



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS

DESARROLLO DE UNA APLICACIÓN MÓVIL QUE GENERE Y
MUEVA FIGURAS GEOMÉTRICAS, MEDIANTE LA
IDENTIFICACIÓN DE PATRONES DE VOZ

Trabajo de Titulación

Tipo: Proyecto Técnico

Presentado para optar al grado académico de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: PEDRO HUMBERTO GUAILLAS GUAILLAS

DIRECTOR: DR. ÁLVAREZ OLIVO ALONSO WASHINGTON

Riobamba-Ecuador

2020

©2020, Pedro Humberto Guillas Guillas

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el derecho del autor.

Yo, Pedro Humberto Guallas Guallas, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; el patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.
Riobamba, 30 de enero de 2020

Pedro Humberto Guallas Guallas
110516769-4

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA EN SISTEMAS

El Tribunal del trabajo de titulación certifica que: El trabajo de titulación: Tipo: Proyecto Técnico, **DESARROLLO DE UNA APLICACIÓN MÓVIL QUE GENERE Y MUEVA FIGURAS GEOMÉTRICAS, MEDIANTE LA IDENTIFICACIÓN DE PATRONES DE VOZ**, realizado por el señor: **PEDRO HUMBERTO GUAILLAS GUAILLAS**, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtual el Tribunal Autoriza su presentación.

FIRMA

FECHA

Ing. Lorena Aguirre

PRESIDENTE DEL TRIBUNAL

Dr. Alonzo Washington Álvarez Olivo

DIRECTOR/A DEL TRABAJO

DE TITULACION

Ing. Eduardo Rolando Villa Villa

MIEMBRO DE TRIBUNAL

DEDICATORIA

Dedico el presente trabajo de titulación a Dios por darme sabiduría y conocimiento, a mis padres y hermanos quienes fueron el pilar fundamental durante la carrera universitaria y durante mi vida, quienes han entregado su sacrificio por encima de las posibilidades que pudieron ayudar, para de esa manera pueda cumplir mis metas personales y profesionales.

Pedro

AGRADECIMIENTO

Agradezco a Dios por darme salud y vida para poder cumplir con la misión que tiene para cada uno de nosotros, a mi familia por ese apoyo incondicional en el transcurso de mi vida universitaria y a la Escuela Superior Politécnica de Chimborazo por abrir las puertas y permitirme formar como profesional, junto con los docentes que compartieron sus conocimientos en las aulas, y me han dado directrices necesarias en el ámbito profesional.

Pedro

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	ix
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xii
ÍNDICE DE ANEXOS	xiii
RESUMEN.....	xv
SUMMARY	xvi
INTRODUCCIÓN	1

CAPÍTULO I

1. MARCO TEÓRICO	6
1.1. Figuras geométricas	6
<i>1.1.1. Aprendizaje de las figuras geométricas</i>	<i>6</i>
<i>1.1.2. Geometría euclidiana</i>	<i>7</i>
1.2. Primitivas geométricas.....	9
1.3. Dispositivo móvil	10
<i>1.3.1. Categoría de los dispositivos móviles.....</i>	<i>10</i>
<i>1.3.2. Asistente Digital Personal (PDA)</i>	<i>10</i>
<i>1.3.3. Teléfonos móviles</i>	<i>10</i>
<i>1.3.4. Smartphone (teléfono inteligente)</i>	<i>11</i>
<i>1.3.5. Android.....</i>	<i>11</i>
<i>1.3.6. Sistema operativo Android.....</i>	<i>11</i>
1.4. Aplicación móvil.....	12
1.5. Kotlin.....	13
<i>1.5.1. Características de Kotlin</i>	<i>14</i>
<i>1.5.2. Propiedades de Kotlin.....</i>	<i>15</i>
1.6. OpenGL ES.....	16
<i>1.6.1. Características de OpenGL ES</i>	<i>16</i>
<i>1.6.2. Propiedades de la librería</i>	<i>18</i>

1.6.3.	<i>Funcionamiento de la librería</i>	19
1.6.4.	<i>Uso de la librería</i>	20
1.7.	Reconocimiento de voz	21
1.7.1.	<i>Proceso del reconocimiento de voz</i>	22
1.8.	Calidad de Software	22
1.8.1.	<i>Norma ISO 9241</i>	23

CAPÍTULO II

2.	MARCO METODOLÓGICO	25
2.1.	Descripción del lugar de estudio	25
2.2.	Alcance	25
2.3.	Tipo de estudio	26
2.4.	Métodos	26
2.4.1.	<i>Método descriptivo</i>	26
2.4.2.	<i>Método analítico</i>	27
2.5.	Técnicas	27
2.5.1.	<i>Entrevista</i>	27
2.5.2.	<i>Observación</i>	27
2.5.3.	<i>Encuesta</i>	27
2.6.	Análisis previo al desarrollo del proyecto	27
2.6.1.	<i>Estudio de factibilidad</i>	28
2.6.2.	<i>Gestión de Riesgos</i>	30
2.7.	Metodología SCRUM para el desarrollo de software	32
2.7.1.	<i>Análisis de requerimientos</i>	32
2.7.2.	<i>Diseño</i>	34
2.7.3.	<i>Desarrollo</i>	36
2.7.4.	<i>Pruebas</i>	48
2.7.5.	<i>Implantación</i>	48

CAPÍTULO III

3.	MARCO DE RESULTADOS Y DISCUSIÓN DE LOS RESULTADOS	49
3.1.	Población y muestra.....	49
<i>3.1.1.</i>	<i>Población.....</i>	<i>49</i>
<i>3.1.2.</i>	<i>Muestra.....</i>	<i>49</i>
3.2.	Parámetros de evaluación.....	50
<i>3.2.1.</i>	<i>Criterios de evaluación.....</i>	<i>51</i>
<i>3.2.2.</i>	<i>Análisis de los parámetros de la aplicación móvil</i>	<i>51</i>
3.3.	Análisis de resultados.....	52
<i>3.3.1.</i>	<i>Análisis de eficiencia.....</i>	<i>52</i>
<i>3.3.2.</i>	<i>Análisis de eficacia.....</i>	<i>53</i>
<i>3.3.3.</i>	<i>Análisis de satisfacción</i>	<i>54</i>
3.4.	Prueba de significancia estadística	55
<i>3.4.1.</i>	<i>Determinación de variables</i>	<i>55</i>
<i>3.4.2.</i>	<i>Formulación de la hipótesis.....</i>	<i>55</i>
<i>3.4.3.</i>	<i>Procedimiento.....</i>	<i>55</i>
	CONCLUSIONES.....	59
	RECOMENDACIONES.....	60
	BIBLIOGRAFÍA	
	ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-1: Principales figuras geométricas planas	8
Tabla 2-1: Figuras geométricas del espacio más conocidas.....	9
Tabla 1-2: Hardware requerido	28
Tabla 2-2: Software requerido	28
Tabla 3-2: Costo de desarrollo	29
Tabla 4-2: Recursos humanos	29
Tabla 5-2: Identificación de riesgos.....	30
Tabla 6-2: Criterios de valoración de probabilidad.....	30
Tabla 7-2: Criterios de valoración del impacto.....	31
Tabla 8-2: Criterios de valoración de la exposición.....	31
Tabla 9-2: Análisis de riesgos.....	31
Tabla 10-2: Priorización de riesgos	32
Tabla 11-2: Tareas por realizar	33
Tabla 12-2: Método de estimación T-Shirt.....	33
Tabla 13-2: Panificación	34
Tabla 14-2: Personas y roles	37
Tabla 15-2: Tipo y usuario de la aplicación móvil.	37
Tabla 16-2: Product Backlog	37
Tabla 17-2: Modelo de la historia de usuario.....	38
Tabla 18-2: Sprint Backlog	39
Tabla 19-2: Sprint 1	40
Tabla 20-2: Sprint 2	40
Tabla 21-2: Sprint 3	41
Tabla 22-2: Sprint 4	42
Tabla 23-2: Sprint 5	43
Tabla 24-2: Sprint 6	43
Tabla 25-2: Sprint 7	44
Tabla 26-2: Sprint 8	45
Tabla 27-2: Sprint 9	45
Tabla 28-2: Sprint 10	46
Tabla 29-2: Reuniones y entregables	46
Tabla 30-2: Actividades para la conclusión del aplicativo.	47
Tabla 1-3: Resultados de la encuesta	51
Tabla 2-3: Análisis de eficiencia.....	52

Tabla 3-3: Resultados agrupados de eficiencia.....	52
Tabla 4-3: Análisis de eficacia.....	53
Tabla 5-3: Resultados agrupados de eficacia.....	53
Tabla 6-3: Análisis de satisfacción.....	54
Tabla 7-3: Resultados agrupados de satisfacción.....	54
Tabla 8-3: Frecuencia observada.....	56
Tabla 9-3: Frecuencia esperada.....	57

ÍNDICE DE FIGURAS

Figura 1-1: Arquitectura del Sistema Android.....	12
Figura 2-1: Proceso para renderizar una figura con la librería OpenGL ES	19
Figura 1-2: Diseño de la interfaz de usuario.....	26
Figura 2-2: Diseño de la interfaz de usuario.....	35
Figura 3-2: Diagrama de componentes de la aplicación.....	36
Figura 4-2: Figura geométrica Línea	40
Figura 5-2: Figura geometría Triángulo	41
Figura 6-2: Figura geometría Cuadrado.....	41
Figura 7-2: Figura geométrica Círculo	42
Figura 8-2: Figura geométrica Cubo.....	42
Figura 9-2: Figura geométrica Esfera	43
Figura 10-2: Figura geométrica Pirámide.....	44
Figura 11-2: Figura geométrica Cilindro	44
Figura 12-2: Botón para dentificar patrones de voz.....	45
Figura 1-3: Distribución Chi cuadrado.	58

ÍNDICE DE GRÁFICOS

Gráfico 1-2: Burndown Chart	48
Gráfico 1-3: Análisis de eficiencia.....	53
Gráfico 2-3: Análisis de eficacia.....	54
Gráfico 3-3: Análisis de satisfacción.	55

ÍNDICE DE ANEXOS

ANEXO A: HOJA DE GESTIÓN DE RIESGOS

ANEXO B: DEFINICIÓN DEL ESTÁNDAR DE CODIFICACIÓN

ANEXO C: MANUAL TÉCNICO

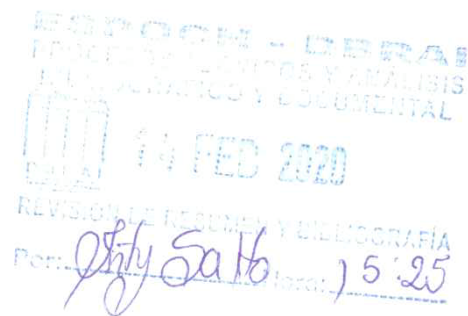
ÍNDICE DE ABREVIATURAS

IDE	Ambiente de Desarrollo Integrado
I/O	Input/Output
OpenGL ES	Librería Sistemas Embebidos
GPS	Sistema de Posicionamiento Global
GPU	Unidad de Procesamiento Gráfico
ISO	Estándar para la Organización Internacional

RESUMEN

El desarrollo de una aplicación móvil que genere y mueva figuras geométricas mediante la identificación de patrones de voz, se desarrolló de acuerdo con los requerimientos planteados desde los propios maestros de la Unidad Educativa Intercultural Bilingüe Monseñor Leónidas Proaño, fue una iniciativa que ayuda en la formación de los niños de tres a cinco años. Una vez obtenidos los requerimientos, se procedió a diseñar las interfaces de acuerdo con las sugerencias de los docentes. Para el desarrollo de la aplicación móvil se utilizó el lenguaje de programación Kotlin y la librería OpenGL ES, para generar y mover figuras geométricas, además en el reconocimiento de voz se utilizó un recurso ya implementado en las tecnologías móviles. En cuanto a la ingeniería de software se utilizó la metodología SCRUM, para facilitar y gestionar el desarrollo de software con entregables a tiempo cortos, y del mismo modo pedir cambios cuando es necesario. En cuanto a la calidad de software se evaluó la usabilidad y sus parámetros eficiencia, eficacia y satisfacción, teniendo presente que la aplicación móvil es para ayudar en la formación de los niños, y por ende la aplicación es muy intuitiva ya que cuenta con un asistente de voz, para informar al niño de las tareas que puede realizar la aplicación y de esa manera el niño menciona la figura geométrica que desea conocer. Finalmente, se concluye que el asistente de voz fue primordial en cuanto a la medición de la usabilidad, ya que debido a ello se pudo evidenciar que los niños en muy poco tiempo pudieron interactuar con la aplicación móvil. Se recomienda continuar desarrollando aplicaciones móviles que ayuden en el área formación académica ya sea inicial, intermedio o superior, con herramientas digitales para adquirir nuevos conocimientos.

Palabras clave: <INGENIERÍA DE SOFTWARE>, <ENSEÑANZA – APRENDIZAJE>, <DESARROLLO DE APLICACIONES MÓVILES>, <RECONOCIMIENTO DE PATRONES DE VOZ>, <FIGURAS GEOMÉTRICAS>.



