -3A. Para cumplir con estos requerimientos se incorpora un adaptador de corriente de pared que proporciona 9V y 1.5A, con el cual se logra la potencia necesaria para que el motor realice los movimientos necesarios, pero cuenta con la desventaja de que restringe completamente la portabilidad del sistema al necesitar estar cerca de una toma de corriente para funcionar.

Debido a los problemas mencionados con el adaptador y con la finalidad de mantener al máximo la portabilidad del sistema, se encontró una nueva alternativa de alimentación en una batería de ácido de plomo con un voltaje de salida 12V y 2.6A; la cual será acoplada en la parte inferior de una silla de soporte para el muñeco, como se puede observar en la figura 2.3. A pesar de ser una solución efectiva en cuanto a portabilidad, el voltaje proporcionado es demasiado alto, por lo que es necesario encontrar una forma de reducir este voltaje hasta 9V. Luego de buscar opciones para lograr la reducción de voltaje, se agregó un adaptador inteligente de voltaje para carros, el cual obtiene un voltaje de 12V de entrada y mediante una rueda se puede regular entre varios voltajes de salida, uno de ellos 9V. Así mediante la utilización de un adaptador para conectar los puertos positivo y negativo de la batería de ácido de plomo hacia el conector de entrada del adaptador inteligente queda ensamblado el sistema de alimentación del motor de pasos.



Figura 2.3 Batería Incorporada al sistema de alimentación del muñeco

Como último agregado, se acopla a la salida del adaptador inteligente un pequeño modulo voltímetro para monitorear el voltaje que llega al motor de pasos, como se observa en la figura 2.4. Esto ayuda a identificar cuando la batería se encuentra descargada para así proceder a recargarla y que el sistema funcione de manera adecuada.



Figura 2.4 Voltímetro que monitorea el voltaje que llega al motor

2.1.6 Conexión Bluetooth

El raspberry pi 3 cuenta con bluetooth integrado, lo cual es muy beneficioso para el desarrollo del presente proyecto por razones que mencionaremos a continuación. Si bien es cierto que la aplicación debe ser fácil de usar para los niños, también debe ser sencilla de manipular para los maestros. Para facilitar el manejo del sistema integrado se optó por aprovechar el bluetooth integrado del raspberry pi para establecer una

comunicación rápida, directa y sencilla entre esta y la aplicación, brindándole a los profesores la posibilidad de iniciar la dinámica con la aplicación en tan solo dos clicks, gracias a que la aplicación identificará los dispositivos bluetooth en la cercanía para comunicarse con el muñeco sin necesidad de internet o algún cable adicional.

2.1.7 Muñeco

El mayor punto enfoque del proyecto es la interacción del usuario final con el sistema, ya que es necesario capturar la atención de manera más efectiva para asegurar un mejor proceso de aprendizaje y obtener mejores resultados que se reflejen en el rendimiento académico de los usuarios. Para esto se presenta un muñeco, que será encargado de capturar la atención hacia la actividad a realizar.

En cuanto al aspecto físico, el muñeco es de fabricación artesanal, fabricado con tela de color piel y relleno de algodón, tiene la apariencia de un niño de entre 5 y 10 años, de cabello y ojos de color negro, vestido con un mameluco jean, camisa azul a rayas, zapatos y gorra, como se puede observar en la figura 2.5; además es capaz de sentarse y mover la cabeza.



Figura 2.5 Muñeco para el desarrollo del proyecto

Cubriendo la parte de funcionalidad, la cabeza se encuentra unida al cuerpo a través de un soporte, el cual se acopla al motor de pasos ubicado en el interior del cuerpo a la altura del pecho; esta estructura permite el

movimiento de la cabeza a través del motor de pasos, así mismo, en la parte superior de la cabeza, dentro de la gorra se encuentra ubicada la cámara web que obtiene las imágenes de entrada para realizar la detección de rostros. El muñeco posee también una pequeña mochila en la cual se almacenaran la Raspberry Pi y demás accesorios de control.

2.1.8 Multijugador

Para reforzar el aprendizaje del niño, no es suficiente concentrarse solo en actividades dentro del aula de clase. Mucho de lo que los niños aprenden en clases debe ser reforzado en los hogares. Es por esto que nuestra solución incluye un módulo que le permitirá al niño usar la aplicación desde su hogar, sin necesidad del muñeco, de forma divertida y competitiva. El módulo multijugador incluye un enfrentamiento de retos matemáticos sencillos contra otros usuarios, lo cual permitirá al niño a mejorar su conocimiento en matemáticas. La temática incluirá criaturas que irán cambiando su forma física a medida que el niño acumula victorias y traerá consigo una interfaz agradable con la que el niño estará jugando por horas.

CAPÍTULO 3

3. DESCRIPCION DE FUNCIONALIDADES DE JUEGO.

Como parte de la estrategia para presentar de manera llamativa el proceso de refuerzo de destrezas matemáticas hacia los niños, el flujo de las actividades de refuerzo ha sido moldeado en forma de juego, con dos escenarios posibles para el uso del sistema; un modo de un jugador para desarrollar el aprendizaje activo y un modo multijugador o de dos jugadores para desarrollar el aprendizaje cooperativo.

3.1 Módulo 1 Jugador

El modo 1 jugador esta implementado para abarcar varios elementos necesarios para un aprendizaje exitoso de niños de edad preescolar y primaria por medio de juegos y aprendizaje activo. Se trata de la interacción de los niños en un aula de clase con un muñeco robot. El muñeco robot seguirá los rostros de los niños y presentará retos matemáticos que, dependiendo de la dinámica del profesor, pueden ser contestados por uno o varios niños de manera oral. El objetivo es que el niño muestre interés y preste atención en el muñeco para poderle responder con entusiasmo el reto propuesto, de manera que así el niño aumentara su interés hacia las matemáticas e ira practicando a su paso la resolución de expresiones matemáticas simples. El desarrollo del modo 1 jugador cuenta con 2 partes que son el desarrollo del muñeco utilizando la raspberry pi y el motor de paso, y el desarrollo de la aplicación móvil en Android que cuenta con la interfaz de acceso al muñeco, el sintetizado y reconocimiento de voz.

3.1.1 Sintetizador de Voz

Como primer paso, se inició un proyecto utilizando Android Studio, en el cual se probaría la sintetización de voz. La configuración es sencilla, en Android es posible utilizar la clase TextToSpeech para inicializar un sintetizador de voz simplemente pasándole parámetros como el idioma o el tono de voz. Rápidamente se desarrolló una aplicación demo con esta funcionalidad, sin embargo para el desarrollo del juego, es necesario que

exista una mayor integración del sintetizado con el resto de funcionalidades. Es necesario detectar cuando el sintetizador ha terminado de hablar para proceder a acciones como cambiar de reto o inicializar el reconocimiento de voz. Para esto hay que realizar una configuración adicional, utilizando `UtteranceProgressListener` (Oyente de Progreso de Pronunciación) cuya tarea es estar pendiente del proceso de sintetizado y emitir eventos según el progreso del mismo. Es posible utilizar el evento de finalización junto con un identificador para ejecutar acciones al final de que el sintetizador se ha comunicado con el niño.

3.1.2 Reconocimiento de Voz

Paralelo a la sintetización de voz, se realizó un demo del reconocimiento de voz, el cual ayudaría a entender cómo funciona y que más adelante nos serviría como base del juego. En Android existen componentes llamados "*Intent*" que son inicializables desde la actividad en la que nos encontremos, esto permite invocar un componente "*RecognizerIntent*" que viene incorporado en Android el cual se encargará de realizar la tarea de reconocer respuestas dichas por el niño. En Android, las actividades cuenta con un método que puede escuchar resultados de componentes inicializados desde la misma; por medio de este método es posible obtener los resultados del reconocimiento luego de que el niño ha terminado de hablar. Parámetros adicionales para la configuración de este componente son la duración del reconocimiento y el idioma de reconocimiento, el cual usa el idioma configurado en el dispositivo. El resultado que obtuvimos fue el siguiente:

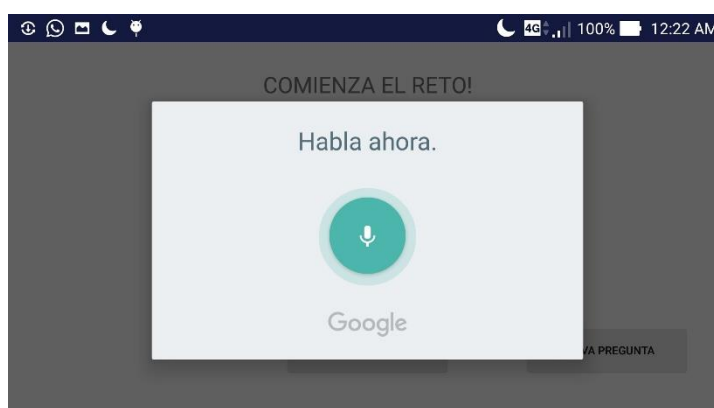


Figura 3.1 Componente de reconocimiento de voz de Google

En la figura 3.1 el componente salta sobre la actividad principal como una interfaz completamente separada de la que está debajo, incluso la tapa casi completamente. Al hablar, la interfaz simula pulsaciones que indican que se está escuchando a alguien hablar y finalmente procede a entregar la frase detectada al método anteriormente mencionado. De esta manera es posible comparar que el resultado que dice el niño es la respuesta correcta al reto.

