



CONSTRUCCIÓN DE MATE-MOVIL A TRAVÉS DE LA METODOLOGÍA MEISE

Eje temático: Mobile Learning

Laura Nelly Parra Hernández

IT de Lázaro Cárdenas, México; lauranelly.parra@gmail.com

Giovanni Sinaí Silva Barragán

IT de Lázaro Cárdenas, México; giovannigatfut@gmail.com

Resumen

En cualquier proyecto de software, el uso de una metodología es importante ya que las fases de esta misma proporcionan el camino a seguir para poder llegar a la obtención de un producto de calidad, a su vez estas fases generan una lista de documentos entregables que constatan el avance y la naturaleza del proyecto e incluso en el análisis decir si el proyecto es viable o no y así poder planificar el desarrollo del producto que se desea obtener.

En un software educativo, una metodología toma un papel aún más importante, ya que además de cumplir con los objetivos del proyecto, el contenido del producto debe ayudar a cubrir las competencias necesarias del nivel al que está destinado.

Palabras clave: MeISE, software educativo, matemáticas, aprendizaje, competencias, TPACK.

Introducción

En los últimos años la tecnología ha avanzado a pasos acelerados, facilitando a las personas el acceso al uso de diferentes dispositivos que les ayudan para poder realizar tareas diarias, pero se han descuidado varios aspectos que son de vital importancia, como lo es la educación, que como se puede ver en este país, sufre de una considerable deficiencia, que es dada por muchos factores, además que los métodos de enseñanza-aprendizaje, están en proceso de obsolescencia, lo que provoca una falta de atención o tedio al momento de recibir una sesión de clase, ya que los niños de hoy en día hacen uso de la tecnología a muy temprana edad, entre los usos principales de la tecnología entre los niños están los videojuegos, ya que estos son intuitivos y

están al alcance de todos, ya que se encuentran principalmente concentrados en una tienda de aplicaciones y son fáciles de encontrar y obtener.

Mate-Movil es una herramienta que tiene como objetivo, mejorar la comprensión de las matemáticas en los temas de multiplicación, división y reconocimiento de figuras geométricas. Al niño se le presenta en forma de videojuego, pero lo que hay de trasfondo, es un sistema que evalúa al niño, por medio de competencias, ya que los contenidos del videojuego son estructurados de acuerdo a las competencias que propone la SEP (secretaría de educación pública). Pero también puede llegar a emplearse el uso de Mate-Movil como una herramienta que ayuda al profesor en su desempeño como docente, y aunado a la ayuda que el docente pueda dar al niño para la resolución de algunos problemas, puede también ser una herramienta que ayude al docente a poder incorporar el uso de la tecnología dentro del aula y que por ende también aporta otra forma de aprender creando una atmosfera de aprendiendo-jugando, lo cual resulta bastante atractivo para el niño.

Antecedentes

Serrano en su artículo cita a Jean Piaget quien define que: El aprendizaje es, por tanto, un proceso interno que consiste en relacionar la nueva información con las representaciones preexistentes, lo que da lugar a la revisión, modificación, reorganización y diferenciación de esas representaciones (Serrano González-Tejero & Pons Parra, 2011, p. 6). Esta definición es la más primitiva, en el momento que empezamos a hablar de educación y este mismo concepto, es la base para poder comprender el objetivo primordial del software educativo, el aprendizaje por medio de una herramienta tecnológica, que aporta otro enfoque a la manera tradicional de aprender.

Según López en su artículo estilo cognitivo y logro académico define: el estilo cognitivo puede ser entendido como el modo habitual o típico que una persona tiene para resolver problemas, pensar, percibir y recordar (López-Vargas, Hederich-Martínez, & Camargo-Uribe, 2011, p. 69). Como es muy bien conocido, todos tenemos una manera propia de resolver problemas, pero todos coincidimos en una respuesta, esto hace que el software tenga una responsabilidad de realmente poder hacerle saber al usuario que lo que hizo, o la manera en que resolvió determinado problema, le fue útil y que puede adoptar esa forma para resolver ese determinado tipo de ejercicio.

Y por último como concepto básico Bermúdez define competencia como: el buen desempeño en contextos diversos y auténticos basado en la integración y activación de conocimientos, normas, técnicas, procedimientos, habilidades y destrezas, actitudes y valores. (Bermúdez et al., 2011, p. 170).

La Metodología de Ingeniería de Software Educativo (MeISE) tiene un ciclo de vida que se divide en dos etapas: la etapa de definición y la etapa de desarrollo. En la primera etapa se determinan los requisitos, análisis y diseño preliminar, también se especifican

de forma general los objetivos o a lo que se quiere llegar con el software, y se termina con un plan de iteraciones cuidando que el producto que se libera que en cada plan está didácticamente completo.

Después inicia la segunda etapa en la que se comienza a desarrollar el software, de modo que el equipo de desarrollo toma cada iteración, la diseña, construye, prueba e implementa, haciendo una evaluación al final si es factible seguir con más iteraciones hasta obtener un producto completo.

(Abud Figueroa, Ma. Antonieta, 2009, p. 2) afirma que las fases que propone MeISE son las siguientes:

(...) la fase conceptual, durante la cual se identifican los requerimiento del sistema, se conforma el equipo de trabajo y se elabora el plan de desarrollo; la fase de análisis y diseño inicial, en la que se propone la arquitectura que servirá de base para la solución del problema y se establecen las características pedagógicas y de comunicación que regirán el desarrollo del software; finalmente la fase de plan de iteraciones, en la cual se divide el proyecto en partes funcionales que permitan mejor control en su desarrollo. En la etapa de desarrollo se tienen: la fase de diseño computacional, en la que se realizará un diseño computacional detallado de un incremento específico del software; la fase de desarrollo, durante la cual se implementa la arquitectura en forma incremental (iteración por iteración); y la fase de despliegue, donde se realiza la transición del producto ejecutable al usuario final.

El ciclo de vida del método de desarrollo MeISE se ilustra en la figura 1.

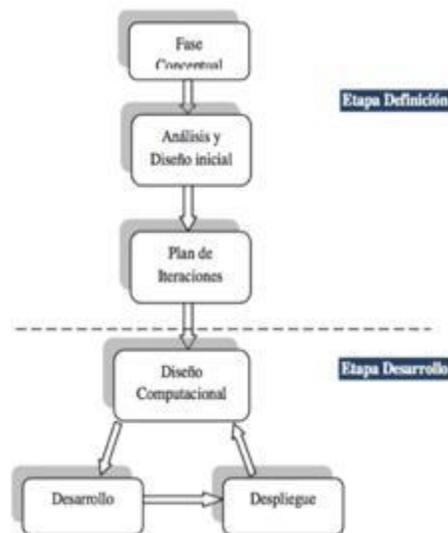


Figura 1. Ciclo de vida de la metodología.

Fuente: (Abud Figueroa, Ma. Antonieta, 2009, p. 2)

El modelo tradicional basado en la simple transmisión de información desde el maestro al estudiante ha comenzado a mostrar la ineficiencia para el desarrollo del aprendizaje, por lo que se requiere un cambio de rol en el maestro. Además, las tecnologías de la información y comunicación (TIC) han alcanzado cada rincón de la vida diaria,

suficiente para justificar el uso de las TIC en un salón de clases y es inevitable ignorar que esto puede existir en la sociedad.