SUMMARY

The development of a mobile application that generates and moves geometric figures through the identification of voice patterns, it was developed according to the requirements set by the teachers of the Unidad Educativa Intercultural Bilingue Leonidas Proaño, it was an initiative that helps in the formation of children from three to five years. Once the requirements were obtained, the interfaces were designed according to the teachers' suggestions. For the development of the mobile application the Kotlin programming language and the OpenGL ES library were used, to generate and move geometric figures, in addition to voice recognition a resource already implemented in mobile technologies was used. Regarding software engineering, the SCRUM methodology was used to facilitate and manage software development with short-term deliverables, and similarly request changes when necessary. Regarding software quality, usability and parameters were evaluated for efficiency, effectiveness, and satisfaction, bearing in mind that the mobile application is to help in the training of children, and therefore the application is very intuitive since it has an assistant voice, to inform the child of the tasks that the application can perform and that way the child mentions the geometric figure to know it. Finally, it is concluded that voice assistants were paramount in terms of measuring usability since due to this it could be shown that children in a very short time were able to interact with the mobile application. It is recommended to continue developing mobile applications that help in the area of academic training, whether initial, intermediate or higher, with digital tools to acquire new knowledge.

KEYWORDS: <SOFTWARE ENGINEERING>, <TEACHING-LEARNING>, <MOBILE APPLICATION DEVELOPMENT>, <SPEECH PATTERN RECOGNITION>, <GEOMETRIC FIGURES>



INTRODUCCIÓN

En un principio los dispositivos móviles se utilizaron para establecer comunicación a través de la red, ya sea por mensajería o llamada de voz siendo poco usado, este surgimiento se dio a cabo en el año 1983 con la aparición del primer teléfono móvil, el Motorola DynaTAC. Con el transcurso de los años el desarrollo tecnológico de estos dispositivos ha hecho que sea mucho más inteligente, como también mucho más usados en tareas del trabajo cotidiano, educativo, social, etc. Tal es así, que son tan imprescindibles en el diario vivir (Baz *et al.*, 2011, p. 3).

El surgimiento de las aplicaciones móviles fue un cambio radical para la generación actual, que se vio conmovida por experimentar los servicios que se podrían desplegar. Hoy en día, existen millones de aplicaciones móviles, de distinto tipo como: redes sociales, juegos, mensajería, entre otros. Estas aplicaciones debido a los recursos de Hardware que posee el dispositivo, por lo general poseen características de: eficiencia, intuitivas, simplicidad, etc., (Bustos *et al.*, 2015, p. 5).

Frecuentemente se ve el uso de las aplicaciones móviles en distintas áreas y disciplinas ayudando de manera rápida a su acceso sin tener que usar un computador de mesa o laptop. Uno de los campos que mayor aporte ha tenido las aplicaciones móviles, es en el ambiente educativo, donde se parte desde el nivel inicial hasta el nivel superior ayudando en tareas sumamente indispensables, y ofreciendo ventajas para los alumnos a la hora de buscar información, recolección de datos, reconocimiento de elementos, entre otros.

Los niños con las capacidades y habilidades que van adquiriendo diariamente, son muy curiosos con la tecnología especialmente con el uso de las aplicaciones móviles ya que en la mayoría de los padres de familia cuentan con un smartphone, y están propensos a esos, y de alguna manera es bueno ir incursionando en esta era digital, como parte de la formación, porque las aplicaciones móviles están tratando de ayudar adquirir o mejorar las habilidades como: hablar, cantar, reconocer objetos, etc. Esto conlleva a tener el mismo nivel de formación en todos los niños, por el contrario, puede implicar algunas debilidades de aprendizaje en los próximos años o tener un retraso en el reconocimiento de algunos objetos, desarrollo intelectual, adquirir una visión espacial, la simetría o proporción, etc., con respecto a otros niños que si los están haciendo.

El reconocimiento de objetos son unas de las tareas que mayoritariamente realizan los niños, siendo una de esas razones que se plantea, la realización de una aplicación móvil, para ayudar a los niños identificar figuras geométricas básicas, la cual conlleva identificar las figuras geométricas que son relevantes de aprender en esa corta edad.

Formulación del problema

Basados en los antecedentes es eminente destacar los rasgos que trae el uso de una aplicación móvil en un niño, partiendo que hoy en día contar con un dispositivo móvil ayuda de manera eficiente al desempeño de las tareas cotidianas, pero específicamente en los niños se inculcan en la formación tecnológica, por ende, se plantea el siguiente problema:

¿Cómo una aplicación móvil educativa ayudaría a los niños a generar y mover figuras geométricas mediante los patrones de voz?

Sistematización del problema

- ¿El desarrollo de una aplicación móvil ayudará a los niños identificar figuras geométricas?
- ¿Cuánto mejoraría el nivel de aprendizaje en los niños con una aplicación móvil?
- ¿Qué figuras geométricas sería recomendable que un niño aprenda?

Justificación

Justificación teórica

Este trabajo de titulación se realiza con el propósito de aplicar conocimientos adquiridos dentro del campo de formación, y aplicar conocimientos de la ingeniería de software para resolver problemas de la vida diaria, el cual incluye la investigación para adquirir conocimiento de las nuevas tecnologías que se va a utilizar para el desarrollo de la aplicación móvil, y de esa manera desarrollar una aplicación innovadora, mediante la utilización de teorías, métodos y conceptos de ingeniería de software.

Debido a la existencia de varias marcas, modelos, sistemas operativos, etc., en los dispositivos móviles se genera la llamada fragmentación. Teniendo en cuenta lo mencionado el desarrollo de la aplicación móvil debe ser capaz poder cubrir la mayor parte de los dispositivos móviles (Lisandro *et al.*, 2014, p. 1).

Uno de los sistemas operativos, para dispositivos móviles más utilizados, y populares en la actualidad es Android que supera el 80% de los smartphones con este sistema operativo, el cual se encuentra basado en el núcleo de Linux (Rohaut, 2015, p. 46).

En base a las estadísticas de Rohaut (2015, p. 46), se ha determinado realizar la aplicación móvil para la plataforma Android, utilizando el lenguaje de programación Kotlin, para el desarrollo de la aplicación móvil, la cual fue anunciado por los desarrolladores de Google en la conferencia I/O en el año 2017, el lenguaje de programación ya es soportado en el 100% para el desarrollo de aplicaciones móviles para la plataforma Android (Leiva, 2018, p. 5), también se utiliza la librería OpenGL ES que permite generar figuras geométricas de una manera eficiente, y el Entorno de Desarrollo Integrado (IDE) Android Studio para el entorno de desarrollo.

Justificación aplicativa

La aplicación móvil consta de una interfaz sencilla e intuitiva permitiendo de esa manera que los niños, puedan adaptarse a manipular lo más rápido posible, sin tener que recurrir a pedir ayuda a una persona que conozca de la aplicación.

El uso de la librería OpenGL ES, permite generar figuras geométricas sin tener que usar muchos recursos de procesamiento, ya que en las aplicaciones móviles es fundamental, ser eficiente en el uso de los recursos, debido a las limitaciones que las posee (Choudhari & Ranjankar, 2015, p. 1).

La aplicación debe reconocer patrones de voz, para dar la orden que genere o mueva la figura geométrica captando a través de los sensores y posteriormente poder representar en la pantalla del dispositivo.

La aplicación móvil se encuentra organizado en 4 módulos necesario para su desarrollo, donde cada módulo presenta funcionalidades que interactúa con el usuario, en la **Figura 1**, se visualiza la estructura modular de la aplicación móvil.

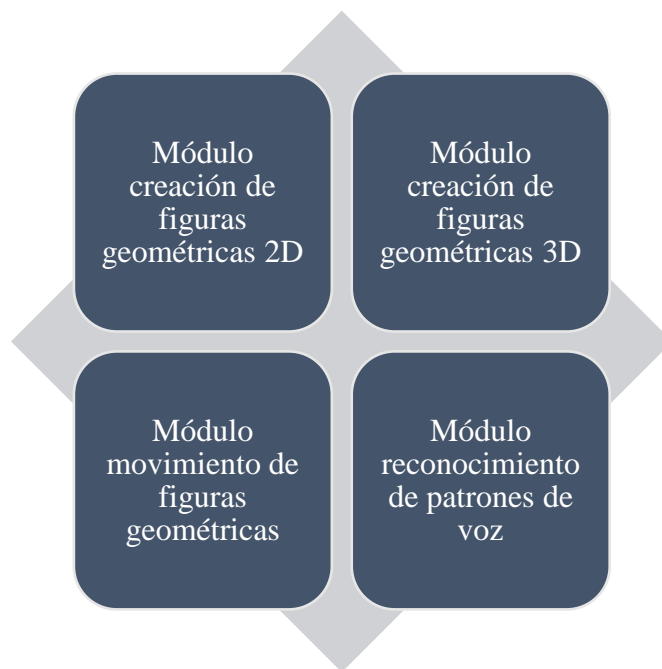


Figura 1: Módulos de la aplicación móvil
Realizado por: Pedro Guailas, 2018

Finalmente, en base a la resolución plateada 582. CP.2012 de la ESPOCH, es pertinente mencionar que el desarrollo del presente proyecto se acopla al área de investigación científica del Ecuador “TIC’S”, la línea de investigación “Tecnologías de la información, comunicación y procesos industriales”, y programas de investigación “Programa para el desarrollo de aplicaciones de software para procesos de gestión y administración pública y privada. Educación”.

OBJETIVOS

Objetivo general

Desarrollar una aplicación móvil que permita generar y mover figuras geométricas mediante la identificación de patrones de voz.

Objetivos específicos

- Estudiar las propiedades y características de la librería OpenGL ES y del lenguaje de programación Kotlin, para el desarrollo de la aplicación móvil.
- Analizar las figuras geométricas que pueden identificar niños entre 3 hasta 5 años.
- Desarrollar una aplicación móvil siguiendo la metodología de desarrollo SCRUM.
- Evaluar la calidad de software de la aplicación móvil acorde a un estándar.

CAPÍTULO I

1. MARCO TEÓRICO

En este capítulo se realiza una investigación teórica de los temas que requieren ser analizados, y las tecnologías que se utilizan para el desarrollo de la aplicación móvil, el cual permite conocer un fundamento teórico del proyecto técnico.

1.1. Figuras geométricas

Las figuras geométricas se definen como “cualquier punto, línea, segmento, raya, ángulo, polígono, curva, región plana, superficie sólida, etc. Formalmente, una figura geométrica es cualquier conjunto de puntos en un plano o en el espacio” (Simmons Bruce, 2017).

De acuerdo con Zúñiga (2016, p. 50), las figuras geométricas se clasifican en:

- A dimensional (sin dimensión). En este tipo pertenece solo el punto ya que su longitud, área y volumen tiene dimensión cero.
- Unidimensional (una sola dimensión). Las figuras que pertenecen a este tipo son: rectas y curvas, ya que poseen una sola dimensión, estas figuras se extienden mediante puntos infinitos.
- Bidimensional (dos dimensiones). En este tipo se encuentra el plano o superficie, ya que posee dos dimensiones y en la cual se puede representar polígonos, triángulos, circunferencias y cuadrados.
- Tridimensional (tres dimensiones). Son las figuras volumétricas ya que se encuentran formando volúmenes o cuerpos sólidos en el espacio, las figuras que se pueden representar son: los poliedros, cilindros, esferas y conos.
- N-dimensional (varias dimensiones). En este tipo se encuentran los politopos que son figuras que tienen muchas superficies.

1.1.1. *Aprendizaje de las figuras geométricas*

Sus inicios de aprendizaje de las figuras geométricas se parten en el nivel inicial como básico, donde el niño aprende involuntariamente figuras geométricas jugando con objetos físicos, las cuales son composición de figuras geométricas como cuadrados, rectángulos, triángulos, rectas, etc., como también cuerpos geométricos que son representaciones en tres dimensiones o

composiciones de figuras geométricas ya sean cubos, pirámides, esferas, cilindros y poliedros en general, entre otros (Zúñiga, 2016, p. 48).

Las figuras geométricas que aprenden los niños no deben ser restringidas, deben aprender de acuerdo a su capacidad de adquirir nuevos conocimientos, el cual es propicio partir desde lo básico cómo en toda área, para ir incrementado el nivel de conocimiento a medida que el niño va adquiriendo sus habilidades y destrezas, durante cada día (Sobalvarro Chavarría & Camacho Álvarez, 2018, p. 9).

Durante el nivel inicial los niños aprenden reconocer figuras en dos y tres dimensiones manipulando elementos lúdicos, ya que no tienen una capacidad desarrollada de entender al nivel geométrico, pero si por entender algunas propiedades como los lados del que está compuesto una figura, por ejemplo, podrían decir que un cuadrado tiene cuatro lados. En un principio los niños describen figuras utilizando vocabulario de los atributos relacionados al: color, tamaño, textura, movimiento, material. El docente debe poner énfasis en que los niños empiecen a adquirir estos conocimientos necesarios y en caso de ser mucho más hábil hacer que el niño empiece adquirir nociones geométricas de las figuras, ya que al adquirir estos principios el niño no solo puede identificar si no ordenar, comparar y clasificar figuras (Ontario, 2005, p. 7).

1.1.2. Geometría euclidiana

Los fundamentos de la geometría euclidiana es un grupo de nociones y axiomas elementales, las nociones elementales son el punto, la línea recta y el plano (Losa *et al.*, 2016, p. 347).

La geometría euclidiana se centra en el estudio de los ángulos, áreas y volúmenes, además la manera en cómo se representa, ya que al efectuar un movimiento rígido esta mantiene sus lados y ángulos, al mismo tiempo conserva su área (Zúñiga, 2016, p. 51).