Este componente realiza la tarea deseada con gran exactitud, sin embargo, presenta un inconveniente; el cual es que no se mantiene constantemente escuchando y detectando frases, para que funcione de manera continua requiere una interacción con el usuario, cosa que no deseamos en el modo 1 jugador. Además, tapa completamente la actividad en pantalla, la cual es fundamental para brindarle retroalimentación al niño de lo que escuchó, el número de intentos que ha realizado por reto, entre otros.

Para solucionar este problema, fue necesario que la actividad principal implemente los métodos de la interfaz “*RecognitionListener*” como si fueran propios. Realizando esto la actividad del juego de 1 jugador se vuelve a la vez un reconocedor de voz que puede acceder a todos los métodos de la interfaz “*RecognitionListener*” y manejarlos según los

necesitemos. Existen 3 métodos importantes que son necesarios para mantener el reconocimiento de voz funcionando:

- `onReadyForSpeech`: este método establecerá un temporizador que luego de unos segundos de inactividad, enviará una señal de error al reconocedor de voz.
- `onError`: este método se dispara cuando existe algún error en el reconocedor, ya sea por error de inicialización o tiempo de espera. En este método lo que hacemos es terminar el servicio anterior e iniciar inmediatamente uno nuevo.
- `onResult`: este método se dispara cuando el reconocedor ha detectado una voz y emite una interpretación de lo que escucho. Este método nos permite analizar el resultado y levantar un nuevo reconocedor de ser necesario.

Manejando de esta manera el reconocedor de voz es posible controlar el estado del mismo e incorporarlo en la actividad sin tener la interfaz de google ocultando toda retroalimentación que es posible darle al niño por medio de la pantalla. El resultado de esta implementación se observa en la figura 3.2.



Figura 3.2 Pantalla de juego usando la interfaz “*RecognitionListener*”

3.1.3 Generación de Retos Matemáticos

Una vez teniendo dos de las funcionalidades principales necesarias para la interacción en el módulo de 1 Jugador, para poder integrarlas, es

necesaria la actividad que utilizará ambos recursos, los retos matemáticos; los cuales serán dichos por medio de sintetización de voz y cuyo resultado será comparado con la respuesta hablada obtenida por medio del reconocimiento de voz. Los retos matemáticos están enfocados en niños de primaria, por lo que deben cambiar constantemente sin incrementar su dificultad drásticamente. Es por esto que se han elegido 3 diferentes operaciones que consideramos prudentes para los niños: suma, resta y multiplicación. El tipo de operación será escogido de manera aleatoria y el reto se generará de la siguiente manera:

- Suma: ambos operandos toman un valor aleatorio entre 1 y 10.
- Resta: el minuendo toma un valor aleatorio entre 1 y 10 mientras que el sustraendo toma un valor entre el minuendo y 1.
- Multiplicación: al igual que en la resta, el primer operador toma un valor entre 1 y 10 mientras que el segundo toma un valor entre el valor del primer operando y 1

Tanto la respuesta como la expresión son almacenadas en variables para que más adelante la aplicación pueda sintetizar la expresión y comparar el resultado hablado que da el niño con la respuesta correcta.

El niño cuenta con 3 intentos para cada reto, una vez que se equivoque la tercera vez, la aplicación le presentará un nuevo reto. Como la interacción va a ser a través de un muñeco, el niño tiene la opción de comunicarse con él y pedirle que repita el reto que le acaba de decir, esto con la finalidad de que el niño se sienta cómodo con el muñeco y no se limite solo a decir la respuesta. Esta última funcionalidad se encuentra implementada también con el reconocedor y sintetizador de voz, que al detectar la palabra “repite” procederá a repetir la expresión.

3.1.4 Configuración Raspberry Pi

Una vez desarrolladas las funcionalidades básicas en el juego en modo 1 jugador, se procede a configurar el Raspberry Pi para poder ser utilizada como el núcleo del procesamiento en el muñeco. Para poder configurarlo

es necesario una tarjeta micro sd de mínimo 4gbs de memoria. En ella será instalado el sistema operativo “raspbian”, el cual es una distribución de debían de Linux para ser utilizado en la Raspberry Pi; se puede observar su instalación en la figura 3.3. Utilizando Linux se obtiene un mayor control con respecto a los puertos GPIOs, las entradas y salidas, además de que es mucho más sencillo configurar dependencias desde una consola de Linux e incluso desarrollar sin un IDE para que la carga del dispositivo durante el trabajo sea la mínima posible.

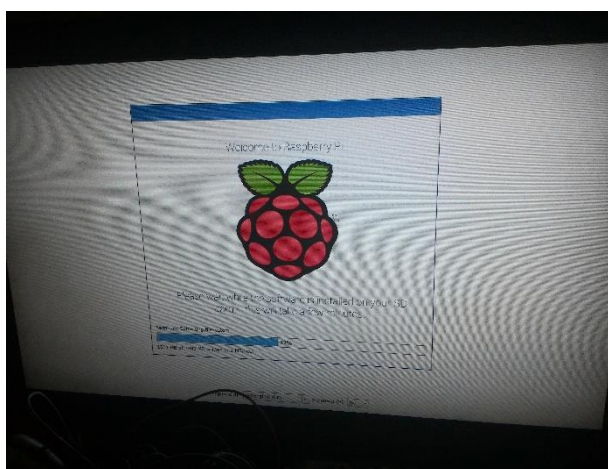


Figura 3.3 Instalación de Raspbian en la Raspberry Pi 3

Para la implementación con la manipulación de los puertos gpios del Raspberry Pi se utilizará Python, puesto que la cantidad de documentación y librerías que existen es bastante amplia y su instalación es sencilla. De igual manera Python cuenta con una librería de OpenCV, la cual permitirá realizar la detección de rostro para que el muñeco pueda mover la cabeza siguiendo los movimientos del rostro de un niño.

3.1.5 Reconocimiento de Rostro

Para el reconocimiento de rostro con Python, tuvimos se tuvo acceso a una cámara especial para Raspberry Pi, como se muestra en la figura 3.4.



Figura 3.4 Módulo de cámara de la Raspberry Pi 3

El script realizado en Python carga el recurso de la cámara y ajusta la resolución a grabar con la misma. Luego de esto se crea un lazo infinito que se mantendrá activo siempre que haya un fotograma nuevo capturado por la cámara, dicho fotograma entra al lazo para ser manipulado. Para poder realizar reconocimiento de rostros, es necesario pasar el fotograma a escala de gris ya que esto hará el procesamiento mucho más ligero y para detectar rostros no es necesario los 3 canales de color, basta con un solo canal, pues la detección se basa en contrastes. Una vez obtenida la imagen en escala de gris, para obtener los rostros presentes en la imagen utilizamos el clasificador que viene por defecto en OpenCv, el cual es un clasificador HaarCascade y pasamos el fotograma como parámetro. Esto nos devolverá un conjunto de rostros detectados. Estos rostros no son nada más que un conjunto de valores, x, y, ancho y alto que forman un rectángulo donde se encontró el rostro. Es posible pintar estos rostros en el fotograma horizontal como primera versión del proyecto, como se demuestra en la figura 3.5.

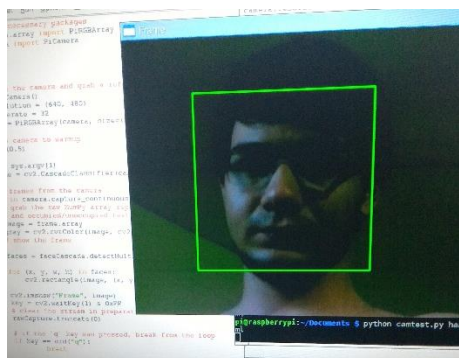


Figura 3.5 Prueba del sistema de detección de rostro en la Raspberry Pi 3

El uso del clasificador HaarCascade conlleva un costo computacional alto, y debido a los recursos computacionales limitados presentes en un mini computador como la Raspberry Pi, provocan que la velocidad de detección para cada frame de video sea lenta. Para corregir esto y permitir que la detección de rostros se realice de la forma más fluida posible, se reemplaza el clasificador Haar Cascade, por un clasificador LBP (Local Binary Patterns) cascade. Debido a que este tipo de clasificador utiliza números enteros dentro de sus operaciones, a diferencia de haar cascade que utiliza flotantes, causa que el procesamiento sea más rápido.

3.1.6 Manipulación del Motor de Paso

Para manipular el motor de paso, se cuenta con la ventaja de que el Raspberry pi tiene puertos o pines de entrada y salida denominados “GPIO” que pueden ser utilizados para muchos fines, pero básicamente lo que hacen es enviar o recibir señales de voltaje. Entre los tipos que existen están:

- Pines de alimentación: Pines que se encargan de entregar carga o enviar la corriente a tierra. Para el proyecto será utilizado el pin que entrega 5 voltios y un pin que envía la corriente a tierra.
- No Funcionales: son pines que actualmente solo ocupan espacio pero que en versiones futuras tendrán algún uso específico.

- GPIO normales: Son aquellos pines que por medio de programación pueden recibir o enviar señales. Para poder hacer girar un motor de paso, necesitamos 4 de estos pines.
- GPIO especiales: Son pines que suelen conectarse a componentes específicos como por ejemplo un visor de interfaz o para comunicación serial.