El nuevo paradigma educacional necesita incorporar nuevas habilidades, capacidades y recursos tecnológicos, los cuales harán más fácil para los estudiantes adquirir competencias básicas. El modelo TPACK (Technological, Pedagogical Content Knowledge) identifica el conocimiento específico que los profesores necesitan tener para que la integración de este exista. De acuerdo con este modelo, una adecuada utilización de la tecnología en el proceso de enseñanza requiere un tipo de formación de profesores basado en los diferentes tipos de conocimiento, el cual puede resumir en la idea de poder utilizar una efectiva metodología para la implementación de las TIC que apoyen estrategias pedagógicas y métodos relacionados a una disciplina específica. Por lo tanto, el modelo TPACK representa el conocimiento de contenido que necesitan los profesores junto con el conocimiento tecnológico y pedagógico con el objetivo de integrar las TIC en el proceso de enseñanza-aprendizaje.

A partir de este modelo, se realizaron las dos encuestas que proporciona TPACK: una aplicada a los maestros, cuyas preguntas van inclinadas hacia una autoevaluación del conocimiento y habilidades tecnológicas con las que cuenta; la otra parte de la encuesta es la percepción que los alumnos tienen hacia su profesor en cuestión de los conocimientos y herramientas que usa su profesor para impartir la clase. Los resultados de estas arrojaron que, los profesores cuentan con habilidades tecnológicas básicas que podrían facilitar la inclusión de las TIC en el aula, pero para los niños los maestros tienen buenos conocimientos sobre los temas que imparten, pero, no incluye ningún tipo de herramienta tecnológica para complementar las clases, encontrando una gran área de oportunidad para la aplicación de TIC en el aula.

Desarrollo de la metodología

Fase conceptual

Análisis de las necesidades educativas

Al realizar encuestas en algunas escuelas se observó que, según los resultados los temas que a los niños más se complican comprender eran los temas de multiplicación, división y reconocimiento de figuras geométricas. Otra de las necesidades de las que se percató, fue la falta del uso de tecnologías en el aula para complementar la clase y hacerla al mismo tiempo más entretenida e interactiva; debido a esto se identifica que el maestro sigue usando una metodología de enseñanza tradicional, lo cual provoca tedio al estudiante.

Los elementos de motivación que se van utilizar en Mate-Movil son imágenes relacionadas con los ejercicios que se muestran en los libros de texto que proporciona la SEP de manera gratuita en México, además de agregar algunos contenidos de guías de apoyo que utilizan los docentes como complemento a los libros de texto, además Mate-Movil contará con música, sonidos e animaciones.

La forma de evaluación que tendrá Mate-Movil, consistirá en evaluaciones conforme las competencias que el niño vaya concretando de manera positiva, de acuerdo en el nivel de profundización en el que se encuentre y así poder pasar a un nivel de profundización superior, hasta concluir con todos los niveles y poder decir que la competencia ha sido alcanzada.

Alternativas de solución

Las opciones que se presentan para desarrollar al software son varias: desarrollar una aplicación web, una aplicación para computadora, una aplicación móvil o un videojuego multiplataforma.

Para el proyecto se elige como más viable la opción de hacer un videojuego educativo multiplataforma ya que, en base a encuestas aplicadas los resultados arrojaron que todos los niños a los que se les aplicó dicha encuesta tenían acceso a un dispositivo móvil (tableta o celular) propio o de algún integrante de su familia, además que también tienen acceso a una computadora, ya sea en su escuela o en su casa.

Dicho videojuego educativo multiplataforma se hará nativa, es decir no necesitará internet para funcionar, esto debido a que estudios estadísticos realizados por el INEGI (Instituto Nacional de Estadística, Geografía e Informática), dice que 46.3 millones de personas (42.6%) en México no usan internet, además nos dice que el 53.9% de usuarios de educación básica no usa internet ((INEGI), 2016, p. 2), y en las encuestas recabadas para este proyecto indican que la mayoría de los niños no cuentan con servicio de internet ni en casa ni en la escuela.

Estudio de riesgos

Algunos de los riesgos que se pudiesen presentar en el desarrollo de la aplicación son:

- Los elementos de motivación (música, sonidos, animaciones, imágenes) pudieran no ser suficientes para los niños.
- Los ejercicios integrados en el videojuego tengan un grado de complejidad más alto o más bajo del que los niños son capaces de responder o no acorde a las competencias que se necesitan.
- El contenido del software carezca de una base bibliográfica que permita cumplir el objetivo del software.
- Que el uso del software se vea obstruido o no sea adoptado por los docentes como una herramienta para fortalecer el aprendizaje de los niños.

Una de las formas para atacar estos riesgos son: el consultar por medio de encuestas y entrevistas a un grupo de niños para conocer cuáles son los elementos de motivación importantes para ellos, por ejemplo: qué tipo de música les gusta, qué tipo de imágenes son sus favoritas, cuáles son sus colores favoritos, entre otros. De esta forma, se tiene una visión generalizada del diseño que el software tiene que alcanzar para que llegue a

un alto grado de aceptación y los cambios dentro de las retroalimentaciones posteriores, solo tengan cambios minúsculos o solo sean corrección preventiva de errores.

Para que el nivel de dificultad de los ejercicios sea congruente con las competencias a las que los niños tienen, primeramente, se toma una parte de la bibliografía que actualmente utilizan en las sesiones de clases para tomar estas mismas como referencia del nivel de profundidad promedio de los ejercicios que se presentan dentro de Mate-Movil. Una vez recuperada y estructurada esta información, se elabora una guía o árbol de ejercicios, que contiene el tipo y nivel de profundización de los ejercicios que posteriormente son revisados por un profesor para poder dar validez de que realizará una evaluación de acuerdo a la de los ejercicios de ese nivel de profundidad y tener un nivel de dominio más preciso. Así, se tiene certeza en que el contenido del software es el adecuado para los niños y también que las evaluaciones realizadas están acordes a los contenidos y a los niveles correctos.

Para poder asegurar el uso directo del software, se llega a capacitar a los docentes acerca de la importancia del uso de estas herramientas en el desarrollo cognitivo del aprendizaje, además, poder llevar a incluir las TIC en el aula, haciendo ver primero al docente el nivel de dominio que tiene del uso de estas tecnologías y de entender la necesidad de la inclusión de estas tecnologías dentro de su institución, asegurando que los contenidos son controlados por el software y este realiza evaluaciones reales de acuerdo a lo que propone (Bermúdez et al., 2011, pp. 173–175), lo cual da confiabilidad, en que las evaluaciones arrojadas por el software reflejan realmente el nivel de competencia alcanzado por el alumno.

Funcionalidad a alcanzar

El software proporcionará un nivel de competencia al niño mediante la implementación del algoritmo propuesto por (Bermúdez et al., 2011), el cual nos indica que, para poder llegar a la competencia deseada, el niño tiene que tener un nivel de profundización, en este caso el software se lo proporciona mediante el implemento de tres niveles: Fácil, intermedio y avanzado, esto a su vez se conjunta con una evaluación sumativa, la cual puede dar una certeza más cercana del nivel de dominio que alcanza el niño en la competencia.