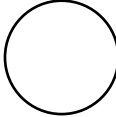
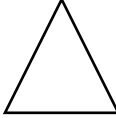
El aprendizaje de la geometría euclidiana se considera como fundamento para la formación del alumno, ya que el niño a tempranas edades están propenso a adquirir los principios de la perspectiva espacial, a través del reconocimiento y mediante las visualización directa de los dibujos y figuras en dos y tres dimensiones, estos son primordiales para el progreso de otras áreas de la geometría y la matemática en general (Morales & Rosas, 2016, pp. 249–250).

1.1.2.1. Figuras geométricas planas

Las figuras geométricas planas son parte de la geometría que se dedica al estudio de las figuras planas o en dos dimensiones, es decir están compuestas de su ancho y largo, algunas de las figuras más representativas son: cuadrado, rectángulo, círculo y triángulo además es necesario mencionar que comprende de algunos elementos como el punto, la recta, los ángulos que se originan en los vértices resultado de la unión de dos rectas (Zúñiga, 2016, pp. 51–52).

En la **Tabla 1-1**, se enseña el concepto de las figuras geométricas planas más conocidas con su respectiva figura.

Tabla 1-1: Principales figuras geométricas planas.

Concepto	Figura
La línea recta es el resultado de unir puntos infinitos que se encuentran alineados, además es de una sola dimensión (Zúñiga, 2016, p. 52).	
“Un rectángulo es un paralelogramo que tiene todos sus ángulos rectos” (Rojas Álvarez, 2016, p. 27).	
“Un cuadrado es un rectángulo que tiene todos sus lados congruentes entre sí” (Rojas Álvarez, 2016, p. 27).	
“Un círculo es la reunión de la circunferencia y su interior” (Rojas Álvarez, 2016, p. 35).	
El triángulo es el resultado de unir tres puntos con segmentos los cuales son denominados lados, internamente forman tres ángulos (Rojas Álvarez, 2016, p. 19).	

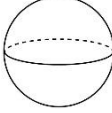
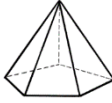
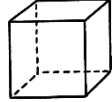

Realizado por: Guayllas, Pedro, 2019

1.1.2.2. Figuras geométricas del espacio

Las figuras geométricas del espacio son parte de la geometría que se dedica al estudio de las figuras en tres dimensiones, que también son el resultado de fusionar figuras de la geometría plana, generando algunas denominaciones como: cuerpos geométricos, cuerpos sólidos, figuras tridimensionales. Algunas figuras más representativas de la geometría del espacio son: la esfera, la pirámide, el cubo y el cilindro (Zúñiga, 2016, pp. 52–53).

De acuerdo con Zúñiga (2016, p. 50), se analiza algunos de las figuras geométricas del espacio más relevantes en la **Tabla 2-1**.

Tabla 2-1: Figuras geométricas del espacio más conocidas.

Concepto	Figura
Esfera es una de las figuras en tercera dimensión, resultado de la rotación de un semicírculo al contorno de un diámetro.	
Pirámide es uno de los poliedros irregulares formados por triángulos que se unen en un punto denominado vértice.	
Cubo es una de las figuras tridimensionales formado por cuatro cuadrados que se unen en cada uno de los lados.	
Cilindro es una figura tridimensional formado por un plano rotario alrededor de un eje, formando un círculo en sus extremos.	

Realizado por: Guailas, Pedro, 2019

1.2. Primitivas geométricas

Las primitivas geométricas son simple figuras geométricas, que en algunos casos pueden ser ensamblados para formar figuras complejas (Kaiser *et al.*, 2018, p. 4).

Definición:

*La definición se basa en la simplicidad del objeto y la frecuencia de uso según el diseño asistido por computadora (CAD), la fabricación asistida por computadora (CAM), la ingeniería asistida por computadora (CAE), los gráficos de computadora, la animación y los sistemas de diseño industrial (Gonzalez *et al.*, 2014, p. 32).*

Las primitivas geométricas son elementos básicos creados por bloques, para representar en la computadora de una manera eficiente mediante el diseño de objetos geométricos. Ayudando a la uniformidad, familiaridad, y estandarización, el cual provee soporte completo del hardware para todos los dispositivos que desean implementar (Gonzalez *et al.*, 2014, p. 32).

Al principio el concepto de primitivas geométricas se hizo uso solo para aprovechar de la mejor manera la capacidad del hardware, donde se consideraron pertinente las siguientes figuras geométricas: puntos, segmentos o líneas, y triángulos, evitando restricciones para esas figuras, pero con la ampliación a otras áreas han forzado implementar otras figuras y también ser consideradas primitivas geométricas. La faceta triangular es una manera de primitivas fácil de generar y con un soporte por completo de hardware, y muy usado para el modelado de objetos gráficos. Las librerías de software han agregado a su gama de primitivas otras figuras muy usadas,

por ejemplo, AutoCAD considera como primitiva a la puerta debido al frecuente uso en el diseño de casas (Gonzalez *et al.*, 2014, p. 32).

1.3. Dispositivo móvil

Es un aparato pequeño, con acceso a internet de manera constante o interrumpida, con la capacidad de procesamiento de datos, con memoria limitada para almacenamiento de datos, diseñadas para llevar a cabo tareas específicas, pero también puede realizar tareas generales (Baz *et al.*, 2011, p. 1).

1.3.1. Categoría de los dispositivos móviles

De acuerdo a Baz, Ferreira, Álvarez y García (2011, p. 2), los dispositivos móviles se dividen en las siguientes categorías:

- **Dispositivo móvil de datos limitados.** Dispositivo de pantalla pequeña en modo texto, utilizado para los servicios de mensajería, y acceso Wireless Application Protocol (WAP) de una manera limitada.
- **Dispositivo móvil de datos básicos.** Dispositivo de pantalla mediana con características de: menús, acceso a correo, mensajería, lista de direcciones, navegador web, entre otros. Ejemplo Smartphone.
- **Dispositivo móvil de datos mejorados.** Dispositivos de pantalla mediana o grande (mayor a 240 x 120px), con las mismas características de un dispositivo de datos básico, pero agrega aplicaciones nativas, incluye un sistema operativo.

1.3.2. Asistente Digital Personal (PDA)

Es un computador de mano que realizan tareas de computador, pero portátil. En un principio contenían aplicaciones de agenda electrónica, lista de contactos, recordatorios y blog de notas, pero se han desarrollado, y actualmente contienen: juegos, opción de ver videos, realizar documentos, reproducir audio y navegar por la web (Baz *et al.*, 2011, p. 2).

1.3.3. Teléfonos móviles

Es un teléfono inalámbrico con las mismas características de un teléfono de línea fija, pero móvil es decir capta los datos a través de ondas en vez de la conexión cableada alcanzando la ventaja portable. Su servicio principal es de la comunicación por voz, pero contiene otras funciones como:

mensajería, juegos, agenda, cámara fotográfica, acceso a internet, reproductor de audio y video, en algunos casos GPS (Baz *et al.*, 2011, p. 3).

1.3.4. Smartphone (teléfono inteligente)

Es un teléfono móvil además posee mejoras en todas sus características como: cámara de mayor resolución, muy similar a un ordenador, permite instalar aplicaciones, la mayor característica que destaca por la pantalla táctil. Se remota sus inicios en 2007 (Baz *et al.*, 2011, p. 3).

1.3.5. Android

Es un sistema operativo desarrollado para dispositivos móviles, corresponde a Google y es respaldado por Open Handset Alliance. El sistema operativo cuenta con en el kernel de Linux (Moscoso Dominguéz, 2014, p. 5).

Android es un sistema operativo de código abierto, esto es una gran ventaja para una inmensa cantidad de desarrolladores que se dedican al desarrollo de aplicaciones para esta plataforma, por otra parte, la ayuda es muy significativa ya que existe una gran comunidad de desarrolladores que optan por liberar código y de esa manera apoyarse entre desarrolladores (Moscoso Dominguéz, 2014, p. 6).

Android está compuesto por una pila de software principalmente el sistema operativo, aplicaciones claves y middleware. Android se basa en el núcleo de Linux, su desarrollo está dividido en capas el cual se divide en: capa de aplicación, capa de aplicación Framework, capa de librerías y el Kernel. (Francis, 2008, p. 2).

1.3.6. Sistema operativo Android

El sistema operativo Android es una plataforma con gran auge de crecimiento en el mercado del desarrollo de aplicaciones móviles, con gran éxito en el mercado, Google ha desarrollado su tienda de aplicaciones Google Play, la misma que viene instalado por defecto en su sistema operativo.

El Framework del sistema operativo corre en la máquina virtual de Java. Al igual que otros sistemas operativos contiene librerías escritas en el lenguaje de programación C, con el Framework OpenCore, la interfaz gráfica basado en la librería OpenGL ES, la base de datos

sQLite, renderizado en el motor WebKit, el motor gráfico SGL, SSL y la biblioteca C Bionic (Guamán Campoverde & Cuvi Ocaña, 2015, p. 26).

La arquitectura del sistema operativo Android se divide en 5 capas **Figura 1-1**, la primera capa corresponde al Kernel del sistema operativo, que se encuentra basado en el núcleo de Linux, la segunda capa corresponde a las librerías, la tercera capa corresponde a la máquina virtual de Android, la cuarta capa corresponde al Framework de aplicaciones, esta capa es la explotada por los desarrolladores, para crear aplicaciones móviles, y la última capa es de aplicación, encargada de interactuar con el usuario final con todas aquellas aplicaciones que tiene instalado el usuario (Moscoso Dominguéz, 2014, p. 6).

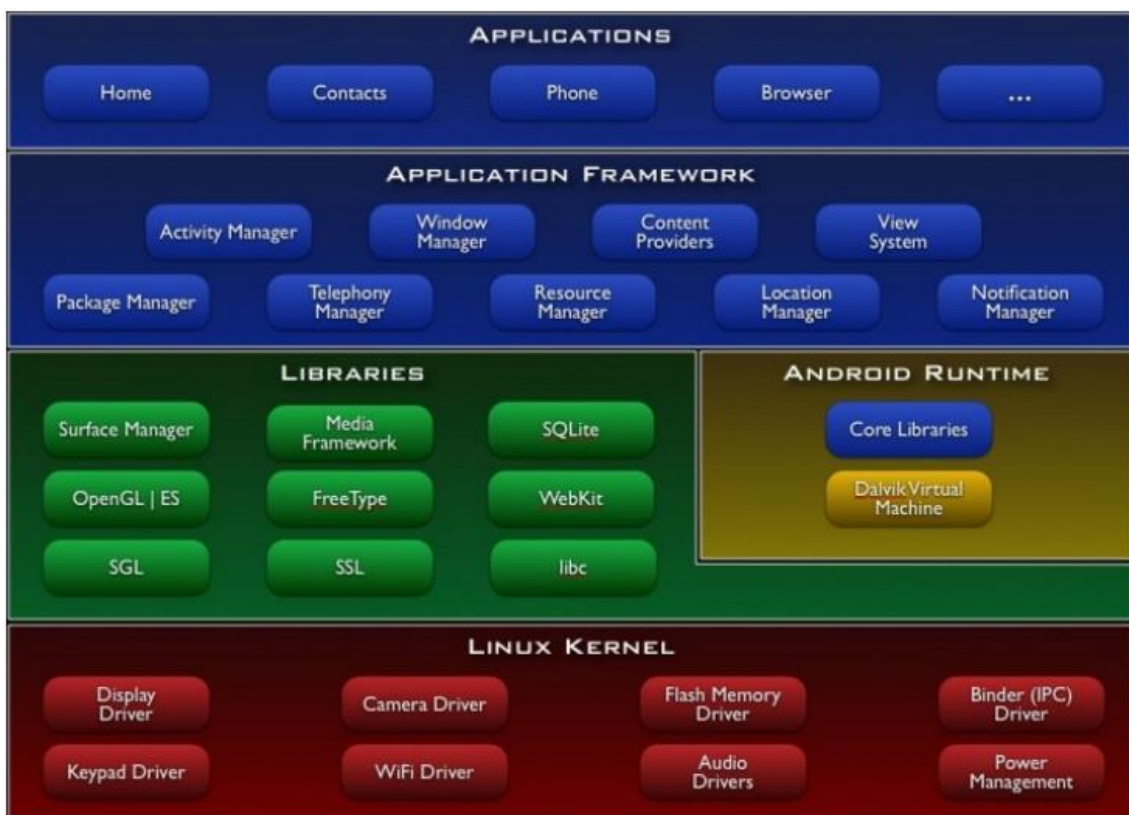


Figura 1-1: Arquitectura del Sistema Android.

Fuente: (Moscoso Dominguéz, 2014, p. 6)

1.4. Aplicación móvil

Una aplicación móvil o simple denominada apps, son programas creados para correr en un dispositivo móvil, smartphone o table y que su distribución de las aplicaciones para Android se realiza en la tienda de aplicaciones Google Play, ya que estas pueden ser desarrolladas por empresas o individuos (Santamaría Puerto & Hernández Rincón, 2015, p. 600).

El termino aplicación móvil se ha popularizado en el mundo entero. El desarrollo de las aplicaciones móviles sigue el crecimiento inesperado, ya que el uso de los teléfonos por parte de los usuarios se frecuenta en el uso de las aplicaciones. La demanda de las aplicaciones ha hecho que se desarrollen más herramientas que sean tan sofisticadas, para el propio desarrollo de las aplicaciones móviles, ya que hoy en día existen herramientas, librerías y Framework que permiten el desarrollo rápido, fácil y más productivo. Estas aplicaciones ofrecen una variedad de servicios a los clientes, ya que ofrecen funcionalidades como: GPS, juegos, banca, tickets online entre otros (Poudel, 2013, p. 6).