Para su manipulación, Python cuenta con una librería para Raspberry Pi llamada RPi la cual cuenta con una interfaz para manipular los pines GPIO. Lo que hay que hacer es especificar los pines que serán utilizados, que en este caso son 4 pines normales de índice 5, 6, 25 y 8 que en la arquitectura del Raspberry Pi se encuentran el uno al lado del otro. Estos 4 pines deben ir conectados a un módulo que permite en específico la rotación de motores de paso. Este módulo se lo conoce como L298N y cuenta con la siguiente distribución:

- 4 Conectores de entrada: estos conectores son los que recibirán las señales del Raspberry Pi, por lo que deben ir conectados a los pines 5, 6, 25 y 8.
- 4 Conectores de salida: estos conectores son los que envían las señales al motor en forma de secuencia para que el motor de paso gire correctamente.
- Entrada de alimentación: para que el motor gire, necesita de energía adicional a la que es usada como señal por los pines. Esta entrada recibe voltaje para alimentar al motor.
- Salida a Tierra: al alimentar el motor, debemos cerrar el ciclo de corriente con una conexión a tierra. Esta salida irá conectada al pin de tierra del Raspberry Pi al igual que a la fuente de alimentación para distribuir mejor el paso de corriente sin dañar el hardware.
- Salida de 5v: este conector nos sirve para alimentar cualquier componente adicional. Para nuestro proyecto no es necesario.

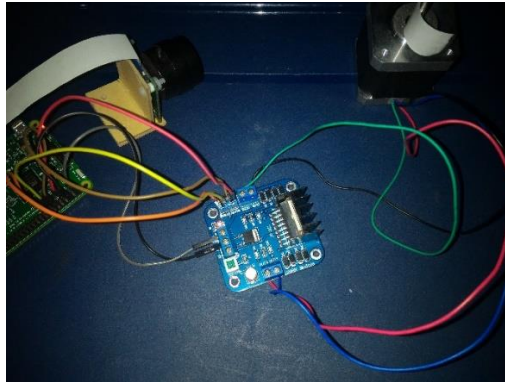


Figura 3.6 Conexión entre la Raspberry Pi 3, módulo L298N y motor de pasos.

Una vez configurada la conexión entre el Raspberry pi, el módulo L298N y el motor de paso como se observa en la figura 3.6, es necesario indicarle la secuencia de giro al motor hacia la derecha, tal como se observa en la figura 3.7 a continuación.

```
print 'derecha'
for i in range(5):
    setStep(1, 0, 1, 0)
    time.sleep(delay)
    setStep(1, 0, 0, 1)
    time.sleep(delay)
    setStep(0, 1, 0, 1)
    time.sleep(delay)
    setStep(0, 1, 1, 0)
    time.sleep(delay)
```

Figura 3.7 Secuencia de señales para giro hacia la derecha.

Para que gire hacia la izquierda, la secuencia debe ser invertida, como se observa en la figura 3.8.

```
print 'izquierda'
for i in range(5):
    setStep(0, 1, 1, 0)
    time.sleep(delay)
    setStep(0, 1, 0, 1)
    time.sleep(delay)
    setStep(1, 0, 0, 1)
    time.sleep(delay)
    setStep(1, 0, 1, 0)
    time.sleep(delay)
```

Figura 3.8 Secuencia de señales para giro hacia la izquierda.

Notemos que `setStep` cambia el voltaje de los pines 5, 6, 25 y 8 respectivamente, entre el valor máximo y el valor mínimo para que, mediante la secuencia, el motor empiece a girar. Además, cada vez que se envía la señal para que el motor gire, se repite la secuencia 5 veces para que el giro sea notorio.

3.1.7 Integración del Motor con la Cámara

Para integrar ambos componentes (cámara y motor) es necesario utilizar las funciones para la rotación del motor, en conjunto con el lazo que se encuentra detectando rostros constantemente. Al momento de detectar un rostro, se debe calcular si este rostro está ubicado muy a la derecha o muy a la izquierda según la resolución del frame capturado y de estarlo, se procede a mover la cámara en el sentido necesario para centrar el rostro detectado en la imagen. Ya que se está detectando constantemente el rostro, únicamente fue necesario ir rotando el motor de manera constante hasta que el rostro se encuentre centrado en relación al frame capturado. Esto fue, ubicando la cámara sobre el eje de rotación del motor como se observa en la figura 3.9.



Figura 3.9 Integración del motor de pasos con la Pi Camera

Los giros del motor se ejecutan en hilos para distribuir mejor los recursos del Raspberry Pi; sobre todo el hardware y operaciones que se están realizando. Aunque esto mejora el rendimiento, puede ocasionar el siguiente problema: si el rostro se encuentra a la izquierda se le dará la

orden al motor de girar hacia la izquierda, pero si el rostro se mueve bruscamente a la derecha, una nueva orden de giro hacia la derecha se le enviará al motor. Tendremos dos hilos cuyas acciones son opuestas compitiendo por quien realiza su paso primero lo que puede hacer que la cabeza del muñeco empiece a vibrar, pero sin movimiento notable, fallando completamente en el seguimiento de rostro.

La solución a este problema fue implementar un candado Mutex, cuyo comportamiento se basa en que, si un hilo que va a hacer girar el motor es inicializado, pedirá permiso para girarlo, si ya hay algún otro hilo girando el motor, este hilo tendrá que esperar su turno para poder girar el motor, que será cuando el hilo que se encuentra girando el motor haya terminado de girarlo. La función para girar el motor queda como se muestra en la figura 3.10.

```
def forward():
    with mutex:
        print 'derecha'
        for i in range(5):
            setStep(1, 0, 1, 0)
            time.sleep(delay)
            setStep(1, 0, 0, 1)
            time.sleep(delay)
            setStep(0, 1, 0, 1)
            time.sleep(delay)
            setStep(0, 1, 1, 0)
            time.sleep(delay)
```

Figura 3.10 Implementación de función de giro con candado Mutex.

3.1.8 Integración de motor y cámara con muñeco

Una vez integrado el proceso de detección de rostro con el movimiento del motor de pasos, el sistema está listo para ser integrado dentro del muñeco. El muñeco fue encargado a hacer en base a un modelo de muñeco artesanal presentado como muestra, pero con varias modificaciones de acuerdo a nuestras necesidades; como, por ejemplo: que la cabeza se encuentre separada del cuerpo para poder ser acoplada al motor de pasos, que el muñeco sea capaz de sentarse y ubicar un dispositivo móvil en su regazo, que cuente con un gorro donde se pueda

ubicar una cámara, y que cuente con una pequeña mochila para poder ubicar los componentes utilizados como la Raspberry Pi y la placa L298N.

Para poder acoplar la cabeza del muñeco al cuerpo del mismo fue necesario construir una pequeña base de soporte, para poder manipular la cabeza del muñeco y que pueda ser acoplada al brazo del motor y se pueda efectuar el movimiento, el resultado se muestra en la figura 3.11.



Figura 3.11 Acoplamiento de la cabeza del muñeco con el brazo del motor de pasos.

Una vez ajustada la base de soporte a la cabeza del muñeco, se procedió a introducir el motor de pasos dentro del pecho del muñeco, para esto fue necesario retirar parte del relleno del mismo y realizar un pequeño corte en la parte posterior para poder retirar los cables que llevan las señales al motor. Se puede observar en la figura 3.12 el resultado del acoplamiento de la cabeza con el cuerpo del muñeco mediante la base de soporte de la cabeza hacia el brazo de rotación del motor de pasos ubicado dentro del cuerpo del muñeco.



Figura 3.12 Integración de motor de pasos con el muñeco.

Sin embargo, debido a la longitud del conector de la cámara de la Raspberry Pi, no es posible ubicarla en la parte superior de la cabeza del muñeco ya ensamblado, por lo que fue necesario reemplazarla con una cámara web convencional para realizar la detección de rostros, proceso que no presento mayor inconveniente, ya que fue acoplada con facilidad a la gorra complementaria, y en lo referente a la codificación solo fue necesario modificar un par de líneas de código.

3.1.9 Comunicación Bluetooth

Una vez implementadas ambas partes del módulo de 1 jugador es necesario que exista una comunicación entre ellas, así la aplicación en android puede saber cuándo existe un rostro frente al muñeco, para así controlar el flujo del juego. Para esto se aprovecha la característica de la Raspberry Pi en su versión 3, ya que esta cuenta con un módulo bluetooth incorporado, el cual es usado para establecer la comunicación hacia la aplicación.

Para poder abrir un canal de comunicación entre la aplicación en android hacia la Raspberry Pi, primeramente, se tienen que conceder los permisos necesarios a la aplicación para que pueda manipular el módulo de bluetooth del dispositivo móvil, esto se realiza en el manifiesto de la

aplicación. Posteriormente para poder establecer la conexión bluetooth se utilizan elementos de tipo “BluetoothDevice” que representa el dispositivo con el cual se establecerá la conexión y “BluetoothSocket” que sirve para iniciar y mantener la conexión.

Para seleccionar el dispositivo con el que se va a realizar la conexión, se desarrolla un nuevo activity, como se observa en la figura 3.13, la cual mediante un objeto “BluetoothAdapter”, es capaz de verificar y manipular el estado del módulo Bluetooth del dispositivo android. En este caso habilita el modulo bluetooth del dispositivo (si se encuentra inactivo), obtiene la lista de los dispositivos bluetooth vinculados, y empieza el escaneo por nuevos dispositivos disponibles.

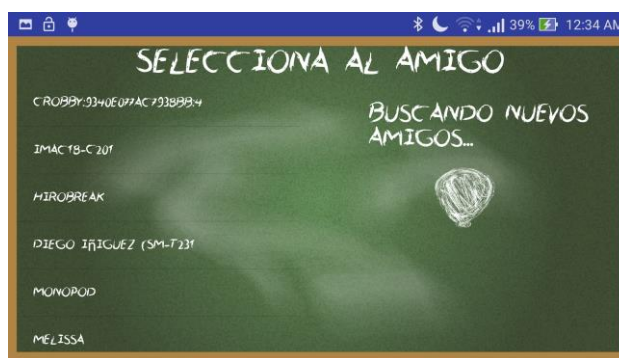


Figura 3.13 Pantalla de búsqueda y conexión de dispositivo bluetooth.