Nuestro software puede ser capaz de brindar una herramienta de aprendizaje, que enseñe al niño a poder aprender, de una manera interactiva, lúdica y sobre todo tecnológica, ya que también el proyecto se interesa en la inclusión de las TIC en el aula, por lo que se desea que el software sea adoptado como una herramienta más de aprendizaje del día a día en el aula.



Modelo de aceptación

Para que el modelo pudiese ser aceptado fueron definidas las siguientes características como mínimas:

- Diseño del software en base a los resultados lanzados por las encuestas y/o entrevistas realizadas a los niños.
- Los ejercicios que componen los diferentes niveles del software deben estar basados en bibliografía que los niños utilicen (libro de texto, guías de estudio, exámenes, etc.) y esté de acuerdo a la temática y grado escolar.
- El software debe realizar evaluaciones sumativas de acuerdo al nivel de dificultad de los ejercicios que fue resolviendo.
- El software debe mostrar el nivel de dominio alcanzado en los distintos niveles que se presentan en el mismo, y poder así permitir un acompañamiento y retroalimentación, que de soporte a las evaluaciones que pueda paralelamente realizar el docente en su clase.

Análisis y diseño inicial

Análisis de requisitos

Requisitos del usuario:

1. Contenido ligado a material bibliográfico. El contenido del software educativo (ejercicios y evaluaciones) será de acuerdo al material bibliográfico que usan en las instituciones y de los temas especificados (multiplicación, división y reconocimiento de figuras geométricas), además de presentarlo de forma clara, concisa y de forma lúdica.)
2. Software multiplataforma. El software tiene que desarrollarse en diferentes plataformas, para que los usuarios puedan usarlo en el momento que quieran y puedan disponer del videojuego cuándo lo deseen. Dando así la libertad al usuario de practicar el tiempo que desee en el dispositivo que considere.
3. Evaluación diagnóstica. Evaluación realizada antes de comenzar con el uso del software para medir el nivel de competencia con el que el usuario inicia y posteriormente, haciendo una comparativa entre esta evaluación y la sumativa se determinará el nivel de avance que ha adquirido el usuario. (módulo de administración de evaluaciones)
4. Evaluación sumativa. Evaluación que se realiza al finalizar un nivel y que, según el resultado obtenido desbloquea el siguiente nivel o da un consejo de volver a practicar para superar la evaluación.



5. Niveles de dificultad. El software debe contener diferentes niveles de dificultad para que reten al usuario a continuar usándolo y reforzar o aumentar sus competencias.

Arquitectura del software

Se define una arquitectura de software tipo modular, en la que se diseñan cinco módulos, en los cuales se distribuyen los procesos que se realizan en el software. Estos módulos son el Módulo de Administración de Usuarios (MAU), Módulo de Administración de Evaluaciones (MAE), Módulo de Operaciones de Multiplicación (MOM), Módulo de Operaciones de División (MOD) y el Módulo de Identificación de Figuras Geométricas (MIFG). Los objetivos de cada uno de los módulos son:

- **Módulo de administración de usuario (MAU):** El objetivo de este módulo es llevar el control de los usuarios que ingresan y administran el sistema, lo cual se lleva a cabo por medio del uso de un registro de usuario, el cual recaudará la información básica y necesaria para que el sistema permita la creación de un perfil de juego.
- **Módulo de administración de evaluaciones:** En este módulo se lleva a cabo el control de los registros de las evaluaciones que el alumno va realizando conforme avance por los niveles del software, así como también es la parte que arrojará la estadística del avance que ha tenido el alumno con el uso de esta herramienta y el nivel de competencia que adquirió.
- **Módulo de operaciones de multiplicación:** Este módulo tiene como objetivo proporcionar los ejercicios y problemas adecuados para que el alumno obtenga un nivel de alto, y que estos ejercicios lleven al alumno a descubrir una manera más fácil de comprender este tipo de operaciones.
- **Módulo de operaciones de división:** Tiene como objetivo proporcionar ejercicios adecuados para que el alumno obtenga un buen nivel de competencias y a su vez domine nuevos métodos de resolución para estos problemas, los cuales pueda aplicar en los niveles de competencia subsecuentes.
- **Módulo de identificación de figuras geométricas:** Tiene como objetivo llevar al alumno a la identificación plena de las figuras geométricas: cómo se conforman, cómo obtener un área, entre otras características de las mismas y así, puede obtener el nivel de competencia requerido.

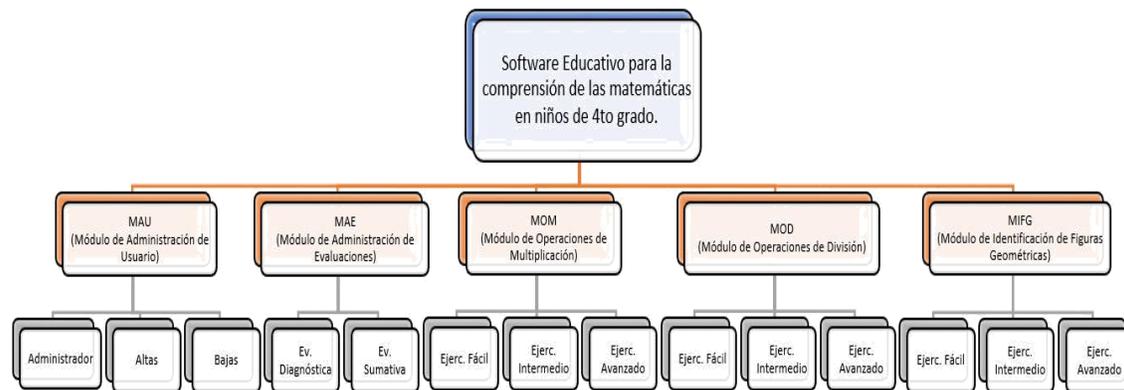


Figura 2. Estructura Modular del software.

Fuente: Elaboración propia

Diseño educativo

El modelo educativo actualmente usado por los profesores de educación básica en México, es el Modelo basado en Competencias, el cual se enfoca en educar integralmente al alumno combinando conocimientos y habilidades, haciendo que el alumno sea capaz de razonar un problema o situación para dar una solución coherente, aplicando sus conocimientos y habilidades para resolverlo.

El software se basa en el m-learning (mobile-learning) o aprendizaje móvil, el cual usa sólo la tecnología móvil o la combina con alguna otra tecnología TIC para facilitar el aprendizaje en cualquier lugar en cualquier momento. En un artículo, (UNESCO, 2013) menciona casos de estudio y las muchas ventajas que tiene la implementación del m-learning; algunas de estas ventajas se resumen en mayor alcance y la igualdad de oportunidades en la educación, respuesta y evaluación inmediatas, aprendizaje en cualquier momento y lugar, entre otras.

Es por este modelo que se diseña un videojuego multiplataforma educativo, con el fin de que los usuarios puedan mejorar sus competencias, sin importar el lugar en donde se encuentren y el dispositivo que estén usando (celulares, tabletas, laptops, computadoras de escritorio, etc).