El objetivo de tener una aplicación es tener un compromiso interactivo con los usuarios, o proporcionar una aplicación que requiera trabajar más como un programa de computadora o un sitio web. Hay aplicaciones libres como pagadas, como también hay una estrategia de que al principio son gratis, pero si requiere mayor funcionalidad deben pagar para actualizar esa funcionalidad. En un principio las aplicaciones fueron creadas para ofrecer más información y propuestas de productividad conteniendo: correo, calendario, contactos, calculadora e información de tiempo. La magnificación rápida en la tecnología y mayor uso ha hecho que la propuesta se expanda a otras categorías como: redes sociales, video chats, automatización industrial, servicios basados en localización, desarrollo físico y aplicaciones médicas (Inukollu *et al.*, 2014, pp. 16–16).

1.5. Kotlin

Es un lenguaje de programación estáticamente tipado, moderno, compatible con Android, inspirado en: C#, Groovy, Scala, Swift, y otros muchos lenguajes, que arregla muchos problemas de Java, tales como excepciones de punteros nulos, o excesiva verbosidad de código. El lenguaje de programación Kotlin fue diseñado por profesionales de JetBrains, basado en el análisis profundo de otros lenguajes de programación. Kotlin intenta evitar errores fuertes de otros lenguajes de programación, haciendo que sea una de las ventajas más fuertes sobre otros lenguajes (Marcin Moskala, 2017, p. 8).

Kotlin más que un lenguaje de programación es un sueño que hace realidad para muchos desarrolladores, en comparativa de lo tedioso que es Java. Para desarrolladores de Android es más interesante y muy fácil de aprender, el cual reduce lo complejo del código y ayuda a manejar su código de una manera menos estridente (Jangid, 2017, p. 259)

Kotlin es una manera de desarrollo fácil de aplicaciones Android, rápido y mucho más agradable. Este lenguaje de programación cambiara la manera de pensar sobre la manera de escribir código

y resolver problemas comunes de programación. Kotlin toma el desarrollo de aplicaciones para todo nivel, que provee calidad, seguridad e impulsa el rendimiento rápido de desarrollo. Oficialmente el soporte de Kotlin para la plataforma Android fue anunciado en la conferencia de Google IO en el año 2017, pero el lenguaje de programación Kotlin ha estado en desarrollo desde el 2011. Tiene una comunidad muy activa de desarrolladores y la adopción a Android está creciendo rápidamente. Se describe a Kotlin como: seguro, expresivo, conciso, versátil, y una herramienta muy amigable, es interoperable con Java y JavaScript (Marcin Moskala, 2017, pp. 7–8).

1.5.1. Características de Kotlin

Marcin Moskala (2017, pp. 9–8), menciona las características más importantes de Kotlin, las cuales son:

- **Seguridad.** Kotlin ofrece característica de seguridad en términos de anulabilidad e inmutabilidad, el tipo de expresión es reconocido en tiempo de compilación. El compilador puede verificar que cada propiedad o método que está intentando acceder a una instancia de una clase si actualmente existe. Esto es común en Java que también es estáticamente tipado, pero incomodo en Java, en cambio Kotlin es mucho más estricto (seguro). En caso de ser nulo se debe expresar explícitamente, que la variable puede guardar valor nulo, esto permite hacer que el sistema no falle en tiempo de ejecución.
- **Fácil depuración.** Los errores pueden ser detectados mucho más fáciles durante la fase de desarrollo, Kotlin ofrece una buena manera de trabajar con datos inmutables. Por ejemplo, puede distinguir dato mutable (lectura - escritura) e inmutable (solo lectura).
- **Conciso.** La verbosidad de Java fue eliminada. Necesita menos código para lograr una tarea común, en una comparación entre Java 8 y Kotlin, como resultado el código es fácil de leer y entender (expresivo).
- **Interoperabilidad.** Kotlin está diseñado para reducir trabajo de lado a lado con respecto a Java con el proyecto cross-language. Las clases de Kotlin puede ser directamente instanciado y transparentemente referenciados dentro del código de Java y viceversa, una semántica especial. Esto permite incorporar Kotlin en un proyecto de Android existente y usar Kotlin fácilmente con Java.
- **Versatilidad.** Que se puede incorporar en muchas plataformas, incluyendo aplicaciones móviles (Android), aplicaciones lado del servido (Backend), aplicaciones de escritorio, correr sobre la Web (Fronddend), y algún sistema desarrollado (Gradle).

Otras de las características técnicas que plantea Jangid (2017, pp. 257–258), son las siguientes:

- **100% Interoperable.** Uno puede llamar un código escrito en Java desde Kotlin y viceversa de una manera muy fácil, sin la necesidad de convertir todo un proyecto ya iniciado de un lenguaje de programación a otro.
- **Libre de excepciones de punteros nulos.** En Kotlin todas las variables no son nulas, y en el caso de requerir declarar una variable nula es necesario especificar con el símbolo “?” al final la de declaración, esto es una ventaja importante con respecto a Java donde declara una variable en principio nulo y posterior esto genera error.
- **Menor verbosidad.** Uno de los problemas que soluciona el lenguaje de programación Kotlin es la verbosidad de código, infiere ciertos fragmentos como la declaración del tipo de dato, no requiere del punto y coma para terminar una sentencia de código, es claro y conciso reduciendo el tiempo repetitivo en escribir código y aumentando la productividad.
- **Conversión inteligente.** No se requiere conversión explícita de los operadores porque el compilador de Kotlin inserta conversiones automáticamente donde sea necesario. Para verificar si un objeto se ajusta a un tipo dado en tiempo de ejecución, se puede usar el operador "is".
- **Declaraciones de desestructuración.** Trata un solo objeto como un conjunto de múltiples variables.

1.5.2. *Propiedades de Kotlin*

Las propiedades del lenguaje de programación Kotlin, se basan en las características ya analizadas de este lenguaje. Las propiedades que se definen son sintaxis para la programación de acuerdo con lo que especifica Marcin Moskala (2017, pp. 8–9):

- La propiedad `var` define una variable mutable es decir que modifica su comportamiento después de haber definido, por ejemplo, al definir un objeto de tipo persona permite la inicialización posterior de sus atributos que los contiene como nombre, edad, peso, y otros de sus atributos definidos en el objeto de tipo persona, esta propiedad también es usado para recibir parámetros ya que sus los valores de los parámetros siempre van a cambiar, a estas variables se les denomina también variables de lectura y escritura.
- La propiedad `val` define lo contrario de `var` ya que al definir una variable obliga asignar a todos sus atributos un valor y esto no puede modificar posterior, con respecto al ejemplo anterior un objeto de tipo persona debe tener un nombre, edad, peso, y si queremos modificar posterior no se permite, por ende, se les denominan variables de solo lectura.
- “`class`” más el nombre de la clase definen una clase, quien contiene atributos para definir características y métodos que determinan el comportamiento de esas características que las contiene.

- “fun” define una función o un método dentro de una clase.

1.6. OpenGL ES

La complejidad que trae la manera de visualizar gráficos ya sea 2D o 3D en la pantalla de los dispositivos, ha hecho que los fabricantes de pantallas se unan, para estandarizar una librería que permita seguir la misma guía para todos, el cual fue liderada por Silicon Graphics Inc. (SGI), en el año 1992, del cual surge la librería OpenGL, en el cual se logra estandarizar el acceso al Hardware, trasladar a los fabricantes de pantallas la responsabilidad de permitir el soporte de las interfaces que se genere mediante OpenGL, delegar las funciones para ventanas al sistema operativo y lograr que todos hablasen el mismo lenguaje (Debuire Enríquez, 2008, p. 127). su desarrollo ha pasado de versión en versión con el fin de mejorar cada vez más la eficiencia que trae esta librería.

OpenGL ES. Es una variante de la librería OpenGL, utilizada para escribir aplicaciones que generan gráficos 2D o 3D, para dispositivos móviles, PDAs y consolas de juegos. La librería permite la generación de gráficos complejos a través de primitivas geométricas simples y convertirlas en píxeles, donde todo este proceso es denominado como máquina de estados. La mayoría de los fabricantes de tarjetas de video utilizan este estándar (Madueño Ortega, 2013, p. 53).

OpenGL ES, el estándar gráfico API para dispositivos móviles y sistemas embebidos. Virtualmente cada píxel en la pantalla es generado por OpenGL ES (JungHyun, 2018, p. 13). La librería es usada por una amplia variedad de industrias de gráficos estandarizados API. Ha sido portado a los mayores sistemas, y pueden ser accedidos desde un número diferente de lenguajes de programación, porque trabaja a través de diferentes plataformas, la mayor parte de la librería opera en modo inmediato, que significa que cada llamada de función a un comando resulta ser enviada directamente a la Unidad de Procesamiento Gráfico (GPU). En algunos casos se puede generar retrasos cuando se realiza, una transformación, iluminado o texturizado (Mounth, 2013, p. 11).

1.6.1. Características de OpenGL ES

Según Debuire Enríquez (2008, pp. 124–125), se analiza las características genéricas que conlleva la librería OpenGL ES, las cuales se detallan en los siguientes ítems:

- En la programación es una librería para la generación y modelado de gráficos en 3D, a una velocidad muy rápida, en base a su estándar planteado con una sofisticada calidad visual.
- OpenGL ES proviene de las siglas en Inglés Open Graphics Library for Embedded System, esto significa que la librería puede permitir contribuciones de otras empresas que ayuden su desarrollo.
- No es un lenguaje de programación, es un set de funcionalidades precompiladas en C, donde un desarrollador escribe código haciendo uso de la librería, para acceder de una manera directa a la generación y modelado de gráficos.
- Es creado para el uso eficiente del Hardware, a pesar de que el dispositivo no cuente con Hardware sofisticado, ayuda de una manera óptima en la manipulación de gráficos en 3D. No obstante, su implementación es por software, durante la programación.
- La librería requiere construir un modelo de una escena en base a las primitivas geométricas (puntos, líneas y polígonos), para posterior efectuar un renderizado del gráfico que va a mostrar en la pantalla del dispositivo.
- La librería es utilizada para una variedad de propósitos, como: el Diseño Asistida por Computadora (CAD), programas de arquitectura, para crear películas de cine, la simulación, los videos juegos, entre otros.

De acuerdo con Choudhari (2015, p. 1755), se analiza otras características de la librería OpenGL ES, las cuales son las siguientes:

- **Diseño independiente del Hardware.** Las funciones creadas por esta librería se ejecutan en cualquier plataforma, siendo una ventaja para los desarrolladores a la hora de utilizar esta API.
- **Rendimiento.** Su rendimiento es muy rápido ya que en los dispositivos se ejecuta al instante, y no tarda a la hora de renderizar gráficos.
- **Calidad de imagen.** Otro de los objetivos de los diseñadores de la API era garantizar la calidad de imagen. Lo esencial es que cada píxel que se dibuja en la pantalla se vea con la mejor calidad.
- **Estable.** Desde un principio se centraron en correr esta API sin interrupciones, al efectuar cualquiera modificación pasa por una etapa de pruebas, para contralar el mejor funcionamiento y posterior anunciar a los desarrolladores su adopción.
- **Confiable y portable.** Mencionado las características de rendimiento y calidad de imagen ha hecho que los desarrolladores adopten confianza en esta API, para seguir utilizando, en el ambiente que lo requiera.

- **Escalable y extensible.** La librería cada cierto tiempo se ve reflejado en las modificaciones de las versiones, buscando mejorar cada vez e incursionar a los desarrolladores su adopción.
- **Buena documentación.** Cada plataforma que lo adopta se centra en utilizar cada función estricta que el estándar lo establece. En caso de adoptar nuevas funciones se tiene una estricta información de las nuevas versiones que el dispositivo lo adopta.

1.6.2. Propiedades de la librería

Para estudiar las propiedades de la librería OpenGL ES, se basa en las características ya analizadas de esta librería, centrándose en definir las sintaxis para la programación, de acuerdo con lo que describe Debuire Enríquez (2008, pp. 124–125):

- `glEnable` y `glDisable` son propiedades esenciales para habilitar cualquier otra propiedad en OpenGL ES.
- `glLine` es una propiedad primitiva para dibujar una línea.
- `glTriangle` es una propiedad primitiva para dibujar un triángulo.
- `glColor` es una propiedad que se encarga de poner color a la figura, el cual se basa en el sistema de colores RGB, el color en una figura se puede manipular de distintas maneras, como: degradados, reflexión, punto de luz, entre otros.
- `glClearColor` es otra propiedad que muestra el color de fondo, en el resto de la vista creada para mostrar figuras con OpenGL ES.
- `glClear` limpiar el color cada vez que existe una interacción con la figura, o cambia a una nueva figura.
- `glViewport` define donde empieza la vista creada para mostrar figuras con OpenGL ES, dado sus coordenadas en el plano X y plano Y, además define el ancho y largo.
- `glCreateProgram` inicializa una vista vacía para definir el modelo de una figura, y posterior renderizar.
- `glCreateShader` determina el número de caras que contiene la figura para visualizar en la pantalla del dispositivo móvil.
- `glShaderSource` determina el origen de las caras es decir donde empieza los vértices de cada primitiva, recibiendo como parámetro el número de caras que va a visualizar.
- `glCompileShader` organiza todas las caras listas para el proceso de renderizado.
- `glUseProgram` inicializa el programa para el renderizado.
- `glGetAttribLocation` determina la posición donde se va a visualizar la figura.
- `glEnableVertexAttribArray` habilita el renderizado triangular de vértices.