Una vez seleccionado un dispositivo disponible de cualquiera de las dos listas presentadas, es utilizado para crear el canal de comunicación entre la aplicación android y la raspberry pi; dicha comunicación se llevará a cabo mediante objetos “BluetoothSocket”, los mismos que serán del tipo “RFCommSocket”, tanto para el socket “servidor”, que estará ubicado en la raspberry pi, escuchando constantemente los requerimientos de la aplicación; como para el socket “cliente”, ubicado en la aplicación android, enviando periódicamente requerimientos hacia la raspberry pi. El tipo de conexión “RFCOMM” es el más comúnmente usado ya que es soportado por los APIs de android, además es netamente orientado hacia la conexión, transmisión y transporte por medio de bluetooth.

```

public void startConnection(BluetoothDevice mDevice){
    this.mmDevice = mDevice;

    try {
        Log.i("BT SINGLETON", "STARTING CONNECTION AT " + mDevice);
        Method m = this.mmDevice.getClass().getMethod("createRfcommSocket", new Class[]{int.class});
        this.mmSocket = (BluetoothSocket) m.invoke(mmDevice, 1);
        mmSocket.connect();
    } catch (Exception e){
        Log.i("myStuff", "EXCEPTION THROWN: " + e.getMessage());
    }
}

```

Figura 3.14 Conexión bluetooth mediante BluetoothSocket.

Una vez establecida la conexión como se muestra en la figura 3.14, esta tiene que ser mantenida durante toda la duración de la sesión de juego, ya que por medio de esta se enviarán requerimientos periódicamente hacia la raspberry para conocer si se encuentra una persona ubicada frente al muñeco y seguir con flujo normal de la actividad. Para poder mantener la conexión activa, fue necesario encontrar una forma de mantener el mismo objeto "BluetoothSocket" activo desde la conexión inicial, durante la actividad de juego y ser capaz de cerrarlo para dar paso a una nueva conexión al finalizar la actividad de juego. Para esto se implementó una clase que será la encargada de manejar todos los aspectos de la conexión bluetooth; tales como iniciar la conexión dado un dispositivo bluetooth, enviar mensajes y finalizar la conexión. Así mismo esta clase implementa el patrón de diseño "Singleton", el cual restringe la instanciación de elementos de esta clase y permite que exista únicamente un elemento de esta clase durante toda la ejecución de la aplicación, permitiendo así que la conexión hacia la raspberry permanezca activa, mientras sea necesario, durante la ejecución de la aplicación. Se presenta un diagrama UML de esta clase en la figura 3.15.

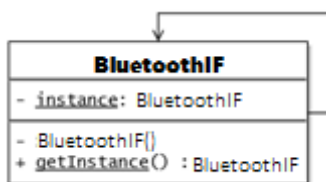


Figura 3.15 Diagrama UML de la implementación del modelo Singleton

Con esta implementación, el siguiente paso es manejar el envío de mensajes hacia la raspberry por medio del canal de comunicación bluetooth que ahora se encuentra abierto. Para esto se crea una clase que será la encargada de manejar el envío y recepción de mensajes en subprocesos dentro de la aplicación. Instancias de esta clase serán ejecutadas periódicamente llevando a cabo una suerte de “máquina de estados” que determinara el comportamiento de la aplicación de acuerdo a la respuesta recibida desde la raspberry.

El manejo de la comunicación bluetooth en la raspberry se lleva a cabo en paralelo a la detección de rostro. Ésta es implementada en un hilo para no interferir con el procesamiento de la detección de rostros. Utiliza un “BluetoothSocket” con modo de comunicación “RFCOMM” como servidor, mediante el cual se abre la conexión, esperando que otro dispositivo se conecte, como se puede observar en la figura 3.16. Una vez conectado el dispositivo, el canal de comunicación se encuentra abierto para recibir los mensajes, y a la vez enviar la respuesta de vuelta hacia la aplicación; completando así la suerte de máquina de estados implementada en la aplicación.

```
def run(self):
    print "Starting BT Communication"
    threadLock.acquire();
    #Bluetooth
    server_sock = bluetooth.BluetoothSocket( bluetooth.RFCOMM )
    server_sock.bind(("", 1))
    server_sock.listen(1)
    port = server_sock.getsockname()[1]

    x=0
    global nfaces
    while x < 3:
        print "waiting for conection at port %d" % port
        client_sock, client_info = server_sock.accept()
        print "Accepted conection from ", client_info
```

Figura 3.16 Implementación en Python del BluetoothSocket en la Raspberry Pi 3.

Dependiendo del tipo de mensaje recibido, la raspberry se comportará de cierta manera, se presenta a continuación el detalle de los posibles mensajes y sus respuestas.

- “hasFace”: Este mensaje pregunta a la raspberry si ha detectado un rostro frente al muñeco durante cierto periodo de tiempo; sus

posibles respuestas son “true!” para la respuesta afirmativa, y “false!” para la respuesta negativa.

- “findFace”: Este mensaje significa que en un mensaje anterior de “hasFace” recibió una respuesta negativa, por lo tanto, indica a la raspberry que mande señales hacia el motor de pasos para que ejecute una serie de movimientos, simulando un movimiento de paneo de búsqueda, para así tratar de encontrar un rostro; y una vez encontrado envía el mensaje de respuesta “found!”.
- “endGame”: Este mensaje es un mensaje de control, el cual se envía cuando la actividad de juego es terminada. Cuando este mensaje es recibido, se cierra el canal de comunicación actual y se abre uno nuevo para la siguiente sesión de juego. Envía como respuesta el mensaje “ended!”.

3.1.10 Mejoras en la interfaz

Para mostrar de forma llamativa la actividad de juego en el modo de 1 jugador, se mejoró la interfaz de la misma, ya que hasta ahora solo se enfocaba en la funcionalidad, pero no era llamativa para niños. Se cambió la imagen de fondo de la pantalla de juego por uno que se asemeja a una pizarra de tiza, fue reemplazada la tipografía en la que son presentados los retos matemáticos por una que se asemeja a letras y números trazados en tiza sobre una pizarra, el número de intentos restantes ahora son representados por globos ubicados en la parte superior derecha y estos poseen animaciones que asemejan el rompimiento del globo. La nueva interfaz se puede observar en la figura 3.17.

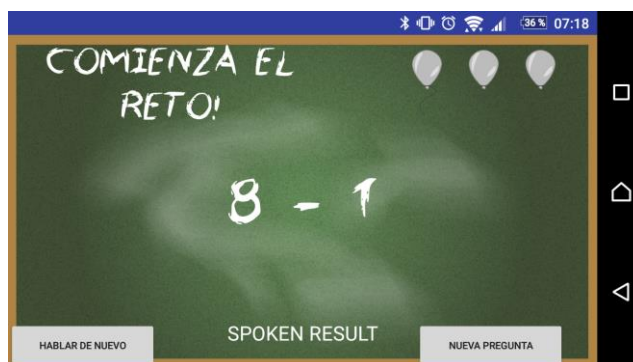


Figura 3.17 Diseño final de la pantalla de juego de 1 jugador.

Por último, la pantalla inicial de la aplicación fue modificada, presentando ahora el nombre de la aplicación y 2 botones en forma de globo que representan los modos de juego de 1 jugador y 2 jugadores. La figura 3.18 muestra la nueva pantalla de inicio.

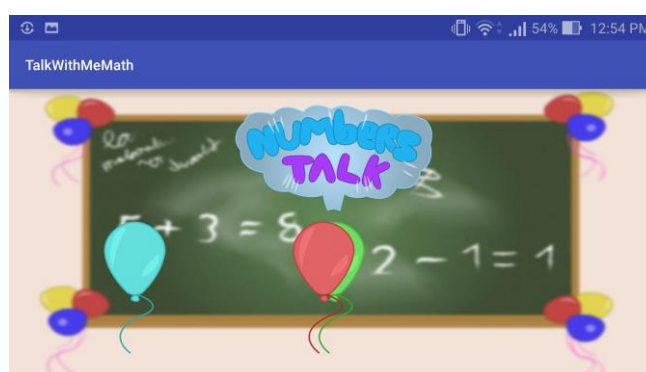


Figura 3.18 Pantalla de inicio de la aplicación.

3.2 Módulo 2 Jugadores

El módulo de 2 jugadores tiene como objetivo ayudar al aprendizaje del niño en matemáticas por medio de competencias fuera del aula de clase. Para esto el participante cuenta con una opción dentro de la Aplicación que le permitirá comunicarse con un servidor desarrollado con Node JS para enfrentarse a otro niño en retos matemáticos.

Para aumentar el rendimiento del niño en matemáticas, este se enfrentará a otro usuario en una competencia en la que tendrán que responder a una operación matemática dentro de un tiempo limitado. Estas operaciones al igual

que en el módulo 1 Jugador se dividen en sumas, restas y multiplicaciones. Al aparecer la operación, también se mostrarán 3 diferentes opciones donde solo una es la correcta. A pesar de que el tiempo es limitado, el simple hecho de que el niño conteste correctamente, le otorga todos los puntos. La razón de esto es no desmotivar al niño, puesto que no darle una recompensa a pesar de haber contestado bien, puede confundirlo y si se encuentra perdiendo constantemente en los retos, perderá completamente el interés en jugar. Para poder mantener el objetivo de la aplicación, el cual es mejorar el conocimiento e interés de los niños hacia las matemáticas, determinar quien responde primero no será parte de las reglas del juego.