Diseño de comunicación general

La comunicación de la aplicación tiene una primera instancia con la tienda de aplicaciones de Android, una vez que el usuario instala la aplicación en su dispositivo móvil, la aplicación inicia y se pide que realice un perfil con nos pocos datos, al hacer esto tendrá acceso a los niveles del juego y a las evaluaciones que se realizan al finalizar cada uno de los niveles. Después de que las evaluaciones hayan realizado, el usuario puede revisar su nivel de competencia en el tema y nivel que ya haya sido superado y, si así lo desea, puede repetir los niveles que desee para aumentar sus

niveles de competencia. De acuerdo a lo anterior, se reconocen como interfaces principales:

- Inicio (creación/elección de perfil de usuario)
- Nivel de competencia (por nivel, por tema)
- Ejercicios (en sus diferentes tipos)
- Acerca de (sobre la aplicación y sus creadores).

Plan de iteraciones

Diseño de iteraciones

La planificación de las iteraciones se realizaron después de un análisis del tiempo que llevaría realizar cada módulo, dejando unos días como colchón de tiempo para que cada módulo saliera con el más mínimo retraso. Se sigue una secuencia de orden según los temas, para después realizar las evaluaciones que corresponden a cada iteración, para después, administrar los usuarios que ingresaran al software y comiencen a usarlo.

Tabla 1. Planificación de iteraciones.

Módulo	Iteración	Comienzo	Fin	Duración
MOM	Iteración multiplicación	05/sep/2016	25/sep/2016	21d
MOD	Iteración División	26/sep/2016	16/oct/2016	21d
MIFG	Iteración Figuras Geométricas	17/oct/2016	06/nov/2017	21d
MAE	Iteración Evaluaciones	07/nov/2016	27/nov/2017	21d
MAU	Iteración Usuarios	28/nov/2016	11/dic/2016	14d

Priorización de iteraciones

Las iteraciones con mayor prioridad corresponden a los módulos que contienen todos los ejercicios y la evaluaciones, debido a que es la parte fuerte y fundamental del software. Debido a que, sin estos módulos el software se asigna la prioridad de 1 a 5 (mayor a menor prioridad).

Tabla 2. Priorización de las iteraciones.

Prioridad	Iteración	Módulo
1	Iteración multiplicación	MOM
2	Iteración División	MOD
3	Iteración Figuras Geométricas	MIFG
4	Iteración Evaluaciones	MAE
5	Iteración Usuarios	MAU

Diseño computacional

Modelo de Diseño

Para comprender más a profundidad el funcionamiento y requerimientos de la aplicación se desarrollaron varios diagramas UML; el diagrama Entidad-Relación (E-R) de la base de datos se describieron 8 tablas para manejar de forma adecuada todos los datos que son necesarios para que la aplicación pueda funcionar. Estas tablas almacenan la información de los usuarios, de las evaluaciones, las puntuaciones, los ejercicios y las respuestas correctas de estos; otro diagrama importante para el software fue el diagrama de flujo de datos, en el cual se puede apreciar el funcionamiento del todo el software de forma abstracta en el diagrama.

Los diagramas de caso de uso, no dejan de ser importantes para este software, ya que ayudan a comprender de forma más detallada la interacción de este con los usuarios y los roles que tendrá cada uno.

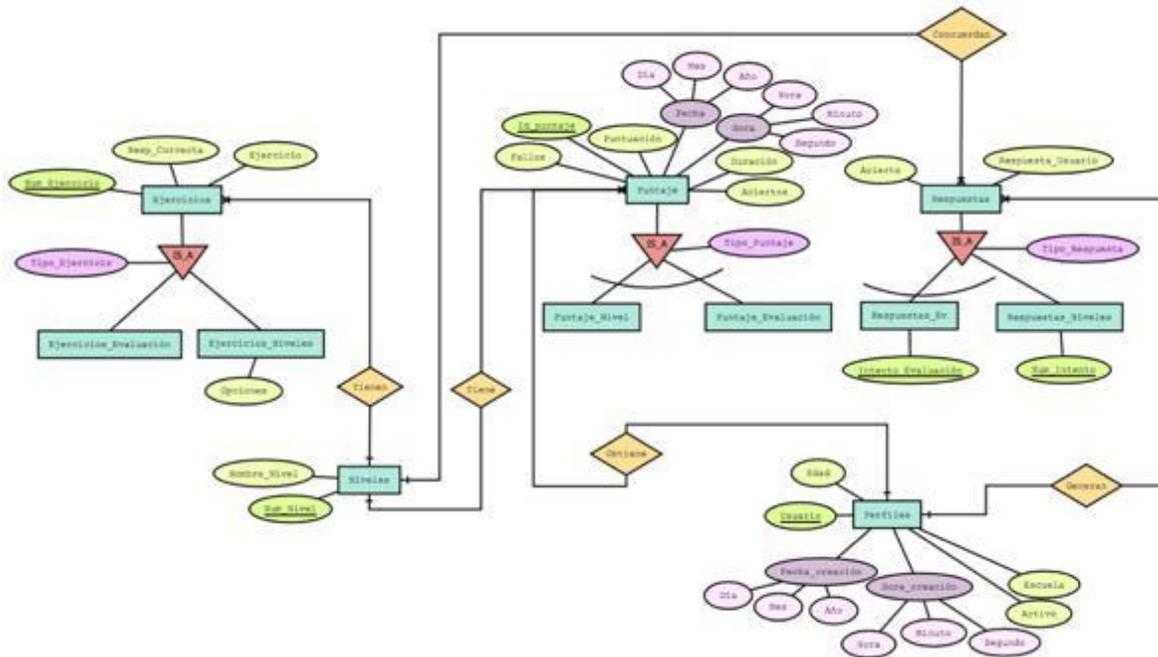


Figura 3. Diagrama E-R.

Fuente: Elaboración propia.

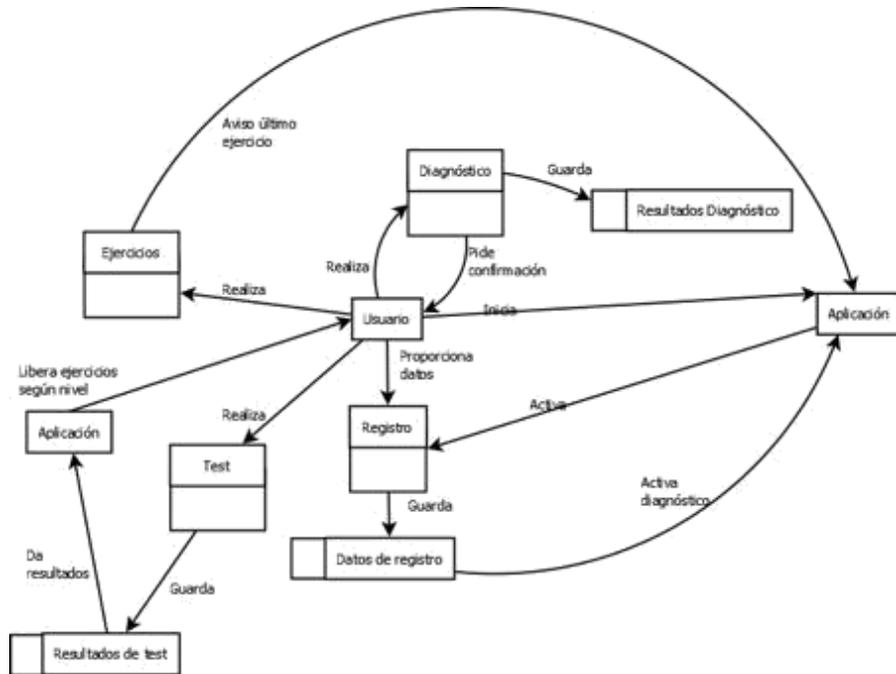


Figura 4. Diagrama de flujo de datos general.