- glVertexAttribPointer prepara los datos del triángulo de coordenadas.
- glGetUniformLocation determina la posición uniforme de cada fragmento.
- glUniform4fv pasa la proyección y la vista de la figura transformada.
- glDrawArrays pasa una matriz de primitivas a renderizar, con el parámetro del tipo de primitiva y el contador de vértices.
- glDisableVertexAttribArray deshabilita el array de vértices una vez ya renderizado la figura.

1.6.3. Funcionamiento de la librería

Las dos principales tareas que son requeridas para crear una escena de imagen en tres dimensiones, es el modelado y el renderizado. La tarea del modelado es genera el modelo, que es la descripción de un objeto que está siendo ser usado por el sistema gráfico. La segunda tarea, renderizado, toma modelos como entrada y genera valores pixelados de la imagen. Donde OpenGL es el concerniente principal con el renderizado del objeto. La arquitectura OpenGL está enfocado primordialmente en renderizar modelos poligonales (McReynolds & Blythe, 2005, p. 3).

La librería se puede utilizar desde la generación de una primitiva geométrica hasta generar una escena compleja, en la **Figura 2-1**, se muestra el proceso para visualizar un gráfico en la pantalla.

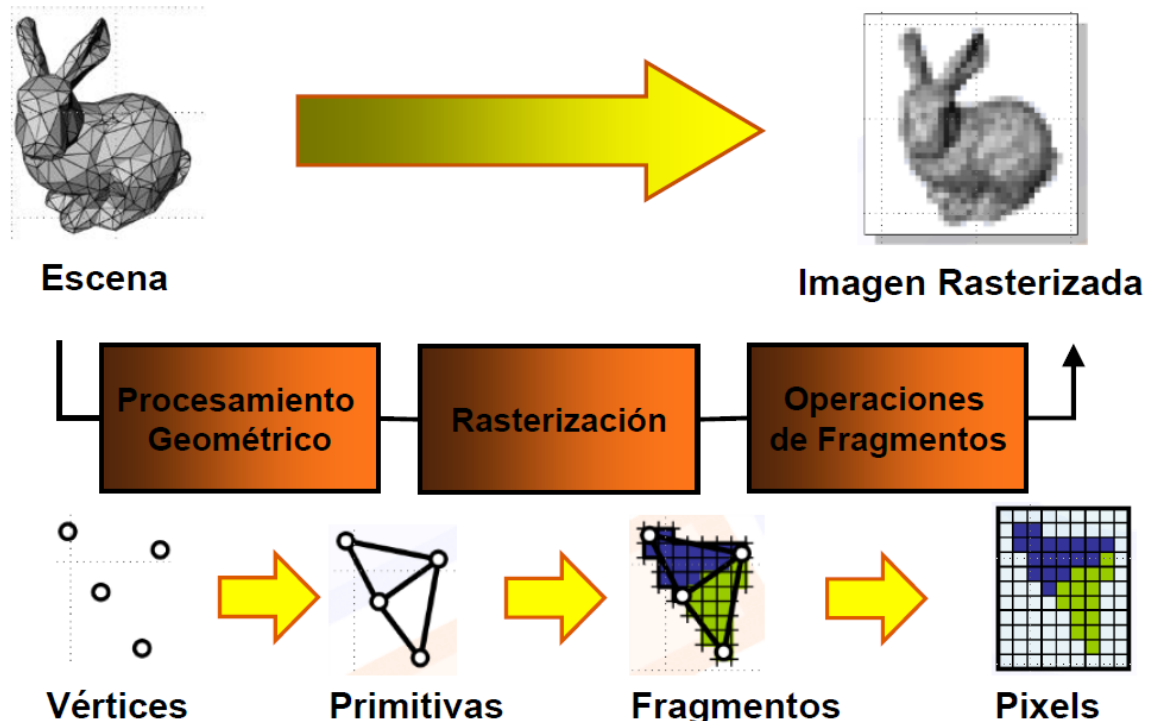


Figura 2-1: Proceso para renderizar una figura con la librería OpenGL ES.

Fuente: (Moscoso Domínguez, 2014, p. 6)

En base a la **Figura 2-1**, que plantea Madueño Ortega (2013, pp. 54–55), detalla los siguientes pasos a seguir para renderizar un gráfico:

- **Procesamiento geométrico.** Se crea vértices y coordenadas en el gráfico, las cuales van a ser transformadas, y teniendo cada uno de los vértices se ensambla cada vértice con las primitivas geométricas que requiera ser representado ya sea punto, línea, o polígono mediante una matriz de proyecciones, también se realiza la operación de transferencia de píxeles, donde se realiza las operaciones de transformación, escalado y recortado, la cual se almacenan en la memoria y se rasteriza a fragmentos.
- **Rasterización.** Se convierte los datos de las transformaciones tanto de cada uno de los píxeles y los geométricos en fragmentos, además agrega textura, color, profundidad, el ancho de línea, y el tamaño de línea. Haciendo que cada fragmento generado sea un píxel, y el conjunto de píxeles genere la figura en la pantalla.
- **Operaciones de fragmentos.** En esta etapa se busca corregir ciertos aspectos, como: texturizado, difusión, suavizado y sombreado, al final estas operaciones dan como resultado un conjunto de buffers que es la manera de almacenar datos temporales hasta que el sistema se encargara de mostrar cada píxel con un color en la pantalla.

1.6.4. Uso de la librería

Durante el uso de la librería es importante destacar las dos clases OpenGL ES necesarios para trabajar en la generación de gráficos que lo describe Román (2015, p. 33):

- `GLSurfaceView`. Es una clase de tipo vista que provee eficiencia y rapidez a la hora de generar imágenes cuando este corriendo la aplicación.
- `GLSurfaceView.Renderer`. Es una interface que se encarga de llamar la API de OpenGL para renderizar el marco. Esta hereda de la clase `Render` para manipular métodos a la hora de la generación e interacción de los gráficos.

Según Zhuanga (2015, p. 2479), es necesario mencionar los tres métodos obligados implementar por la clase `Render` para manipular el despliegue de los gráficos:

- *onSurfaceCreated*. Método que es llamado cuando crea la clase `GLSurfaceView`. De modo que la operación inicial de OpenGL fue puesta aquí.
- *onDrawFrame*. Método que cada vez que dibuje `GLSurfaceView` llama a este método, para que la operación de dibujo basada en OpenGL ES a través de Java Native Interface (JNI) se pusiera aquí y luego la pantalla visualiza la imagen o figura.

- *onSurfaceChanged*. Método de devolución de llamada cuando se cambia el tamaño de la pantalla de visualización o se cambia la orientación de la pantalla.

1.6.4.1. Coloreado de vértice

Son programas que llaman una vez por cada vértice en una escena. Se encarga de efectuar algunos cálculos como iluminación, transformaciones de geometría, etc., averiguar la posición final del vértice, y también pasa algunos datos al fragment shader. Además Choudhari (2015, p. 1756) menciona los parámetros requeridos por esta función:

- *Atributos*: vector de datos suministrados por cada uno de los vértices.
- *Uniformes*: constantes que especifican el sombreado de vértices.
- *Muestreadores*: un tipo definido de uniformes que simbolizan texturas utilizada para el sombreado de vértices.
- *Programa de sombreado*: fragmento de código de la aplicación sombreado de vértices o ejecutable que representa las operaciones que se efectúan en el vértice.

1.6.4.2. Coloreado de fragmento

Los fragment shaders son programas que se encargan de calcular la luminosidad, pero su más importante trabajo es colorear cada píxel. El sombreador de fragmentos, se ejecuta para cada fragmento generado por la etapa de rasterización. Los valores de salida son variables del sombreado de vértices creadas por la componente de rasterización para cada fragmento a través de la interpolación (Choudhari & Ranjankar, 2015, p. 1757).

1.7. Reconocimiento de voz

La comunicación hombre máquina es un mecanismo del reconocimiento de voz. La finalidad de los sistemas de reconocimiento de voz permite a un sistema reconocer la voz de una persona en todo instante sin ser repercutidos por factores externos (Apolo Díaz & Coba Castillo, 2017, pp. 12–13).

Un patrón de voz es el producto de la pronunciación de una palabra, la cual produce una forma de onda. Los patrones de voz son diferentes en cada persona, así como las huellas dactilares. Por ende, la identidad de la persona se podría también realizar por su patrón de voz (Rashid *et al.*, 2008, p. 900).

Hoy en día los comandos de voz ayudan realizar actividades de una manera sencilla. La mayoría de las empresas tecnológicas continúan desarrollando sistemas de voz para la asistencia en sus servicios entre las cuales se encuentra, Google Assistant, Cortana de Microsoft, Siri entre otros, de esta manera los asistentes de voz son cada vez más personales en los dispositivos móviles que vienen incorporando en la mayoría de los dispositivos. Estos sistemas inteligentes ayudan realizar tareas como: búsquedas, controlar sistemas, llamadas de voz, entre otros (Morales Montero & Yáñez Jácome, 2018, p. 46).

1.7.1. Proceso del reconocimiento de voz

El proceso que sigue el reconocimiento de la voz conlleva utilizar un software para establecer un canal de comunicación hombre-máquina. De acuerdo con Apolo y Coba (2017, p. 13) se sigue el siguiente proceso para identificar el patrón de voz:

- Preprocesamiento: son todos aquellos procesos aptos para recibir la entrada de la voz.
- Reconocimiento: determina la voz recibida.
- Comunicación: emite el patrón reconocido del hardware al software para la siguiente actividad a ser utilizada.

1.8. Calidad de Software

Según la Organización Internacional de Normalización (ISO) la calidad de software se define como “La totalidad de características de un producto, proceso o servicio que cuenta con la habilidad de satisfacer necesidades explícitas o implícitas” en base a esta definición se puede decir que la calidad de software satisface necesidades de un usuario de una manera parcial o total. Por otra parte, se menciona que el software tiene calidad cuando cumple las perspectivas del usuario en cuanto a: funcionalidad, que el software este hecho para cumplir un propósito, ejecución, que el software sea práctico, confiable que realice la tarea que debe, disponible que el software funcionando bajo cualquier acontecimiento.

La calidad de software hace referencia al nivel de desempeño, de las importantes características con las que posee un sistema computacional durante su periodo de desarrollo, estas características buscan garantizar que el cliente adquiera confianza, de esa manera ayude frente a la funcionalidad y la eficiencia de que el sistema está bien construido, y por ende garantiza su uso (Callejas Cuervo *et al.*, 2017, p. 237).

Uno de los principios de la calidad del software es extraer los requerimientos y las cuales al crear un producto software cumplan las expectativas del cliente, lo que conlleva a tener calidad en el producto y el proceso, donde las metas en cuanto a la calidad del producto van a determinar las metas a establecer para la calidad del proceso de desarrollo (Mascheroni *et al.*, 2012, p. 656).

Para medir la calidad del software se realiza mediante los atributos de calidad de software, las cuales se pueden dividir en sub-atributos, para evaluar características más específicas. El estándar ISO 25000 es un estándar centrado en evaluar el producto software, el cual plantea 6 atributos de calidad de software a evaluar, las cuales son: usabilidad, fiabilidad, funcionalidad, portabilidad, mantenibilidad y eficiencia, donde cada atributo de calidad de software cuenta con métricas de calidad de software a evaluar y las cuales pueden ser cualitativas o cuantitativas. La calidad de software se puede medir de acuerdo con las necesidades del proyecto, como también aplicar la norma, modelo o estándar de acuerdo con esa necesidad, la evaluación de calidad de software se pueden aplicar en el proceso de desarrollo del software, el producto o la calidad de uso (Callejas Cuervo *et al.*, 2017, p. 257), estos mismo autores analizan cada uno de los ellos de la siguiente manera:

- Calidad a nivel de proceso. En base a este enfoque, para evaluar la calidad de software se incursiona desde un principio para posterior llevar a cabo la evaluación en cada etapa de desarrollo del software, donde se pueda llevar control y seguimiento en cuanto a la calidad, y de esa manera verificar los riesgos y poder continuar con el proyecto sin interrupciones, por el contrario si se deja de llevar el control eso generara deficiencia y reduce la calidad de software, no solo repercutiendo en el proceso sino que también en el producto.
- Calidad a nivel de producto. El principio de este modelo es evaluar el producto para determinar si cumple con las especificaciones iniciales, donde se aplica métricas de evaluación tanto internas como externas, el cual determinar la satisfacción del cliente en el cumplimiento de los requisitos definidos al principio, por ende, también repercute en el proceso de desarrollo que tuvo buena calidad.
- Calidad en uso. Hay que manifestar que la usabilidad y la calidad de uso para muchos ha estado ligado, pero en sí no tienen el mismo significado donde la calidad abarca la usabilidad y esta evalúa los atributos respecto al usuario final con relación a la eficacia, eficiencia y satisfacción del usuario de acuerdo con la norma ISO 9241.

1.8.1. Norma ISO 9241

La norma ISO 9241 se centra en evaluar la usabilidad que es uno de los requerimientos ergonómicos, donde expresa que un producto software puede alcanzar un nivel de usabilidad de

acuerdo con los objetivos alcanzados por los usuarios sin tener las dificultades algunas a la hora de realizar una tarea con efectividad, eficacia y satisfacción en determinadas tareas que se evalúan (Mascheroni *et al.*, 2012, p. 656).