Como se ha mencionado anteriormente, los niños necesitan de estimulación visual para capturar su atención y está claro que al ser las matemáticas una materia para muchos de gran dificultad, esta no tiende a capturar el interés ni la atención de los niños. Es por esto que se ha utilizado material didáctico que no solo servirá para capturar la atención del niño, sino que, además colocará un objetivo adicional al juego que permitirá al niño estar más tiempo aprendiendo matemáticas. Se ha incluido un sistema de entrenamiento de criaturas, las cuales acumularán puntos de experiencia que los hará aumentar de nivel y una vez que lleguen a cierto nivel, cambiarán de forma. De esta manera el niño si no juega por competir en los retos matemáticos, jugará para descubrir todas las formas de cada una de las criaturas.

Cada partida contará con tres rondas donde en cada una el niño tendrá 12 segundos para responder. Durante las rondas, ambas criaturas serán presentadas a ambos usuarios, estas estarán animadas y reaccionarán cambiando de animación cuando el niño al que corresponde la criatura ha dado su respuesta. Una vez que ambos han respondido o el tiempo se ha acabado, los puntos serán repartidos y la animación de la criatura cambiará a una alegre si su niño ganó, o una triste si su niño perdió. Al concluir las 3 rondas, el niño con más puntos es declarado el ganador y solo su criatura recibe puntos de experiencia equivalentes a los puntos acumulados por el niño en las rondas; como se puede observar en la figura 3.19. Esto último con la

intención de que el niño tome en serio los retos matemáticos y no juegue solo por darle experiencia a la criatura, pues hemos hecho que ambos usuarios puedan responder correctamente dentro del tiempo definido con la finalidad de que ese sea el enfoque del juego, responder bien, no rápido ni al azar.



Figura 3.19 Pantalla de resultado de sesión de juego multijugador.

3.2.1 El Servidor

Una de las grandes ventajas de Node JS es que no necesita de algún servidor como Apache para que dispositivos puedan acceder a las funcionalidades de los servicios online, a diferencia por ejemplo de PHP. Otra de las grandes ventajas es que cuenta con un manejador de paquetes llamado npm que le permite a la comunidad desarrollar un infinito número de librerías que pueden ser accedidas directamente e instantáneamente por medio del mismo. Esto nos permite desarrollar un servidor de manera ágil y eficiente sin tener que reinventar la rueda.

Debido a que el modo 2 jugadores requiere una conexión en tiempo real entre 2 niños, es necesario implementar una comunicación que nos permita mantener a los dispositivos sincronizados. La forma más viable para hacer esto posible es utilizando conexión por medio de websockets utilizando Socket.io, una librería realizada en Javascript para aplicaciones

en tiempo real. Android también cuenta con una versión de Socket.io realizada en Java cuya manipulación es casi idéntica.

Esta comunicación se basa al igual que http, en un servidor y múltiples clientes, sin embargo, no usa precisamente la arquitectura cliente-servidor. En una arquitectura cliente-servidor, el cliente solicita y el servidor responde, mientras que en una aplicación en tiempo real el cliente y el servidor están comunicándose constantemente sin que haya alguien que necesariamente realice una solicitud. Para esta comunicación utilizaremos el puerto 9090, un puerto que es muy utilizado en ejemplos y APIs debido a su disponibilidad en cualquier sistema operativo.

Socket.io permite mantener conexiones activas con múltiples usuarios y permite comunicarnos con ellos por medio de eventos. Estos eventos se ejecutarán tanto del lado del cliente como del lado del servidor y dependerá de cuando uno de ellos decida disparar un evento del otro. También es posible crear cuartos o canales donde podemos introducir clientes para que la comunicación se limite a usuarios de esos canales con el servidor. De esta manera las partidas se convierten en canales donde solo permitiremos la comunicación entre 2 jugadores y el servidor.

3.2.2 Estructuras Dentro del Servidor

Antes de establecer una conexión en el servidor, es necesario tomar en cuenta las estructuras o clases que se instanciarán a medida que jugadores se conectan al servidor. Las dos estructuras que han sido construidas son “Jugador” y “Partida”.

El jugador es un objeto y contará con la estructura que se muestra en la figura 3.20.

```
1 player = {
2     points : int,
3     id : string,
4     playingRoom : string,
5     monsterId : int,
6     name : string,
7     answered : boolean,
8     readyRound : boolean
9 }
```

Figura 3.20 Estructura del objeto Jugador.

En base a las mecánicas del juego, es necesario tener los puntos acumulados por el niño en la ronda y la criatura que ha elegido, la cual será identificada con un índice. De igual manera es necesario saber cuándo un niño ha respondido en la ronda actual y cuando un niño está listo para la siguiente ronda. La propiedad “playingRoom” permitirá identificar la partida donde se encuentran jugando el niño y su contrincante, así como los datos de la misma. Por último, se tiene el id del usuario que es su identificador para realizar vinculaciones en el servidor y el nombre que es solo para mostrar.

La partida también es un objeto que contiene la estructura mostrada en la figura 3.21.

```
1 match = {
2     round : int,
3     players : [id1(string), id2(string)],
4     roundInfo : {
5         expression : string,
6         result : int,
7         option1 : int,
8         option2 : int,
9         option3 : int
10    },
11     id1 : int,
12     id2 : int
13 }
```

Figura 3.21 Estructura del objeto Partida.

La estructura de la partida cuenta con un indicador de la ronda actual, los identificadores de los 2 jugadores de la partida y la información respectiva de la ronda, la cual es la expresión matemática a resolver, el resultado y las 3 opciones que presentarán las aplicaciones móviles a los niños. Además, aquellas propiedades id1 e id2 no poseen los valores de los indicadores, si no los valores de las opciones que eligió cada niño para que, una vez que ambos han respondido, el servidor pueda determinar si cada niño respondió bien o mal y pueda otorgarle la cantidad de puntos correcta.

3.2.3 Eventos

A continuación se detallan los eventos que forman parte de la comunicación entre dispositivos y servidor para analizar como logran que la sincronización entre clientes y servidor sea exitosa.

Por parte del servidor se presentan los siguientes eventos:

- **Connect:** Evento que será llamado cuando un cliente establece conexión con el servidor de socket. Este evento añadirá el resto de eventos al cliente para que puedan ser disparados desde la aplicación de Android.
- **Disconnect:** Evento que será llamado cuando el cliente se desconecte o cuando el servidor pierda conexión con el cliente. Borrará toda información local sobre el cliente y si él se encontraba en una partida, le avisará al usuario restante que ha ganado por descalificación.
- **Player Data:** Evento que recibirá los datos del usuario cuando se conecta por primera vez, como el nombre y la criatura que ha elegido para posteriormente añadirlo a la cola de emparejamiento de usuario para las partidas.
- **Answer:** Evento que recibirá la respuesta del usuario para la ronda actual y hará los cálculos pertinentes para asignarle puntos al

usuario. Una vez que este evento se dispare para ambos usuarios, la ronda concluirá.

- Ready Round: Evento que será disparado cuando el jugador está listo para la siguiente ronda. Cuando ambos jugadores disparen este evento, el servidor generará la nueva ronda o calculará los resultados finales de la partida y enviará la información correspondiente a los usuarios.
- Play Again: Evento que recibe el servidor para agregar nuevamente al usuario a la cola de emparejamiento para la generación de partidas.

Al ser la comunicación de dos vías, se presentan los eventos del cliente:

- Connect: Evento que se ejecutará cuando la conexión con el servidor es exitosa. Este evento se encargará de emitir la información del usuario al servidor.
- Error: Si llega a ocurrir un error durante la conexión, este evento se disparará.
- Start: Este evento será disparado por el servidor para que la ronda comience para cada usuario. Recibirá la información completa de la ronda.
- Other Answer: Este evento será disparado por el servidor al usuario que no haya respondido cuando el servidor recibe respuesta de un usuario. De esta manera la aplicación puede indicarle al niño que el otro jugador ya respondió.
- End Round: Este evento recibirá los resultados de la ronda cuando esta finalice.
- End Match: Este evento será disparado por el servidor y recibirá los resultados finales de la partida para poder presentarle al niño el ganador.
- Other Disconnect: Este evento será disparado por el servidor cuando un usuario se desconecta durante una partida. El usuario

que aún se mantiene en la partida recibirá los puntos que ha acumulado hasta ese momento y ganará.

Entre los eventos previamente mencionados hay una conexión, un flujo con respecto al juego que está siendo desarrollado para los niños. Para ver claramente cómo funciona la comunicación por sockets de nuestro proyecto y como trabajan los eventos, se presenta el diagrama de flujo de eventos en la figura 3.22.