Fuente: Elaboración propia

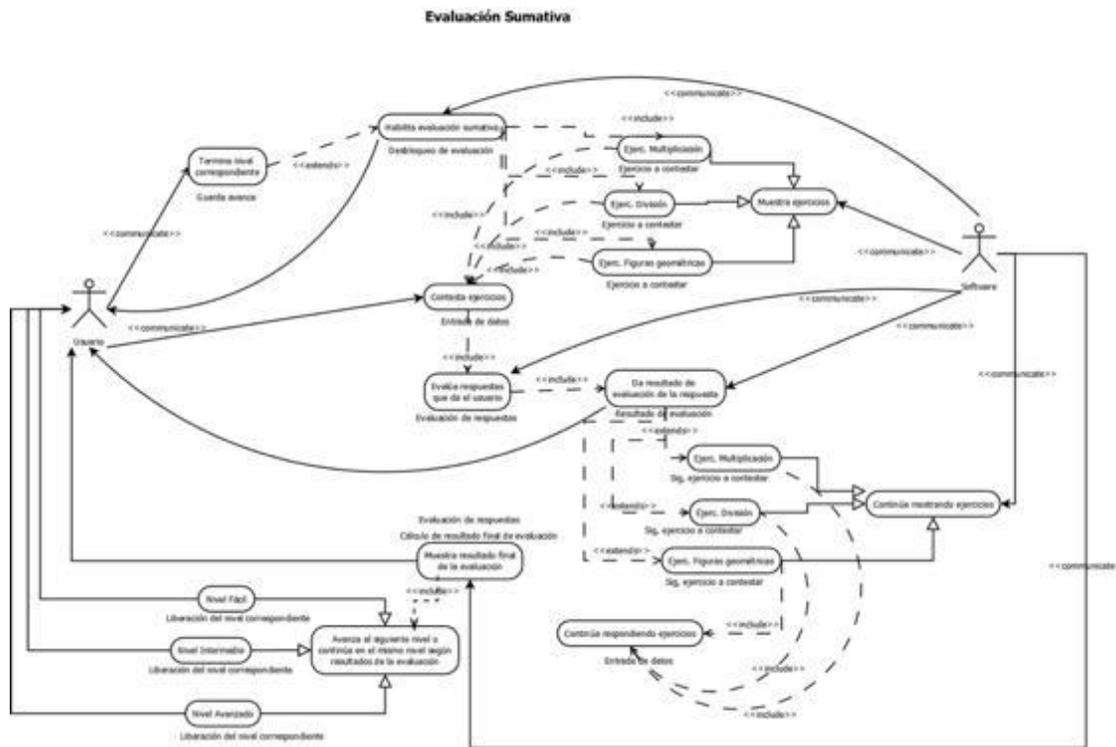


Figura 5. Caso de uso: Evaluación sumativa.

Fuente: Elaboración propia.

Diseño de navegación

La primera pantalla del software, es la de inicio, donde se crea o elige un perfil que previamente fue creado. Una vez registrado o iniciado con su perfil, el usuario puede comenzar a usar la aplicación. Se tiene un menú de opciones al usuario: jugar, tienda, records, créditos y salir; donde cada opción arrojará una interfaz de acuerdo a lo que se desea realizar, respectivamente. En el caso de “jugar” muestra al usuario un primer nivel que debe ser superado para que pueda acceder a los siguientes, los ejercicios comenzarán a mostrarse de forma aleatoria dando la oportunidad al usuario de cometer más de una vez un error hasta que pueda superar ese nivel y los siguientes. En el menú de records, el usuario puede ver el nivel de competencia que adquiere según los resultados obtenidos en las diferentes evaluaciones que realice conforme vaya avanzando. En el menú de créditos, se puede consultar la información de los participantes en la realización del software.

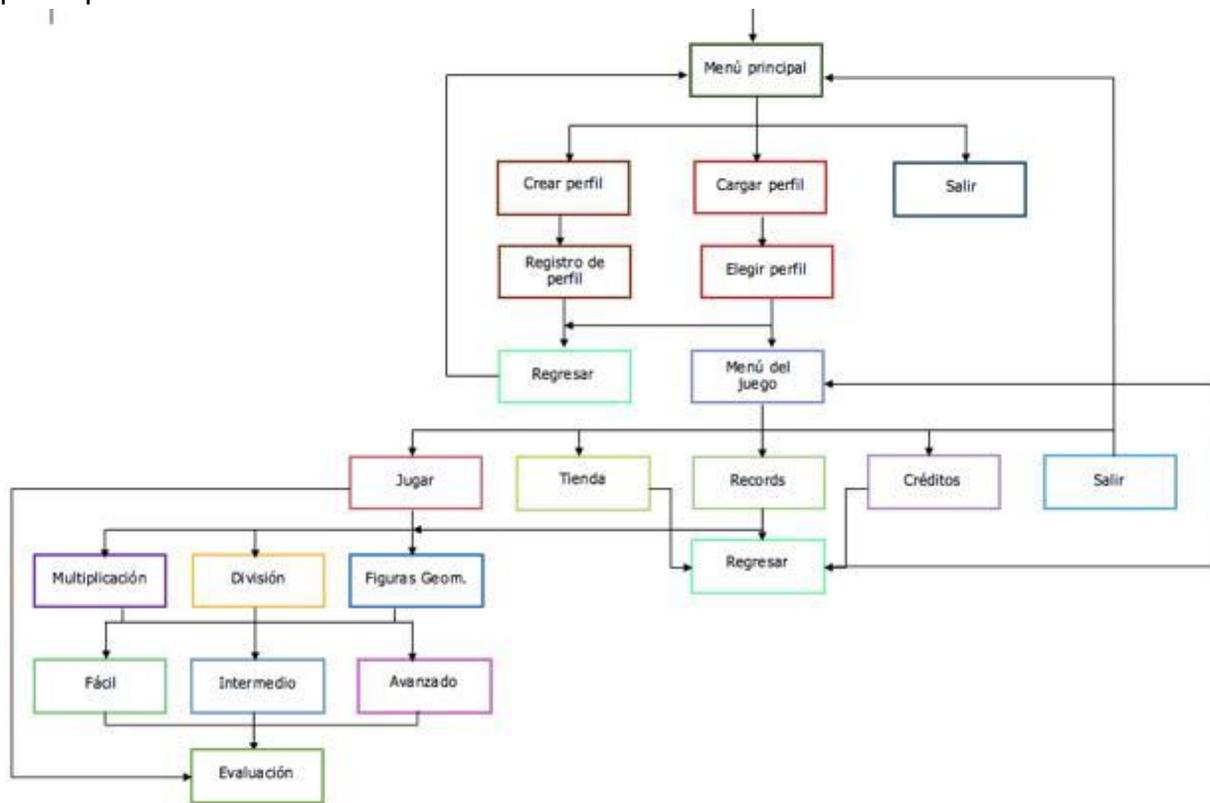


Figura 6. Navegación del software.