1.8.1.1. Norma ISO 9241-11

La norma ISO 9241-11 es un estándar que extiende de la norma ISO 9241, que evalúa el atributo de calidad de software correspondiente a la usabilidad para dispositivos móviles, las metas que plantea este estándar es evaluar las métricas en cuanto a la eficiencia, eficacia y satisfacción, el cual busca asegurar el producto software sea usable cuando el usuario pueda realizar sus tareas y sienta la satisfacción confortable y acepte su uso, de esa manera se puede determinar el grado de usabilidad de un determinado software, por otra parte para evaluar este atributo se puede realizar de dos maneras, la primera con una inspección donde intervienen evaluadores expertos o también denominados heurísticos y por otra parte mediante una encuesta donde se realiza al usuario real o también denominados empíricos (Arthana *et al.*, 2019, p. 4), además estos autores analizan las métricas de la siguiente manera:

- Eficiencia. Que el usuario pueda realizar tareas específicas.
- Eficacia. Hace referencia al tiempo que toma en realizar una tarea y determina si logra cumplir el objetivo, el cual puede influir el grado de utilizar otras aplicaciones relacionadas, haciendo que reduzca el tiempo, y de esa manera completar la tarea en el menor tiempo.
- Satisfacción. Se refiere a la conformidad y aceptación del uso del producto software. En cuanto a medir por lo general se realiza a través de una serie de preguntas que el usuario responde después de utilizar el sistema.

CAPÍTULO II

2. MARCO METODOLÓGICO

En este capítulo se detalla los métodos, técnicas y la metodología que se sigue en el desarrollo del proyecto técnico, se utiliza SCRUM una metodología ágil y flexible para desarrollar proyectos de desarrollo de software, muy conocido por recomendaciones de buenas prácticas a seguir, además permite gestionar roles entre los miembros del equipo que plantea esta metodología, haciendo muy eficiente en el recorrido del desarrollo del proyecto.

2.1. Descripción del lugar de estudio

La Unidad Educativa Intercultural Bilingüe Monseñor Leónidas Proaño, es una institución educativa que oferta la formación al nivel inicial, básica y bachillerato, el cual cuenta con 3 sedes dentro de la ciudad de Riobamba de la provincia de Chimborazo. La sede número 2 se crea en noviembre del 2014, el cual se encuentra ubicada en el sector tierra nueva entre las calles Dionicio de Alcedo y Esteban Marañón, donde se imparte la formación para los niños del nivel inicial. Es el lugar donde se realiza el estudio.

2.2. Alcance

La aplicación móvil se encuentra organizado en 4 módulos necesario para su desarrollo las cuales son: módulo de creación de las figuras geométricas en 2D, módulo de creación de figuras geométricas en 3D, módulo de movimiento de figuras geométricas y módulo de reconocimiento de patrones de voz, el cual se visualiza en la **Figura 1-2**, donde se especifica lo que contiene cada módulo.

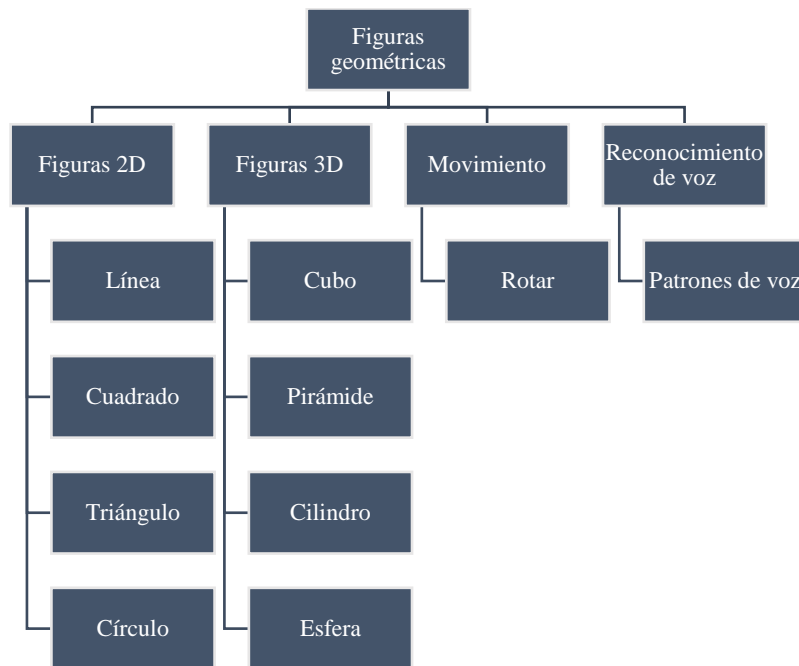


Figura 1-2: Esquema funcional de la aplicación móvil.
 Realizado por: Guailas, Pedro, 2019

2.3. Tipo de estudio

Se efectúa un estudio de tipo aplicativo, que aplica nuevos conocimientos técnico y existentes para la solución del problema planteado, en beneficio de los niños, quienes son usuarios de la aplicación móvil.

En cuanto a la ingeniería de software, se utiliza una metodología ágil para el proceso de desarrollo del software, como también se utiliza técnicas como la entrevista para extraer los requerimientos de parte de los clientes, además se utiliza la observación para conocer, cuando los niños adquieren conocimiento con materiales lúdicas, y de esa manera poder ayudar con la solución que se plantea.

2.4. Métodos

2.4.1. Método descriptivo

Se utiliza el método descriptivo para realizar la descripción de la mayor parte del contexto escrito del documento, mayormente se usa en el marco teórico para describir las características, definiciones y ventajas con respecto a los temas investigados relevantes al tema principal del proyecto técnico.

2.4.2. Método analítico

El método analítico permite recabar lo relevante de un tema general, es decir permite dividir en partes el tema, para de esa manera analizar las partes por separadas ayudando entender mejor el contexto de estudio.

2.5. Técnicas

2.5.1. Entrevista

La entrevista se utiliza para recabar información a los maestros de la escuela, de esa manera extraer los requerimientos necesarios de la aplicación móvil, de una manera verbal y entender mejor lo que precisamente requieren.

2.5.2. Observación

Esta técnica permite observar, como los estudiantes aprenden identificar figuras mediante el uso de objetos físicos, de una manera habitual que realizan sus tareas, también ayuda durante la evaluación del uso de la aplicación móvil, para ver como interactúan los niños con la aplicación móvil desarrollada.

2.5.3. Encuesta

Esta técnica permite evaluar la usabilidad de la aplicación móvil, donde se planta un cuestionario de preguntas en base a las características que plantea la norma que se utiliza, la encuesta se lleva a cabo preguntándoles verbalmente, y anotarles las respuestas que contestan, para posterior efectuar el análisis correspondiente.

2.6. Análisis previo al desarrollo del proyecto

Durante esta fase se lleva a cabo un breve análisis para el desarrollo del proyecto, en el cual se efectúa el estudio de factibilidad tanto técnica, operativa y económica, además se analiza los riesgos, que puede generar durante el desarrollo del proyecto.

2.6.1. Estudio de factibilidad

Durante el estudio de factibilidad se viabiliza, si es conveniente llevar a cabo el proyecto, teniendo en cuenta cada uno de los recursos que se tiene disponible, además lo que se requiere para llevar a cabo el mismo.

2.6.1.1. Factibilidad Técnica

En este ámbito se analiza los recursos tanto de Hardware como Software que son requeridos durante el desarrollo de la aplicación móvil, para lo cual en la **Tabla 1-2**, se especifica el Hardware requerido.

Tabla 1-2: Hardware requerido

Cantidad	Equipo	Características
1	Laptop para desarrollo	Intel® Core™ i7 7th Generación
1	Dispositivo móvil	Samsung J5 Prime

Realizado por: Guailas, Pedro, 2019

En la **Tabla 1-2**, se detalla que se requiere de una laptop con características admisibles para desarrollar la aplicación móvil, además el dispositivo móvil para efectuar las correspondientes pruebas, de cada una de las funcionalidades.

En cuanto al software requerido se detalla en la **Tabla 2-2**.

Tabla 2-2: Software requerido

Nombre	Descripción	Estado
Android Studio	IDE utilizado para el desarrollo de la aplicación móvil.	Software libre
Android Emulator	Software utilizado para emular el funcionamiento de la aplicación móvil.	Software libre

Realizado por: Guailas, Pedro, 2019

En la **Tabla 2-2**, se especifica el software requerido para el desarrollo de la aplicación, el cual requiere del IDE Android Studio para el desarrollo de la aplicación móvil, además del Android Emulator que es un emulador del funcionamiento de las aplicaciones móviles.

El análisis realizado lleva a una conclusión, que se cuenta con los recursos requeridos para llevar a cabo el desarrollo de la aplicación móvil.

2.6.1.2. Factibilidad económica

Teniendo en cuenta que los recursos económicos es un factor determinante a la hora de desarrollar cualquier proyecto, en base a eso se ha detallado cada uno de los costos de los recursos que se utilizan, de tal manera obtener un total, para determinar la factibilidad en cuanto a los recursos económicos del proyecto, para lo cual, en la **Tabla 3-2**, se detalla los costos de los recursos utilizados.

Tabla 3-2: Costo de desarrollo

Recurso	Cantidad	Tipo de recurso	Fuente financiamiento	Precio unitario	Precio total
Laptop	1	Hardware	Personal	1130	1130
Dispositivo móvil	1	Hardware	Personal	140	140
Android Studio	1	Software	Personal	0	0
Android Emulator	1	Software	Personal	0	0
				Total	1270

Realizado por: Guailas, Pedro, 2019

Según la **Tabla 3-2**, se detalla cada uno de los recursos utilizados para el desarrollo de la aplicación móvil, con sus respectivos costos, dando una suma total del costo \$1270.00, donde este costo es asumido por el desarrollador, y de esa manera es viable en cuanto a los recursos económicos.

2.6.1.3. Factibilidad operativa

El análisis de la factibilidad operativa, se presta atención en lo específico a los usuarios, quienes utilizan la aplicación móvil, como también se ha tomado en cuenta a los clientes, quienes especifican los requerimientos que necesitan para dicha aplicación. Teniendo en cuenta lo mencionado, también es necesario que tenga alguna manipulación previa con el dispositivo móvil, con el cual tiene noción de poder utilizar de una manera intuitiva. En la **Tabla 4-2**, se detalla el personal que utiliza la aplicación.

Tabla 4-2: Recursos humanos

RRHH	Descripción	Rol
Docente	Profesor de la escuela	Persona encargada de ayudar a manipular la aplicación móvil, a los niños.
Alumno	Niño de la escuela	Usuario final que utiliza la aplicación móvil.

Realizado por: Guailas, Pedro, 2019

Teniendo en cuenta el detalle de la **Tabla 4-2**, los usuarios de la aplicación móvil son niños, mientras que los docentes de la Unidad Educativa son usuarios latentes de la aplicación móvil.

2.6.2. Gestión de Riesgos

La gestión de riesgos es necesario tener presente en cada uno de los proyectos, ya que al surgir algún inconveniente durante el desarrollo se podría afrontar ese percance, por ende, se identifica los riesgos, analiza y prioriza, para de esa manera gestionar cada uno de los riesgos que se presentan en este proyecto.

2.6.2.1. Identificación de riesgos

Cada uno de los riesgos que puede generar durante el desarrollo del proyecto, se encuentra en la **Tabla 5-2**.

Tabla 5-2: Identificación de riesgos

Id	Descripción	Tipo	Consecuencia
R_01	Cambio constante de los requerimientos	Riesgo del proyecto	Replanificación y mayor tiempo
R_02	Falta de conocimiento en el uso de la tecnología para el desarrollo	Riesgo técnico	Mayor tiempo para asesoramiento en la etapa de desarrollo.
R_03	Daños en el Hardware utilizado para el desarrollo	Riesgos del proyecto	Incremento en el costo del proyecto
R_04	Mala planificación	Riesgo del proyecto	Utilizar tiempos fuera del rango establecido
R_05	Mal diseño de la interfaz de usuario	Riesgo técnico	El usuario abandona el uso de la aplicación

Realizado por: Guayllas, Pedro, 2019

Como se visualiza en la **Tabla 5-2**, se ha identificado 5 posibles riesgos, que se puede generar durante el desarrollo del proyecto, de las cuales 4 tipos de riesgos corresponde al riesgo del proyecto, y 1 al riesgo técnico.

2.6.2.2. Análisis de riesgos

Para un correcto análisis de los riesgos, es importante tomar en cuenta la probabilidad y el impacto que puede generar los riesgos y su resultado establece la exposición que tiene esos riesgos. Para lo cual se ha tomado en cuenta los criterios de valoración de la probabilidad que se visualiza en la **Tabla 6-2**.

Tabla 6-2: Criterios de valoración de probabilidad.

Rango de probabilidad	Descripción	Valor
1%-33%	Baja	1
34%-66%	Media	2
67%-100%	Alta	3

Realizado por: Guayllas, Pedro, 2019

Se ha tomado en cuenta los criterios del impacto que genera cada uno de los riesgos, en la **Tabla 7-2**, se establece los criterios de valoración del impacto.

Tabla 7-2: Criterios de valoración del impacto.

Descripción	Valor
Crítico	4
Alto	3
Moderado	2
Bajo	1

Realizado por: Guailas, Pedro, 2019

Se toma en cuenta los criterios de valoración de la exposición al riesgo, que es el resultado de la evaluación entre la probabilidad y el impacto en la **Tabla 8-2**, se visualiza los criterios de valoración de la exposición.