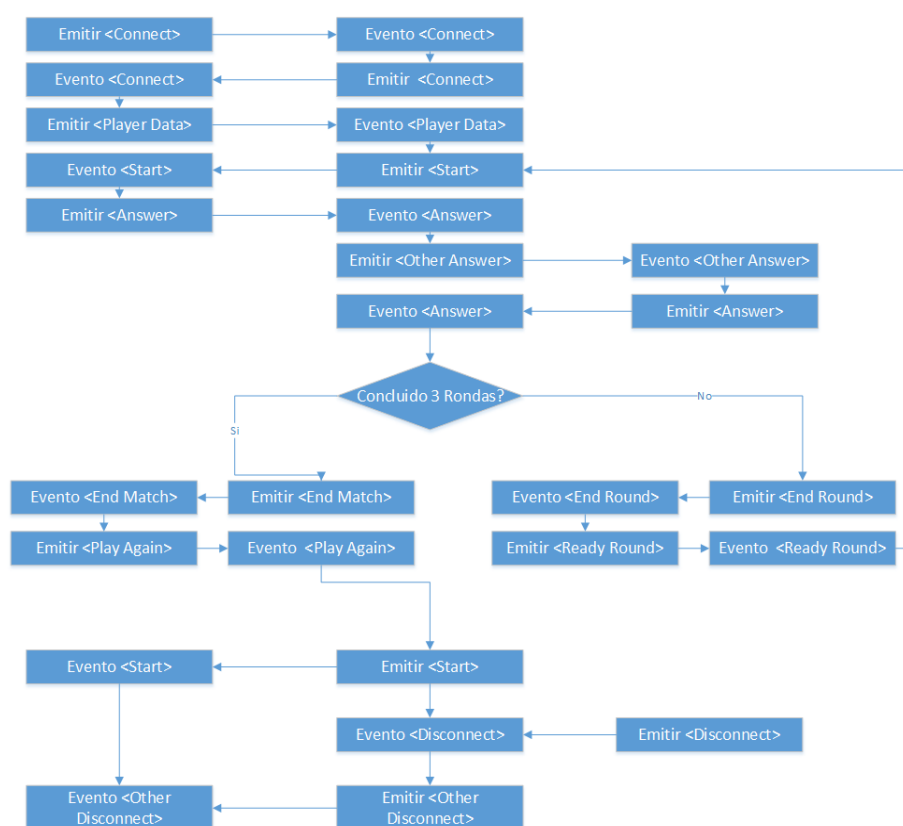


Figura 3.22 Diagrama de comunicación y eventos.

3.2.4 Estructura Principal del Juego

A diferencia del modo de 1 Jugador, el modo de 2 jugadores tiene la atención completa del niño, por esto las dimensiones tanto de las criaturas como las dimensiones de los elementos y su distribución en la pantalla tienen que ser lo más invariantes posibles entre dispositivos. Como se presentarán múltiples elementos en escena, tener un Grid Layout o un

Relative Layout en Android no presentaría el resultado deseado, por suerte contamos con una clase llamada SurfaceView que nos permitirá trabajar sobre un canvas que podemos manipular utilizando coordenadas en el plano xy. Como esto va a ser un juego, no pueden faltar animaciones y efectos que le den a entender al niño que el juego funciona y que esta interactuando con él.

Para poder controlar todos estos eventos, es necesario que la escena que va a ser colocada en el canvas este continuamente refrescándose sin interrumpir las acciones que por debajo realiza la aplicación, como la comunicación por sockets usando eventos. Se ha decidido que el canvas se refresque 30 cuadros por segundo. Esta decisión fue tomada por 2 razones:

- El sistema de visión humano percibe máximo 25 cuadros por segundo: aunque no es totalmente cierto y a que estamos comparando sensores analógicos como los del ojo humano con medidas digitales, es una buena referencia para buscar que los dispositivos alcancen 30 cuadros.
- Si se intenta que el número de cuadros por segundo sea mayor a 30, es probable que muchos dispositivos no logren alcanzar esta cifra y los niños con dispositivos más poderosos tendrán mayor ventaja debido a que si el dispositivo de otro niño es muy lento, puede que no reciba una retroalimentación rápida, o que, al intentar responder, la respuesta no sea seleccionada y enviada antes de que acabe el tiempo de la ronda.

El canvas estará actualizándose dentro de un lazo infinito el cual permitirá obtener el intervalo de tiempo en el cual se ejecuta el refrescamiento del canvas y de esta manera poder controlar el número de cuadros por segundos que se pinta el mismo. Si dicho tiempo es menor a 0.033 segundos, es necesario mandar a pausar la ejecución del lazo la cantidad de tiempo restante para completar los 0.033 segundos. Por otro lado si el

tiempo es mayor, ocurrirá el problema que fue mencionado anteriormente con respecto a la ventaja de un niño sobre otro.

Una vez configurado el canvas es necesario asegurarse de que no esté interrumpiendo los servicios que ocurren en la aplicación y vice-versa. Para esto hemos implementado el SurfaceView dentro de un hilo que se va a encargar de refrescar el canvas y recibir los eventos de este, como el touch, de manera paralela.

3.2.5 Estructuras del Cliente

Ahora que está listo el lugar donde interactuará el niño, se necesita aquello que interactuará con el niño: los jugadores. Recordemos que el niño podrá elegir una criatura a su gusto y utilizarla para enfrentarse a otro niño y su criatura. Estas criaturas aparecerán sobre el canvas junto con el reto matemático y son las que harán de representantes del jugador y necesitan una estructura con la que serán instanciadas en el canvas.

```
public abstract class GameObject {  
    protected int x;  
    protected int y;  
    protected int dy;  
    protected int dx;  
    protected int width;  
    protected int height;  
}
```

Figura 3.23 Clase abstracta GameObject.

Contamos con una clase abstracta llamada GameObject, como se observa en la figura 3.23. Una clase abstracta es una clase que posee atributos y funciones pero que no puede ser instanciada dentro del código, es decir, que no es posible crear un elemento GameObject en la escena. Esta clase sirve como base para ir armando la clase que tomará la forma de la criatura, pues cuenta con elementos básicos como posición x, y, alto y ancho.

```
public class Player extends GameObject{
    private Bitmap spritesheet;
    private int score;
    private Animation animationIdle = new Animation();
    private Animation animationSuccess = new Animation();
    private Animation animationFailure = new Animation();
    private Animation animationAnswer = new Animation();
    private Animation currentAnimation = null;
    private long startTime;
    private int exp;
    private int level;
```

Figura 3.24 Estructura de la clase Player

Como se observa en la figura 3.24, se construye la clase Player que heredará de la clase GameObject para adquirir las propiedades descritas anteriormente y que permitirá adicionarle propiedades exclusivas para la criatura. Las propiedades más importantes a observar son aquellas que servirán para las animaciones, pues es la mayor interacción que tendrá el niño con las criaturas. Tenemos 4 diferentes tipos de animaciones:

- Idle: Cuando la criatura se encuentra en reposo o no realiza ninguna actividad en particular
- Success: Cuando la criatura gana la ronda o partida, realizará una pose de victoria.
- Failure: Cuando la criatura pierda, ya sea en una ronda o en la partida, realizará una pose de derrota.
- Answer: Cuando un niño responde el reto matemático, su criatura correspondiente realizará una pose de ventaja, indicando que su niño ha respondido al reto.

Las Animaciones tienen su propia clase, la cual se observa en la figura 3.25.

```
public class Animation {  
    private Bitmap[] frames;  
    private int currentFrame;  
    private long startTime;  
    private long delay;  
    private boolean playedOnce;  
    private boolean playOnce;
```

Figura 3.25 Estructura de la clase Animation.

Las propiedades de la clase son los cuadros para cada movimiento, el cuadro que se está mostrando al pintar los elementos en el canvas, e indicadores como: cada cuantos cuadros de juego debo pasar al siguiente cuadro de la animación o si debo repetir la animación en forma de lazo o solo ejecutarla una vez.

3.2.6 Animaciones

Si revisamos la clase animación, es posible darse cuenta que una de sus propiedades es un arreglo de bitmaps, donde bitmap es una imagen representada por bits. Así mismo es posible ver que en la clase Player tenemos una propiedad llamada spritesheet. Lo interesante de las animaciones en el proyecto es que cada cuadro de la animación es creado a partir de una imagen que contiene cada movimiento de la criatura (spritesheet).

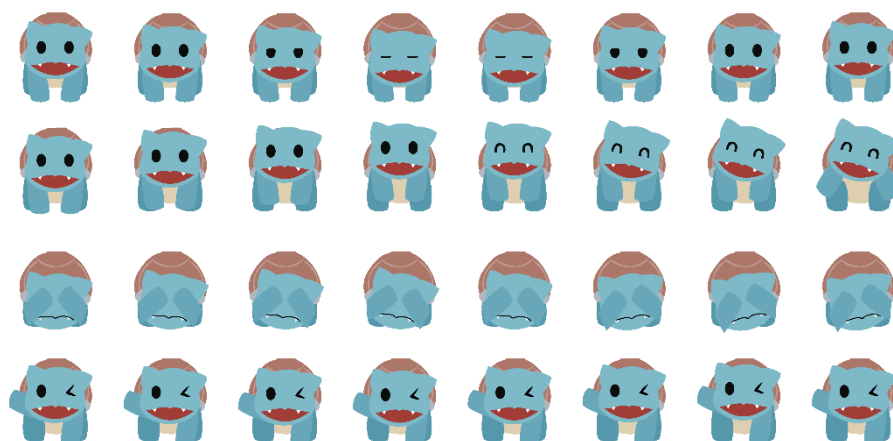


Figura 3.26 Spritesheet de animación de personajes.

Como se puede observar en la figura 3.26, en la imagen se encuentran cada uno de los cuadros de movimiento para cada una de las animaciones que tiene la criatura. La aplicación lee dicha imagen y la va separando en imágenes más pequeñas que posteriormente pasan a ser almacenadas en arreglos con la secuencia correcta de imágenes para generar cada una de las animaciones. De esta manera la clase Animation puede ir actualizando el cuadro actual de la animación de manera sucesiva para dar el efecto deseado.