Fuente: Elaboración propia.

Modelo de interfaz

La estructura de la interfaz se diseña a partir de varios prototipos realizados, tomando en cuenta los resultados obtenidos de la retroalimentación que, usuarios potenciales dieron mediante el método de recolección de datos Focus Group, donde se realizó una muestra del Producto Mínimo Viable a tres grupos de 15 niños, donde se les permitió la



interacción con el software y después se realizaron una serie de preguntas donde todos aportaron sus observaciones sobre el software.

En el videojuego, los usuarios deben mover al personaje de forma que esquive los obstáculos que se presentan en su camino, mientras que da solución al ejercicio que se le presenta, seleccionando el cubo que tiene la respuesta correcta. Los ejercicios pueden ser de selección o complementación, de temas de multiplicación, división e identificación de figuras geométricas y con niveles de dificultad fácil, intermedio y avanzado.

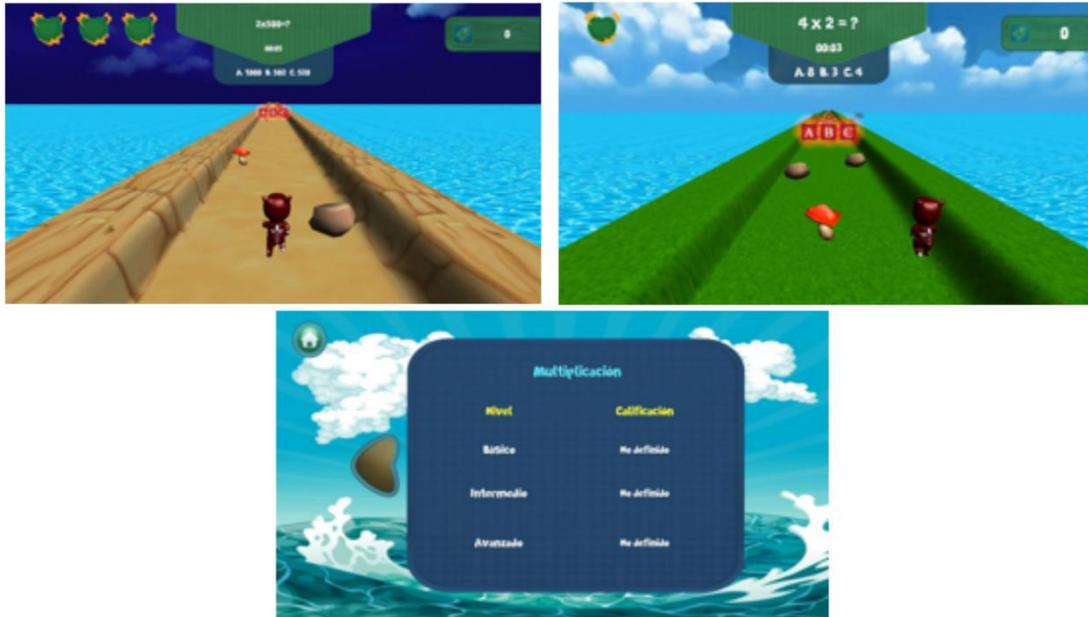


Figura 7. Ejercicios de multiplicación en Mate-Movil.

Fuente: Elaboración propia

Como se puede observar en la figura 7, las diferentes pantallas de Mate-Movil se usa una paleta de colores neutra en la que todos los objetos que conforman las interfaces puedan reconocerse e identificarse con facilidad, tomando en cuenta las observaciones que fueron realizadas por los usuarios potenciales.

Desarrollo

Desarrollo de componentes

Los diferentes componentes que forman parte del software, se desarrollan de acuerdo al plan de iteraciones: se realizan según su prioridad y conforme al tiempo que tiene asignado el desarrollo de cada uno de los módulos. El diseño del personaje, escenarios y otros objetos que se incluyen en el videojuego, se comenzaron a trabajar unas semanas antes de comenzar a desarrollar los módulos, esto para trabajar de forma más fluida en los módulos al tener los elementos generales que se usarían en los módulos.

El personaje fue modelado desde cero en 3D y los movimientos fueron modelados también, y posteriormente controlados desde el motor de videojuegos: se crea un controlador donde se especifican estados del movimiento del personaje dependiendo de cómo se comportan las variables alrededor de este: hacia donde el usuario mueve el personaje, si chocó con un obstáculo y el cubo que haya elegido.

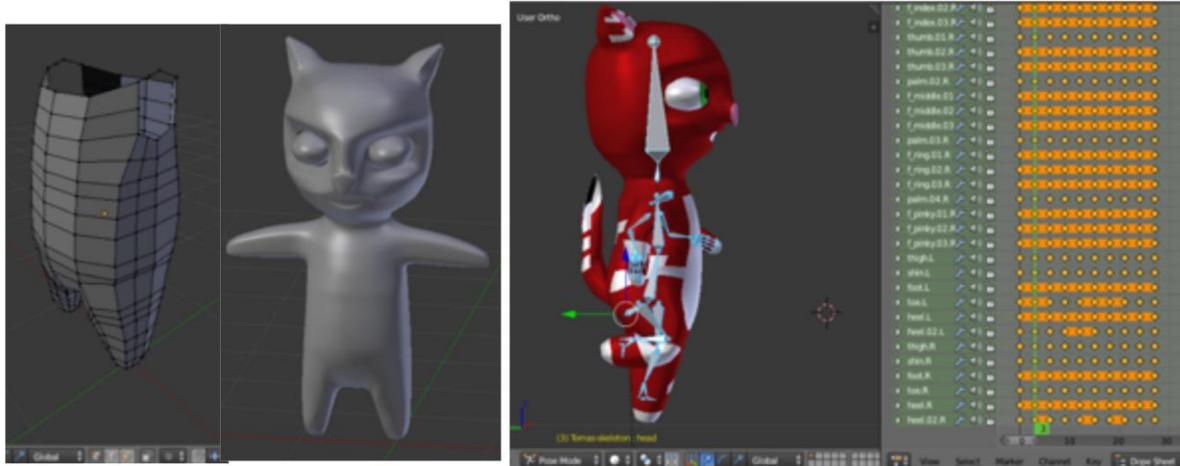


Figura 8. Modelado y animación de personaje.

Fuente: Elaboración propia.