Tabla 8-2: Criterios de valoración de la exposición.

Exposición	Rango = probabilidad * impacto	Valor
Baja	1 – 2	1
Media	3 – 4	2
Alta	Mayor 6	3

Realizado por: Guailas, Pedro, 2019

Teniendo en cuenta cada uno de los criterios de valoración antes mencionados, se ha realizado su respectivo análisis, en la **Tabla 9-2**, se detalla el análisis.

Tabla 9-2: Análisis de riesgos.

Id	Descripción	Probabilidad			Impacto		Exposición	
		%	Probabilidad	Valor	Impacto	Valor	Exposición	Valor
R_01	Cambio constante de los requerimientos	70%	Alta	3	Alto	3	Alta	9
R_02	Falta de conocimiento en el uso de la tecnología para el desarrollo	60%	Media	2	Alto	3	Alta	6
R_03	Daños en Hardware utilizado para el desarrollo	15%	Baja	1	Alto	3	Media	3
R_04	Mala planificación	50%	Media	2	Crítico	4	Alto	8
R_05	Mal diseño de la interfaz de usuario	60%	Media	2	Bajo	1	Bajo	2

Realizado por: Guailas, Pedro, 2019

2.6.2.3. Priorización de los riesgos

Esta etapa de la gestión de riesgos es muy importante, ya que permite conocer cuáles son los riesgos de mayor prioridad a ser gestionados durante el desarrollo del proyecto, para lo cual se detalla en la **Tabla 10-2**, cada uno de los riesgos con los colores rojo, amarillo y verde siendo los rojos de mayor prioridad a ser gestionados.

Tabla 10-2: Priorización de riesgos

N°	Id	Descripción	Exposición	Valor	Prioridad
1	R_01	Cambio constante de los requerimientos	Alta	9	1
2	R_04	Mala planificación	Alta	8	2
3	R_02	Falta de conocimiento en el uso de la tecnología para el desarrollo	Alta	6	3
4	R_03	Daños en Hardware utilizado para el desarrollo	Media	3	4
5	R_05	Mal diseño de la interfaz de usuario	Baja	2	5

Realizado por: Guailas, Pedro, 2019

Una vez establecido la prioridad de los riesgos que se muestra en la **Tabla 10-2**, se determina 3 riesgos de alta, 1 de media y 1 de baja exposición, lo cual se deben tener en cuenta para la gestión de estas que se encuentran en el **Anexo A**, cada una de las hojas de gestión de riesgos.

2.7. Metodología SCRUM para el desarrollo de software

Durante el desarrollo de la aplicación móvil se utiliza la metodología SCRUM, ya que permite un trabajo colaborativo en equipo, permitiendo de esa manera desarrollar un producto mucho más ágil. SCRUM propone roles, artefactos y actividades, siendo una de las principales características que ayuda llevar el control del desarrollo de un proyecto, ya que de esa manera se pretende cubrir todas las necesidades que hace el cliente, desde un principio que lo plantea hasta llegar a concluir con la aceptación entre cada una de las partes que conforman el equipo del proyecto (Monte Galiano, 2016, pp. 15–16).

2.7.1. Análisis de requerimientos

Durante el análisis de los requerimientos, se obtiene los requerimientos funcionales o tareas por hacer, que lo denomina la metodología SCRUM, y requerimientos no funcionales que debe tener la aplicación móvil.

En reunión con los docentes de la Unidad Educativa Intercultural Bilingüe Monseñor Leónidas Proaño, manifiestan que los niños durante la edad de 3 a 5 años están propensos en aprender a identificar figuras geométricas como: línea, rectángulo, círculo, cubo, esfera, cuadrado, pirámide, cilindro, triángulo, rombo y otras figuras, las cuales aprenden identificar con juguetes didácticos geométricos, por otra parte, aprenden reconocer figuras geométricas de acuerdo con sus capacidades cognitivas de cada niño, sin tener definido figuras geométricas específicas. Teniendo presente cada una de las especificaciones se llega a un acuerdo con los docentes de la institución generar las siguientes figuras geométricas línea, triángulo, cuadrado, rectángulo, círculo, cubo esfera, polígono y cilindro, que se visualiza en la aplicación móvil, además la aplicación permite reconocer la voz para generar dicha figura, mencionado toda las especificaciones y los acuerdos, se especifica los requerimientos en las tareas por hacer.

2.7.1.1. Tareas por hacer

Las tareas que se realizan durante el desarrollo de la aplicación se obtienen de las propuestas que plantean los docentes de la escuela, los mismo que se observan en la **Tabla 11-2**, de esa manera, se obtiene los requerimientos funcionales de la aplicación móvil.

Tabla 11-2: Tareas por realizar

N°	Tareas
1	Como desarrollador, diseñar la interfaz de la aplicación
2	Como desarrollador, definir el estándar de codificación
3	Como desarrollador, definir la arquitectura de software para la aplicación
4	Como usuario, ver línea
5	Como usuario, ver triángulo
6	Como usuario, ver cuadrado
7	Como usuario, ver círculo
8	Como usuario, ver cubo
9	Como usuario, ver esfera
10	Como usuario, ver polígono
11	Como usuario, ver cilindro
12	Como usuario, mover figuras 2D
13	Como usuario, mover figuras 3D
14	Como usuario, identificar patrón de voz
15	Documentación
16	Registro de culminación del proyecto

Realizado por: Guailas, Pedro, 2019

2.7.1.2. Requerimientos no funcionales

Usabilidad. La aplicación móvil debe tener una interfaz que sea amigable para el usuario.

2.7.1.3. Estimación

Para el registro de la estimación, se realiza con la técnica de T-Shirt que son las tallas de la camiseta siendo una de las técnicas muy usadas para estimar con SCRUM, el cual toma valores de: XS, S, M, L, XL, que se visualiza en la **Tabla 12-2**, donde cada uno de estos valores representa la talla de la camiseta, pero para la estimación representa el tiempo que toma realizar dicha tarea.

Tabla 12-2: Método de estimación T-Shirt.

Talla	Puntos estimados	Horas de trabajo
XS	5	5
S	10	10
M	20	20
L	40	40
XL	80	80

Realizado por: Guailas, Pedro, 2019

De acuerdo con lo establecido en la **Tabla 12-2**, también se establece que un día tiene 8 horas de trabajo por una persona, encargada del desarrollo de la aplicación.

2.7.1.4. Planificación

Para la planificación del proyecto se ha tenido en cuenta las estimaciones definidas según la metodología SCRUM, además se calendariza de acuerdo con la prioridad que tiene cada historia de usuario o historia técnica, en la **Tabla 13-2**, se visualiza la planificación que se divide por Sprints de acuerdo con la metodología SCRUM.

Tabla 13-2: Panificación

Tareas	Fecha inicio	Fecha fin	Puntos estimados	Puntos totales
Sprint 1				
HT01 Diseño de la interfaz	5-11-2018	7-11-2018	20	20
HT02 Definición de estándar de codificación	7-11-2018	9-11-2018	20	20
Sprint 2				
HT03 Definición de la arquitectura de software	12-11-2018	14-11-2018	20	20
HU01 Generar línea	14-11-2018	16-11-2018	20	20
Sprint 3				
HU02 Generar triángulo	19-11-2018	21-11-2018	20	20
HU03 Generar cuadrado	21-11-2018	23-11-2018	20	20
Sprint 4				
HU04 Generar círculo	26-11-2018	28-11-2018	20	20
HU05 Generar cubo	28-11-2018	30-11-2018	20	20
Sprint 5				
HU06 Generar esfera	3-12-2018	7-12-2018	40	40
Sprint 6				
HU07 Generar polígono	10-12-2018	12-12-2018	20	20
HU08 Generar cilindro	12-12-2018	14-12-2018	20	20
Sprint 7				
HU09 Mover figuras 2D	17-12-2018	21-12-2018	40	40
Sprint 8				
HU10 Mover figuras 3D	7-01-2019	11-01-2019	40	40
Sprint 9				
HU11 Identificar patrón de voz	14-01-2019	18-01-2019	40	40
Sprint 10				
HT04 Documentación	21-01-2019	23-01-2019	40	40
Total			400	400

Realizado por: Guailas, Pedro, 2019

La planificación del proyecto se divide en 10 Sprints, cada Sprint contiene 40 puntos estimados, dando un total de 400 puntos para el desarrollo del proyecto, donde constan todas las funcionalidades requeridas para la aplicación.

2.7.2. Diseño

En la etapa del diseño de la aplicación móvil se realiza, el diseño de la interfaz de usuario, la definición del estándar de codificación, y la definición de la arquitectura de software que son requeridos antes de iniciar la etapa de desarrollo.

2.7.2.1. Diseño de la interfaz de usuario

Para el diseño de la interfaz se ha tomado en cuenta, las recomendaciones que realizan los maestros de la escuela, teniendo en cuenta todas las peticiones donde se destaca mayormente la simplicidad, y los colores claros siendo recomendados para los niños, se ha llegado a obtener la interfaz de la **Figura 2-2**, con el triángulo una de las figuras que se visualiza en la pantalla del dispositivo.

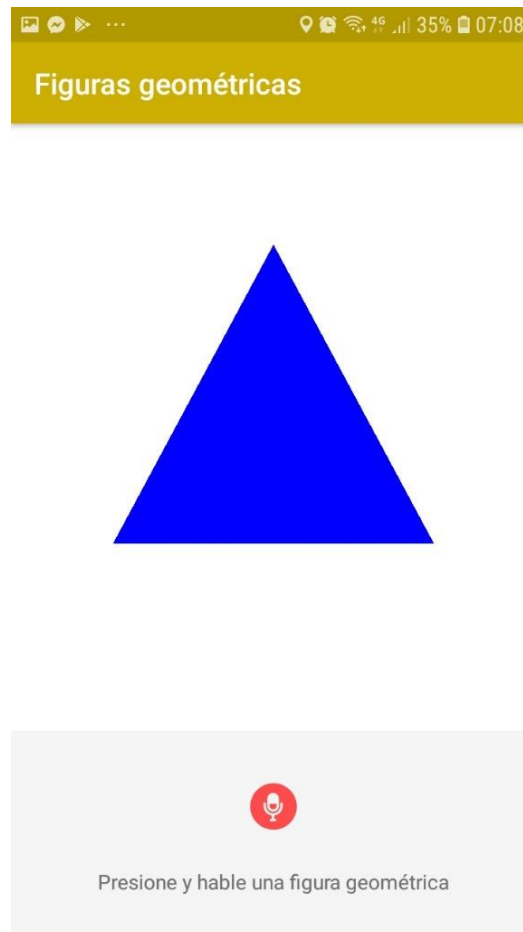


Figura 2-2: Diseño de la interfaz de usuario.
Fuente: Pedro Guaiillas, 2019

De acuerdo con la **Figura 2-2**, se visualiza el diseño de la interfaz de usuario que consta de una cabecera que tiene el texto Figuras geométricas, en la parte inferior se muestra un texto Presione y hable una figura geométrica, y en el cuerpo se visualiza la figura geométrica generada.

2.7.2.2. Definición de estándar de codificación

La definición del estándar de codificación trata de definir una manera de escribir código, y estructurar los directorios y paquetes de una manera definida, para el cual se utiliza el estándar UperCamelCase de CamelCase, uno de los estándares de codificación muy utilizados para la

codificación de las aplicaciones, en el **Anexo B**, se describe cada una de las estructuras necesarias para el desarrollo de la aplicación móvil.

2.7.2.3. Definición de la arquitectura de software

Con respecto a la arquitectura de la aplicación móvil, se escoge el patrón de diseño Modelo Vista Controlador MVC, donde la vista se encarga de visualizar los datos en la pantalla, el controlador está monitorizando de algún suceso que genere al interactuar el usuario con la aplicación, para posterior consumir datos del modelo o visualizar una nueva vista, y el modelo se encarga de estructurar los datos que se utilizan en la aplicación móvil. La definición de la arquitectura del software se visualiza en el diagrama de componente de la **Figura 3-2**.

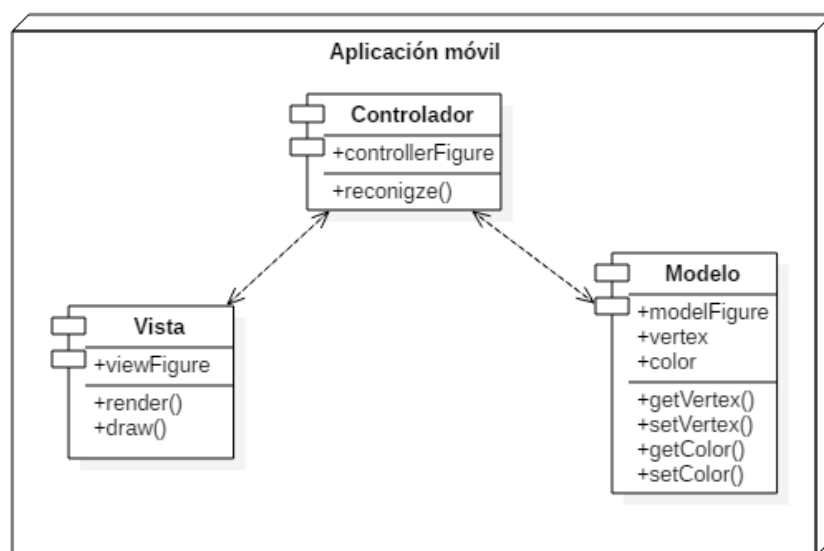


Figura 3-2: Diagrama de componentes de la aplicación.