3.2.7 Lazo del Juego

El lazo del juego es aquel que refresca el canvas cada 0.033 segundos y en el cual se presenta la pantalla principal del juego multijugador, como se observa en la figura 3.27. Además, cuenta con diferentes etapas según el estado del juego.

- Estado de Conexión: un estado que ocurre sumamente breve debido a que la conexión con el servidor es casi instantánea.
- Espera en Cola: le presentará al usuario un mensaje indicando que se está buscando una partida. Este estado se mantendrá hasta que en el servidor el usuario se pueda emparejar con otro para comenzar el juego y pasar al siguiente estado.
- Estado de Juego: es aquel en el que los usuarios se enfrentarán en los retos matemáticos hasta que ambos respondan o se acabe el tiempo. Al terminar una ronda pasarán al estado de resultados parciales.

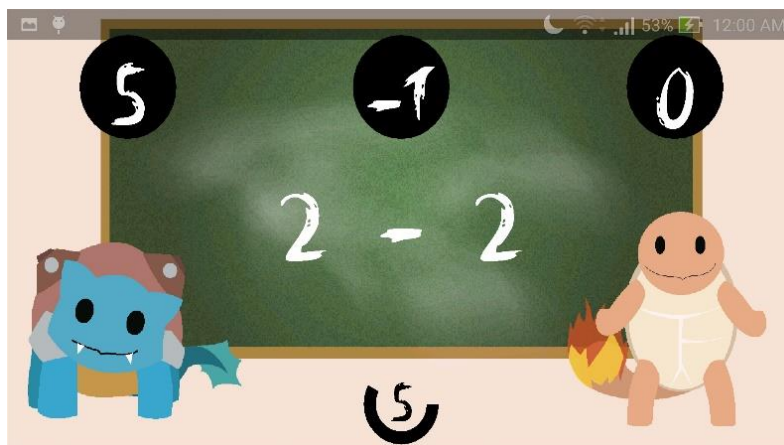


Figura 3.27 Pantalla principal de sesión de juego de 2 jugadores.

- Resultados Parciales: Se presentará al ganador y se cambiarán las animaciones de las criaturas.
- Resultados Finales: Cuando las 3 rondas finalicen, pasaremos al estado de resultados finales donde se mostrará al ganador y se repartirá la experiencia correspondiente para la criatura del vencedor. Si la criatura suma suficiente experiencia y alcanza cierto nivel, el estado cambia al estado de evolución.
- Estado de Evolución: La criatura se toma un tiempo para cambiar de forma una vez alcanzado determinado nivel.

Al ser esto un lazo donde vamos cambiando de estados, finalmente regresa al estado de espera.

3.2.8 Manejo de las Criaturas

Una vez descrito el desarrollo del modo 2 jugadores, es posible hablar sobre la selección de criaturas. La selección parte de mostrarle las diferentes criaturas al niño para que este escoja una e inicie la partida con la misma, como se observa en la figura 3.28. Al iniciar el modo 2 jugadores, obtenemos la criatura y su respectivo nivel y la presentamos en pantalla.

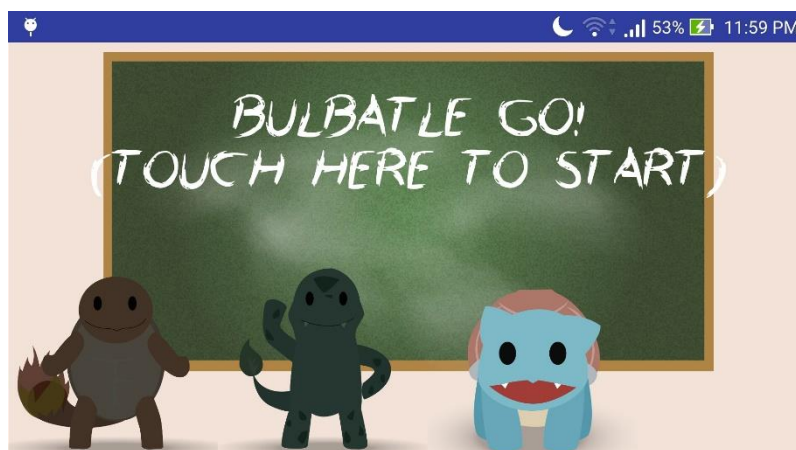


Figura 3.28 Pantalla de selección de criaturas.

La parte importante es como indicarle a la siguiente actividad la criatura seleccionada, su nivel y de igual manera almacenar la experiencia y el nivel de las criaturas a medida que el niño va jugando. En Android contamos con un sistema de archivos local, propio de la aplicación que se lo puede acceder por medio de la clase `SharedPreferences`. Esta clase permite crear archivos con respecto a una cadena de caracteres que servirá como identificador para acceder a un mismo archivo por cadena. Como se trata de archivos, no podemos almacenar estructuras complejas como objetos, por lo que es necesario dividir la información de las criaturas tanto por propiedades como por índices, teniendo un almacenamiento llave/valor de la siguiente manera:

- `monster<indice>level`: el nivel de la criatura según su índice
- `monster<indice>exp`: la experiencia de la criatura según su índice

Así, se obtiene una lista de las criaturas con sus respectivos índices e información constante como el nombre, además de su experiencia y nivel, y al ser estos datos variables, están almacenados en un sistema de archivos local. Como se puede observar, la cantidad de información que es necesaria almacenar es sumamente pequeña, por lo que leer o escribir información resulta más eficiente que manejar una base de datos.

CAPITULO 4

4. PRUEBAS Y RESULTADOS

Una vez completado el desarrollo del sistema, es necesario verificar si esta solución presenta un impacto positivo sobre la problemática a resolver, para esto se realizarán pruebas para cada uno de los modos de juego desarrollados. Así mismo, es necesario probar el desempeño del sistema en un ambiente de la vida real, fuera del ambiente controlado en el que fue desarrollado, ya que esto permitirá identificar posibles fallas en la implementación, y al mismo tiempo obtener una visión real del comportamiento del sistema frente a sus usuarios finales. A continuación, se describen las pruebas realizadas y los resultados obtenidos.

4.1 Pruebas al módulo de 1 Jugador

Para realizar las pruebas del modo de 1 jugador, era necesario llevar la aplicación junto con el muñeco hacia una locación de pruebas que encajara con el modelo de usuarios objetivo, es decir niños de 5 a 10 años de edad. Para esto se contactó con las autoridades directivas de la Unidad Educativa Particular “Jesús de Nazareth”, en la ciudad de Machala, para coordinar visitas al aula de clases del cuarto año de educación básica, y realizar una actividad en clase haciendo uso del sistema desarrollado.

Cabe recalcar que las pruebas se realizaron en esta unidad educativa ya que uno de los miembros del grupo de trabajo realizó sus estudios de primaria en la misma, y gracias a esto se pudieron realizar las reuniones con las autoridades y concretar las visitas de manera más fácil.

En total, se realizaron 2 visitas al aula de clases con los siguientes resultados:

- Jueves, 25 de agosto de 2016: Durante la primera visita al aula de clases, se obtuvieron los siguientes resultados: Luego de una pequeña explicación sobre el funcionamiento del muñeco, los niños mostraron claramente su interés hacia la actividad debido a la novedad del muñeco, demostrando así la efectividad del sistema en capturar la atención de los niños; sin

embargo, tanta fue el entusiasmo que fue necesario el apoyo de la profesora encargada del aula para poder realizar la actividad en orden, esto se puede observar en la figura 4.1.



Figura 4.1 Primer contacto de estudiantes de cuarto año de educación básica con el muñeco.

Luego de recuperar el orden de la clase, la actividad pudo realizarse de forma ordenada. La aplicación funcionó de forma fluida, a pesar del ruido presente en el ambiente, característico de un aula de clases de primaria, sin embargo, fue notoria la falta de protocolos en relación al volumen y velocidad de entonación requerida por los niños para poder garantizar una experiencia de juego fluida. De lo observado en esta prueba, de cada 6 intentos del niño por responder un desafío, solamente 4 respuestas eran detectadas correctamente; 40% de los niños detectaron el movimiento de la cabeza del muñeco, mientras que el 60% restante no lo pudo detectar; además como última observación, se notó que, para niños de primaria, el uso del símbolo asterisco (*) para la multiplicación era desconocido, lo que fue tomado en cuenta para una siguiente revisión.

- Jueves, 1 de septiembre de 2016: Durante la segunda visita al aula de clase, se tomaron en cuenta los resultados obtenidos en la primera visita para corregir varios aspectos de la aplicación y el muñeco. Primero que nada, queda establecido un protocolo para la interacción con el muñeco: los niños tienen que estar ubicados de 50 a 100 cm de distancia del muñeco, y hablar en un tono alto y fluido, se puede observar un ejemplo de la implementación del protocolo en la figura 4.2. El algoritmo de movimiento de la cabeza del muñeco fue ligeramente modificado para

mejorar su movimiento; y por último el cambio del símbolo asterisco (*) de multiplicación por una equis (x) que es el símbolo de multiplicación comúnmente enseñado en educación básica.



Figura 4.2 Interacción de estudiantes con el muñeco siguiendo parámetros establecidos.

Durante esta segunda prueba, se obtuvieron resultados mucho más favorables. El 90% de los estudiantes logro detectar el movimiento de la cabeza del muñeco y detecto el seguimiento de su rostro; Por cada 6 intentos de los niños por responder un reto matemático, todos eran detectados correctamente, siguiendo los parámetros de distancia, volumen y velocidad establecidos al inicio de la actividad.