La BD se crea desde el motor de videojuegos y se hace una configuración para que en la instalación de Mate-Movil en un nuevo dispositivo, no tenga problemas de conexión con la BD y ocasione diferentes problemas en el funcionamiento del software. Los componentes más importantes de Mate-Movil se describen a continuación:

- **CameraController:** se manejan los movimientos que obtendrá el objeto cámara, estos movimientos se realizan en los ejes X, Y, y Z. Se le asigna una función que inicia la acción para perseguir al objeto “player” el cual sería el personaje principal del juego. Además, en este bloque de código se le asigna un movimiento inicial en el que, al principio de cada nivel de juego, la cámara recorre una parte del mapa hasta llegar al player.
- **TestController:** Esta clase permite que se visualicen los ejercicios de las evaluaciones dentro de la escena. Llevando un control de los aciertos y errores que se obtienen de la respuesta que brinda el usuario a cada ejercicio. Además, al inicio de la escena extrae todos los ejercicios de la base de datos y al final inserta las respuestas obtenidas.
- **Records:** Extrae los datos del perfil que se encuentra activo de la base de datos y le asigna un nivel de dominio al usuario dependiendo de los aciertos que haya obtenido por nivel, estos pueden ser: muy bajo, bajo, medio, alto y muy alto.
- **ObstacleController:** Detecta cuando el personaje colisiona con alguno de los objetos denominados obstáculos que se encuentran dentro de la escena.



- **LoadProfile:** Este script contiene un método que se activa cuando el usuario escoge un perfil para jugar dentro del juego, ya que este método modifica el estado del perfil cambiándolo de 0 a 1 o inactivo a activo.
- **PlayerController:** Se le asigna movimiento al objeto player el cual será el personaje principal dentro del juego, este tendrá una velocidad la cual se le asigna dentro de este script al igual que se verá afectado por la gravedad la cual también se asigna dentro de la clase. El personaje cuenta con 2 animaciones, la primera en la cual el personaje se encuentra parado y la segunda donde el personaje se encuentra corriendo, por lo que se necesita controlar el momento en el que se ejecuta cada animación, además con el uso de las teclas o la pantalla, dependiendo del dispositivo, el jugador desplaza al personaje hacia los lados.
- **LevelController:** Extrae los ejercicios del nivel en el que se está jugando para posteriormente, mostrarlos en el cuadro del ejercicio. Además, cada ejercicio contiene diferentes respuestas las cuales se muestran dentro del cuadro de opciones. Este script también controla cuándo deben aparecer los obstáculos dentro de la escena, así como los objetos que contienen las opciones. Al final de la partida se insertan todas las respuestas obtenidas por el usuario dentro de la base de datos.

Prueba de componentes

Los componentes se probaron un vez terminados para comprobar que se comportaran de acuerdo a su función. Debido al motor de videojuegos, se minimiza la posibilidad a la posibilidad de nulos en los componentes.

El componente más susceptible a nulos, es el controlador que obtiene los ejercicios de la BD, los coloca en un arreglo y después los muestra en los cubos y el panel del ejercicio a resolver. Este componente al presentarse un nulo, ni siquiera se podía ejecutar el software. Esto se controló haciendo una validación de la existencia de los datos que se extraían de la base de datos, y una vez validado, poder asignar la información al arreglo y poderla manejar dentro del software.

Las pruebas realizadas se enfocaron en la consistencia de la información, ya que es de gran importancia que exista coherencia entre el módulo y la información a mostrar: que los resultados concuerden con el perfil de usuario correspondiente, que los ejercicios se muestren de acuerdo al tema y nivel y que las respuestas concuerden a estos, entre otros.

Integración

La integración se realizó entre componentes que debían funcionar para un fin: el controlador de obstáculos, el controlador del jugador y el controlador de los cubos deben funcionar en conjunto para que: cuando el personaje choque tres veces con los obstáculos el juego termine, en el momento que el usuario elija una respuesta incorrecta haga un cambio en su puntuación y avise al usuario del error.



La integración más importante, es la integración de todos los módulos: esta integración se puede observar y acceder en los menús, ya que es en estos en los que se hace la vinculación de un módulo con otro y la información que se usa en varios de los módulos.



Figura 9. Integración de módulos.

Fuente: Elaboración propia.

Pruebas de integración

Las pruebas de integración se realizaron en torno al correcto funcionamiento de la integración, probando los vínculos de los menús, la consulta y validación de información de varios módulos así como su consistencia y coherencia. Debido a la propiedad de aleatoriedad de los ejercicios, se pudo comprobar que los ejercicios correspondían al nivel y tema que le correspondía

Todos los vínculos y módulos trabajaban de forma correcta, y no ocurrió ningún error que detuviera el software o que interrumpiera el juego: esto se comprobó en el momento en el que se pidió a un pequeño grupo de usuarios potenciales interactuar con el software y no se registrara alguna actividad anormal durante la interacción con los usuarios.

Despliegue

Retroalimentación del usuario

Para obtener retroalimentación, se realizan encuestas a dos escuelas primarias muestra, para exponer los objetivos principales del videojuego y los niños de cuarto grado pudieran utilizar el videojuego y pudiera calificar su experiencia. Una vez que fueron expuestos los contenidos y los objetivos que tiene el videojuego, se realiza un sondeo para saber la opinión popular y general de los niños acerca si el videojuego fue de su agrado, así mismo cuáles eran las cosas que les gustaban o resaltaban más en el videojuego y cuáles elementos no eran de su agrado o se les complicaba entender o incluso algunos elementos que ellos consideraran que podían ser agregados para que

su experiencia fuera mucho mejor y pudieran así poder usar el videojuego el mayor tiempo posible.

Después de esto se da paso a la aplicación de una encuesta previamente elaborada, para poder obtener datos de la experiencia que tuvieron los niños después de haber probado el videojuego y así poder recabar más información para las futuras retroalimentaciones.

Evaluación de calidad y satisfacción del cliente

Tomando en cuenta la retroalimentación, la satisfacción de los usuarios finales fue de un 95%, ya que los comentarios sobre los componentes, animaciones, sonidos, ejercicios, niveles, etc. Que se consideraron a mejora fueron relativamente pocos.

Evaluando la calidad de la aplicación en cuanto a la interfaz gráfica y funcionalidad, se califica con un 90%, debido a que se obtuvieron más observaciones fue en la parte de gráficos y cuestiones de jugabilidad.

Evaluación de conveniencia de nueva iteración

Debido a las observaciones que se obtuvieron de la iteración, se considera que existe conveniencia de realizar otra iteración ya que el diseño, la estructura y el contenido fueron aprobados, pero se opta por mejorar en aspectos de diseño y jugabilidad, porque los elementos que se presentan en el videojuego en ocasiones son demasiados, tanto como para que el niño pueda tener su concentración en resolver el ejercicio que se le muestra, y en términos de jugabilidad se aprecian algunas deficiencias de movimiento mismas que en la siguiente iteración serán resueltas como una nueva versión de actualización del videojuego.

Resultados

El poder trabajar con la metodología MeISE, el equipo de desarrollo pudo observar que la manera de desarrollo propuesta por esta metodología es agradable y también como metodología ágil su grado de iteraciones está ligado al cliente, ya que al tratarse de software educativo, el cliente debe encargarse de poder dar la aprobación, para poder seguir con el desarrollo del producto, que en este caso es muy favorable, porque así se da más confiabilidad al software y más sustento a la parte del contenido, ya que principalmente esta es la diferencia que tiene con otras aplicaciones y juegos actuales, los cuales no tienen una estructura definida y en ocasiones la estructura propuesta no abarca los temas necesarios o que debiera estar cursando el alumno en el nivel en el que se encuentra, y por eso esta metodología se adapta a las necesidades de desarrollo y fines del videojuego a desarrollar.

Pudimos observar las reacciones de los niños con la prueba del videojuego, los niños mostraban una actitud receptiva hacia lo que el videojuego les mostraba y conforme avanzaban en los niveles que tiene el videojuego, se iba haciendo más notoria la



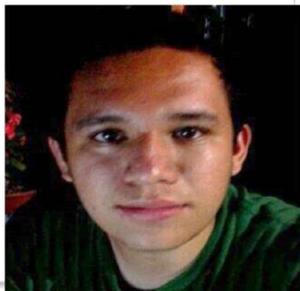
alegría y concentración que despertaba en la mayoría de los niños. Según lo observado y los comentarios de los niños, la primera versión de la aplicación está casi lista: aunque aún le falten algunos componentes y aspectos pequeños a mejorar, aunque se realizarán en la próxima iteración.

Conclusiones

El trabajar con MeISE fue una buena experiencia, ya que se puede trabajar de una manera agradable, gracias a la estructura que tiene y al fin que tiene, ya que cada una de las fases de la metodología están bien especificadas y entendibles para al llegar a la etapa de las iteraciones, ya se tenga una conceptualización y conocimiento amplio de lo que se va a desarrollar. A su vez pudo observar que en lo general se cubren los requisitos funcionales, ya que a los niños les gustó usar el software y a su vez les generaba curiosidad por saber qué era lo que venía y que más se agregaría en la siguiente iteración de Mate-Movil, ya que al finalizar la prueba y las encuestas de satisfacción, se comentó que eso que habían visto y probado, estaba terminado, pero que aún así como todo software se puede mejorar y poder mejorar la calidad tanto del software como el entendimiento de los contenidos que tiene el software agregando características que actualmente no existen, haciendo posible la inclusión de las tecnologías de la información en el aula en pos de la educación y el m-learning.

Referencias

- Abud Figueroa, Ma. Antonieta. (2009). MeISE: Metodología de Ingeniería de Software Educativo. *Revista Internacional de Educación En Ingeniería*, 2(1), 1–9.
- Bermúdez, A., García-Varea, I., López, M. T., Montero, F., De la Ossa, L., Puerta, J. M., ... Sánchez, J. L. (2011). Una Definición precisa del concepto "nivel de dominio de una competencia" en el marco del aprendizaje basado en competencias. *Jornadas de Enseñanza Universitaria de La Informática (17es: 2011: Sevilla)*. Retrieved from <http://upcommons.upc.edu/handle/2099/11958>
- (INEGI), I. N. de E. y G. (2016, June 23). Directorio Estadístico Nacional de Unidades Económicas. Retrieved June 24, 2016, from <http://www3.inegi.org.mx/sistemas/mapa/denue/default.aspx#>
- López-Vargas, O., Hederich-Martínez, C., & Camargo-Urbe, Á. (2011). Estilo cognitivo y logro académico. *Educación Y Educadores*, 14(1), 67–82.
- Roig-Vila, R., Mengual-Andrés, S., & Quinto-Medrano, P. (2015). Primary Teachers' Technological, Pedagogical and Content Knowledge/Conocimientos tecnológicos, pedagógicos y disciplinares del profesorado de Primaria. *Comunicar*, 23(45), 151.
- Serrano González-Tejero, J. M., & Pons Parra, R. M. (2011). El constructivismo hoy: enfoques constructivistas en educación. *Revista Electrónica de Investigación Educativa*, 13(1), 1–27.
- UNESCO. (2013). *Directrices para las políticas de aprendizaje móvil*. París, Francia: UNESCO. Retrieved from <http://unesdoc.unesco.org/images/0021/002196/219662S.pdf>



Giovanni Sinaí Silva Barragán

Edad: 22 años

E-mail: giovannigatfut@gmail.com

ESTUDIOS

Escuela Primaria Urbana Federal "Ricardo Flores Magón"
2006, Arteaga, Michoacán, México.

Escuela Secundaria Particular "Vasco de Quiroga"
2009, Arteaga, Michoacán, México.

Escuela Preparatoria Particular Fray Bernardino de Sahagún
2012, Lázaro Cárdenas, Michoacán, México.

IDIOMAS

Español: Natal

Inglés: Básico

FORMACIONES ADICIONALES E INTERESES

IV Congreso de Ingeniería en Sistemas Computacionales: Open Source
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

Curso-Taller Desarrollo de Aplicaciones Móviles Android
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

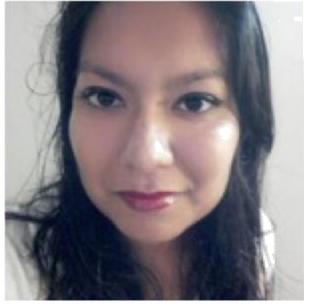
V Congreso de Ingeniería en Sistemas Computacionales: Seguridad
Informática Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

III Congreso de Ingenierías
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

XXI Expo Ingenierías: Concurso Nacional
Instituto Tecnológico de Querétaro
Santiago de Querétaro, Querétaro, México

VI Congreso de Ingeniería en Sistemas Computacionales: ¡Preparándote para innovar el futuro!
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

XXIII Congreso Internacional Sobre Educación, Ciencia y Tecnología 2016
Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, México



Laura Nelly Parra Hernández

Edad: 22 años

E-mail: lauranelly.parra@gmail.com

ESTUDIOS

Escuela Primaria Urbana Federal "Ignacio Zaragoza"
2007, Lázaro Cárdenas, Michoacán, México.

Escuela Secundaria Federal No. 1 "Ricardo Flores Magón"
2010, Lázaro Cárdenas, Michoacán, México.

Técnico Profesional Bachiller en Informática
Colegio Nacional de Educación Profesional Técnica, Plantel Lázaro Cárdenas (Conalep 035)
2013, Lázaro Cárdenas, Michoacán, México.

IDIOMAS

Español: Natal

Inglés: Básico

FORMACIONES ADICIONALES E INTERESES

IV Congreso de Ingeniería en Sistemas Computacionales: Open Source
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

Curso-Taller Desarrollo de Aplicaciones Móviles Android
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

V Congreso de Ingeniería en Sistemas Computacionales: Seguridad
Informática Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

III Congreso de Ingenierías
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

XXI Expo Ingenierías: Concurso Nacional
Instituto Tecnológico de Querétaro
Querétaro, Querétaro, México

VI Congreso de Ingeniería en Sistemas Computacionales: ¡Preparándote para innovar el futuro!
Instituto Tecnológico de Lázaro Cárdenas
Lázaro Cárdenas, Michoacán, México

XXIII Congreso Internacional Sobre Educación, Ciencia y Tecnología 2016
Facultad de Ingeniería Mecánica y Eléctrica, Universidad Autónoma de Nuevo León, México