La duración de ambas visitas fue de máximo 40 min, en este tiempo todos los 28 alumnos del aula pudieron participar en la actividad. Como último punto, después de finalizar ambas visitas, la profesora encargada pudo detectar las falencias de los estudiantes en las tablas de multiplicar de los alumnos, y posterior a la actividad realizaba ejercicios de refuerzo enfocados en las fallas detectadas. Esto demuestra uno de los puntos clave expuestos en la problemática y a la vez demuestra utilidad de la herramienta en un ambiente de la vida real.

4.2 Pruebas al módulo multijugador

Para las pruebas de estrés del servidor, se participó en una feria donde el proyecto fue presentado a alumnos de secundaria de múltiples colegios. Durante todo el día estudiantes se acercaron a jugar el módulo multijugador, generando una gran cantidad de partidas y utilizando diferentes criaturas según lo deseaba el usuario.

Bajo una conexión estable, el servidor se comunicaba rápidamente con los dispositivos móviles y hacía de la competencia una agradable experiencia. Se recibió retroalimentación con respecto a la cantidad de criaturas y muchos estudiantes mencionaron que efectivamente la dificultad del juego era para niños mucho más pequeños, pero no hubo queja alguna con respecto a la conexión con el servidor y el flujo del juego.

Luego de esto, se mantuvo el servidor corriendo durante 1 mes mientras se realizaban ajustes en el modo 2 jugadores y el servidor en ningún momento presentó alguna falla donde hubiera la necesidad de reiniciarlo, incluso cuando la conexión a internet no era buena, esto no presentaba problema para el servidor e incluso si un jugador se desconectaba, el servidor declaraba al otro niño como ganador.

La prueba con niños se realizó principalmente con familiares, donde se identificaron problemas de interacción, como por ejemplo donde tocar para continuar luego de haber seleccionado una criatura o que hacer luego de que finaliza una partida. También se tuvo la oportunidad de probar la aplicación con una niña de 8 años, quien, al comenzar a jugar, no quería soltar el dispositivo a pesar de que su madre le insistía en que se tenían que ir, incluso la niña logró ganar suficiente experiencia con una criatura como para hacer que esta cambiará de forma, esto se puede observar en la figura 4.3.

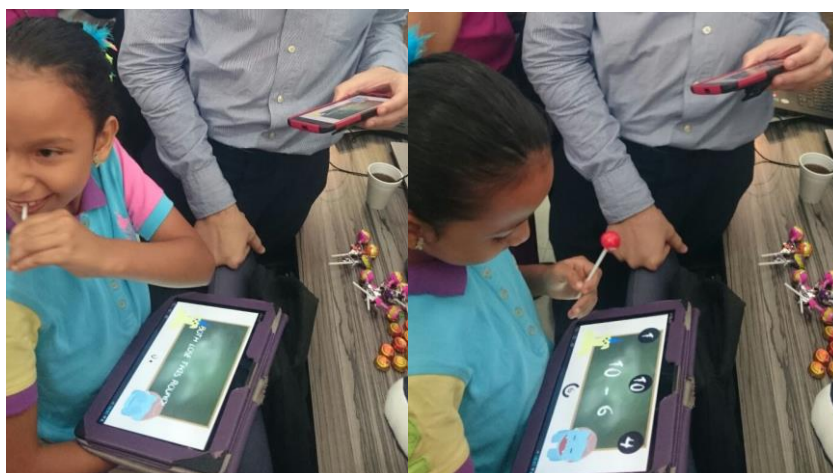


Figura 4.3 Interacción de una niña de 8 años con la aplicación durante la Expo Materia Integradora 2016.

Con respecto a los retos matemáticos, las operaciones de suma y resta eran aquellas que para los niños eran las más fáciles, siendo las multiplicaciones lo más complicado y que no siempre respondían correctamente, sin embargo, a medida que jugaban, se iban familiarizando más con estas operaciones y los errores se reducían. Esto es un buen indicador de que por medio de la aplicación los niños podrán reforzar sus conocimientos en matemáticas de una forma divertida.

CONCLUSIONES Y RECOMENDACIONES

Gracias a las pruebas realizadas, se pudo comprobar que el sistema logra captar de manera efectiva la atención de niños y niñas hacia el refuerzo de destrezas matemáticas, mediante las varias características que posee este recurso. Con esto se logra impulsar el aprendizaje activo en los niños.

El uso de este recurso en el aula de clase, probó ser una herramienta efectiva que ayuda al maestro a identificar áreas o tópicos que necesitan refuerzo en sus alumnos, de esta manera ayudando al maestro a llevar un currículo de enseñanza que se adapte mejor a las necesidades de sus alumnos.

El modo de juego de 2 jugadores probó ser una herramienta llamativa que motiva al jugador a seguir el proceso de aprendizaje fuera del aula de clases, enfocándose en el aprendizaje cooperativo mediante la competencia y llegando tener bastante enganche con los niños.

El elemento que captó mayor atención en el modo de juego de un jugador fue el muñeco, ya que este funciona como puente de interacción entre el niño y la aplicación. Mientras que, en el modo de 2 jugadores, el elemento que captó la mayor atención fue seleccionar una criatura al inicio y lograr que dicha criatura crezca a medida que se avanza en el juego.

Para garantizar una experiencia de juego fluida dentro del modo de un jugador, es altamente recomendado seguir los siguientes lineamientos: el jugador debe estar ubicado a una distancia de 50 a 100 cm de la base del muñeco, hablar en un tono de voz alto, sin gritar, y de manera fluida.

La incorporación de la detección de rostros en el sistema probó ser una de las características que más llamó la atención de los niños durante la actividad, al aumentar la sensación de inmersión debido la retroalimentación que obtenían mediante el movimiento de la cabeza del muñeco.

El servidor implementado para el modo de juego de 2 jugadores probó ser lo suficientemente robusto para manejar los requerimientos de la aplicación durante sus

primeras fases, como por ejemplo el acceso a este desde redes externas a través de internet.

El uso constante de la fuente de poder del motor de paso, eventualmente provocara a que la carga se consuma; se recomienda cargar la batería de 12v con un adaptador de corriente eléctrica, no electrónico, de 12v a la toma de corriente de pared por alrededor de una hora. La sobrecarga de esta batería puede llevar a daños en la misma que impedirán el correcto funcionamiento del motor de pasos, en ese caso la única solución es reemplazarla por una nueva. Se puede deducir en que momento la batería necesita ser recargada, gracias al voltímetro que se encuentra en la parte posterior del muñeco; cuando el valor del voltaje mostrado sea menor a 8.00v es un buen momento para recargar la batería.

El sistema se desenvuelve muy bien en ambientes con ruido moderado, sin embargo, no todos los dispositivos móviles cuentan con un sistema de amplificación lo suficientemente potente como para garantizar la correcta emisión del reto matemático por medio de la sintetización de voz, por esto es altamente recomendado adaptar al dispositivo móvil un pequeño parlante de cualquier naturaleza, sea bluetooth o a cable, para poder apreciar el sistema a toda capacidad.

Durante la presentación del proyecto en la Expo Materia Integradora 2016, fueron recibidos varios comentarios acerca del proyecto, uno de ellos queda como posible implementación a futuro del mismo, esto es, implementar retos matemáticos que incluyan divisiones, potenciaciones y raíces cuadradas para estudiantes de 10 a 12 años.

BIBLIOGRAFÍA

- [1] P. Jodi et al., "New Technologies in the Classroom" en International Journal for professional educators, Austin, Texas, 2012, pp. 16-22.
- [2] P. O'Mailey et al., "Effectiveness of Using iPads to Build Math Fluency", en Council for Exceptional Children, San Antonio, Texas, 2013, Online Submission, Disponible en: <http://files.eric.ed.gov/fulltext/ED541158.pdf>
- [3] S. Rushton, A. Joula-Rushton, "Classroom Learning Environment, Brain Research and The No Child Left Behind Initiative: 6 years Later" Early Childhood Education Journal, Springer, 2008.
- [4] GreatSchools Staff (2016, Marzo 16). "Math Disability in Children: an Overview". [Online]. Disponible en <http://www.greatschools.org/gk/articles/math-disability-in-children-an-overview/>
- [5] A. Guadamud. (2014, Abril 21). Los problemas de aprendizaje más comunes en niños y adolescentes. [Online]. Disponible en: <https://www.institutoneurociencias.med.ec/categorias-guia/item/867-problemas-aprendizaje>
- [6] S. Nusir et al., "Studying the impact of using multimedia interactive programs on children's ability to learn basic math skills", Acta Didactica Napocensia, vol. 5, no. 2, pp. 17-32, Mayo, 2012
- [7] INEC (2013, Diciembre 31), "Ecuador en cifras – Tecnologías de la información y comunicación (TIC's) 2013" [Online], Disponible en: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/Resultados_principales_140515.Tic.pdf
- [8] L. Shams & A.R. Seitz, "Benefits of multisensory learning", "Trends in cognitive sciences", vol. 12, no. 11, p. 411-417, Noviembre, 2008.
- [9] G. Edgar & M. Duval, "Aplicación para el desarrollo de destrezas matemáticas para un grupo de personas", Informe de Materia Integradora, FIEC, ESPOL, Guayaquil, Ecuador, 2015.

[10] P. Viola & M. J. Jones. "Robust real-time face detection", "International journal of computer vision", vol. 57, no. 2, p. 137-154, Mayo, 2004.

ANEXOS

Evidencias de las pruebas realizadas en la Unidad educativa “Jesús de Nazareth”.







Foto del muñeco ensamblado