Fuente. Pedro Guailas, 2019

Según la **Figura 3-2**, se muestra el diagrama de componentes, donde se identifica todos los componentes en que se divide la aplicación móvil, y su relación entre cada uno de ellos, el cual se utiliza para el desarrollo.

2.7.3. Desarrollo

2.7.3.1. Personas y roles del proyecto

La conformación del equipo de trabajo está basada en los roles que se especifica en la **Tabla 14-2**, donde la tabla consta de las siguientes columnas: persona, contacto y el rol que desempeña según la metodología.

Tabla 14-2: Personas y roles

Persona	Contacto	Rol
Msc. Pedro Balla		Product Owner
Dr. Alonzo Álvarez	aalvarez@epoch.edu.ec	Scrum Master
Pedro Guailas	pedro.guailas@epoch.edu.ec	Desarrollador

Realizado por: Guailas, Pedro, 2019

2.7.3.2. Tipos y roles de usuario

El usuario de la aplicación móvil simplemente se ha determinado un tipo de usuario que en la **Tabla 15-2**, se especifica.

Tabla 15-2: Tipo y usuario de la aplicación móvil.

Tipo de usuario	Rol
Niño	El niño es directamente el usuario de la aplicación móvil, quien usa la aplicación cuando lo necesite identificar figuras geométricas.

Realizado por: Guailas, Pedro, 2019

2.7.3.3. Product Backlog

Los requerimientos funcionales o Historias de usuario (HU), se han registrado en reuniones con los maestros de la Escuela, que son las funcionalidades que va a tener la aplicación, por otra parte, se registra las Historias técnicas (HT) que son historias, y deben ser realizados en el proyecto de una manera obligatoria para compensar las historias de usuario. Además, se registra la prioridad donde los maestros creen ser necesarios todos, por ende, todas las historias tienen prioridad Alta. En la **Tabla 16-2**, se visualiza de una manera detallada el Product Backlog.

Tabla 16-2: Product Backlog

ID	Historias de técnicas/usuarios	Prioridad	Estimación
HT01	Diseño de la interfaz	Alta	20
HT02	Definición de estándar de codificación	Alta	20
HT03	Definición de la arquitectura de software	Alta	20
HU01	Visualizar línea	Alta	20
HU02	Visualizar triángulo	Alta	20
HU03	Visualizar cuadrado	Alta	20
HU04	Visualizar círculo	Alta	20
HU05	Visualizar cubo	Alta	20
HU06	Visualizar esfera	Alta	40
HU07	Visualizar polígono	Alta	20
HU08	Visualizar cilindro	Alta	20
HU09	Mover figuras 2D	Alta	40
HU10	Mover figuras 3D	Alta	40
HU11	Identificar patrón de voz	Alta	40
HT04	Documentación	Media	40
Total, puntos estimados			400

Realizado por: Guailas, Pedro, 2019

El product backlog contiene un total de 11 historias de usuario, y 5 historias técnicas, dando un total de 400 puntos estimados.

2.7.3.4. Historias de usuarios

Según (Scrum Manager, 2016, p. 77) la estructura de una historia de usuario es la siguiente:

- **ID:** Identificador único de la historia de usuario.
- **Nombre:** nombre de la historia de usuario a desarrollar.
- **Usuario:** persona que va a manipular el requerimiento.
- **Sprint:** número del Sprint en el que se va a desarrollar el requerimiento.
- **Prioridad del negocio:** permite determinar el orden en el que se va a desarrollar el requerimiento.
- **Riesgo en el desarrollo:** permite determinar el peligro que puede ocasionar al proyecto el no cumplir correctamente el desarrollo del requerimiento.
- **Puntos de Estimación:** tiempo estimado que se necesita para realizar el requerimiento.
- **Puntos Reales:** tiempo real que se necesita para realizar el requerimiento
- **Descripción:** información que sirve para explicar el desarrollo del requerimiento.
- **Criterios de validación:** son pruebas que se realizan para validar el funcionamiento correcto de la historia de usuario.
- **Observaciones:** esclarecer información.

En la **Tabla 17-2** se puede visualizar un modelo de las historias de usuario que se utilizan a la hora de aplicar el marco de trabajo SCRUM las cuales fueron adaptadas por Scrum Manager.

Tabla 17-2: Modelo de la historia de usuario

HISTORIA DE USUARIO	
Número: HU01	Nombre de la historia: Visualizar línea
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar una línea, para que los niños reconozcan e identifiquen una línea.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que la línea se grafique correctamente en la pantalla del dispositivo móvil. 	

Fuente: Adaptado de (Scrum Manager, 2016, p. 77).

Realizado por: Guayllas, Pedro, 2019

2.7.3.5. Sprint Backlog

En cada Sprint, se desarrollan una o varias historias de usuario que son las funcionalidades que tiene la aplicación, o historias técnicas que son historias, y deben ser realizados en el proyecto de una manera obligatoria, para compensar las historias de usuario. Las historias se desarrollan según la planificación establecida con una fecha de inicio y fin, teniendo presente el número de horas estimadas para cada tarea, como también el responsable o encargado del desarrollo de esa tarea. En la **Tabla 18-2**, se registrar los Sprint Backlog, donde hace constancia lo antes detallado.

Tabla 18-2: Sprint Backlog

Tareas	Horas	Fecha inicio	Fecha fin	Responsable
Sprint 1				
HT01 Diseño de la interfaz	20	5-11-2018	7-11-2018	Pedro Guaillas
HT02 Definición de estándar de codificación	20	7-11-2018	9-11-2018	Pedro Guaillas
Sprint 2				
HT03 Definición de la arquitectura de software	20	12-11-2018	14-11-2018	Pedro Guaillas
HU01 Visualizar línea	20	14-11-2018	16-11-2018	Pedro Guaillas
Sprint 3				
HU02 Visualizar triángulo	20	19-11-2018	21-11-2018	Pedro Guaillas
HU03 Visualizar cuadrado	20	21-11-2018	23-11-2018	Pedro Guaillas
Sprint 4				
HU04 Visualizar círculo	20	26-11-2018	28-11-2018	Pedro Guaillas
HU05 Visualizar cubo	20	28-11-2018	30-11-2018	Pedro Guaillas
Sprint 5				
HU06 Visualizar esfera	40	3-12-2018	7-12-2018	Pedro Guaillas
Sprint 6				
HU07 Visualizar polígono	20	10-12-2018	12-12-2018	Pedro Guaillas
HU08 Visualizar cilindro	20	12-12-2018	14-12-2018	Pedro Guaillas
Sprint 7				
HU09 Mover figuras 2D	40	17-12-2018	21-12-2018	Pedro Guaillas
Sprint 8				
HU10 Mover figuras 3D	40	7-01-2019	11-01-2019	Pedro Guaillas
Sprint 9				
HU11 Identificar patrón de voz	40	14-01-2019	18-01-2019	Pedro Guaillas
Sprint 10				
HT04 Documentación	40	21-01-2019	23-01-2019	Pedro Guaillas

Realizado por: Guaillas, Pedro, 2019

2.7.3.6. Sprints

De acuerdo con la metodología SCRUM, recomienda que cada Sprint tenga una duración de entre 1 a 3 semanas de duración, por ende, se ha optado por establecer rangos de tiempo en promedio de una semana, y en cada Sprint desarrollar 2 historias, también recomienda documentar las historias tanto técnicas como de usuario, con sus respectivas pruebas de aceptación, para más información revisar el **Anexo C**.

Sprint 1: Durante el primer Sprint se desarrolla historias correspondientes al diseño que son historias técnicas necesarias para compensar las historias de usuario, las cuales comprenden el

diseño de la interfaz y la definición del estándar de codificación, la cual se detalla en la **Tabla 19-2**, del Sprint 1.

Tabla 19-2: Sprint 1

Sprint 1				
Inicio: 5-11-2018		Fin: 9-11-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HT01	Diseño de la interfaz	20	Desarrollo	Pedro Guaitas
HT02	Definición de estándar de codificación	20	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Sprint 2: Se desarrolla una historia técnica y una historia de usuario, las cuales comprende, la definición de la arquitectura de software, y se empieza a la visualización de la línea que es la primera figura definida en 2D, de esa manera se resumen en la **Tabla 20-2**, del Sprint 2.

Tabla 20-2: Sprint 2

Sprint 2				
Inicio: 12-11-2018		Fin: 16-11-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HT03	Definición de la arquitectura de software	20	Desarrollo	Pedro Guaitas
HU01	Visualizar línea	20	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Al finalizar la funcionalidad del Sprint 2 perteneciente a la figura geométrica de la línea, se obtiene la visualización de figura en la pantalla del dispositivo móvil, la cual se puede visualizar en la **Figura 4-2**.

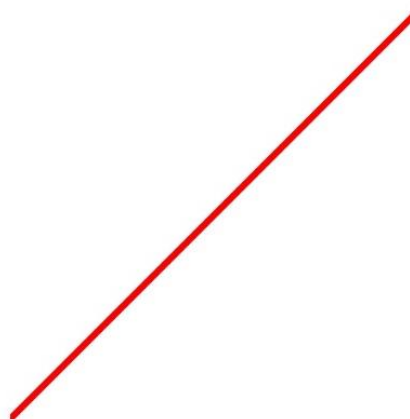


Figura 4-2: Figura geométrica Línea
Realizado por: Guaitas, Pedro, 2019

Sprint 3: En este Sprint se desarrollan dos historias de usuario, las cuales comprende la visualización del triángulo y la visualización del cuadrado, que son figuras en 2D, de esa manera se resumen en la **Tabla 21-2**, del Sprint 3.

Tabla 21-2: Sprint 3

Sprint 3				
Inicio: 19-11-2018		Fin: 23-11-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU02	Visualizar triángulo	20	Desarrollo	Pedro Guailas
HU03	Visualizar cuadrado	20	Desarrollo	Pedro Guailas

Realizado por: Guailas, Pedro, 2019

Al finalizar las funcionalidades del Sprint 3 perteneciente a las figuras geométricas del triángulo y cuadrado, se obtiene la visualización de las figuras en la pantalla del dispositivo móvil, las cuales se pueden visualizar en la **Figura 5-2** y **Figura 6-2**.

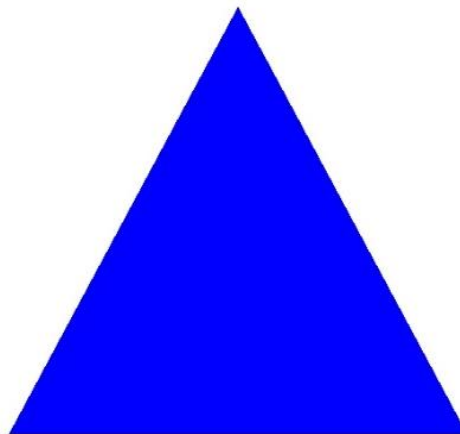


Figura 5-2: Figura geometría Triángulo
Realizado por: Guailas, Pedro, 2019

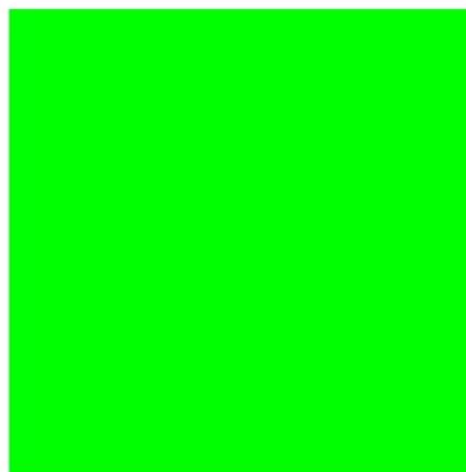


Figura 6-2: Figura geometría Cuadrado
Realizado por: Guailas, Pedro, 2019

Sprint 4: En este Sprint se desarrollan dos historias de usuario, las cuales comprende la visualización del círculo y la visualización del cubo la primera figura en 3D, de esa manera se resumen en la **Tabla 22-2**, del Sprint 4.

Tabla 22-2: Sprint 4

Sprint 4				
Inicio: 26-11-2018		Fin: 30-11-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU04	Visualizar círculo	20	Desarrollo	Pedro Guailas
HU05	Visualizar cubo	20	Desarrollo	Pedro Guailas

Realizado por: Guailas, Pedro, 2019

Al finalizar las funcionalidades del Sprint 4 perteneciente a las figuras geométricas del círculo y cubo, se obtiene la visualización de las figuras en la pantalla del dispositivo móvil, las cuales se pueden visualizar en la **Figura 7-2** y **Figura 8-2**.

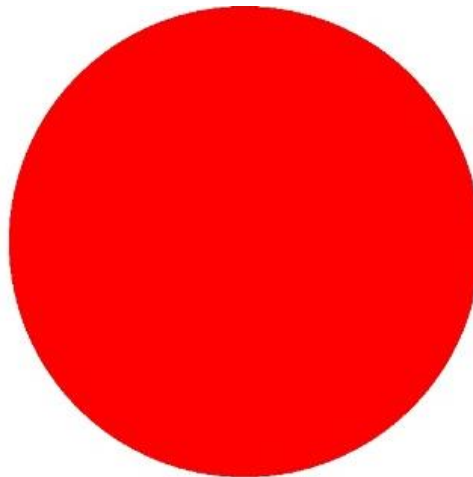


Figura 7-2: Figura geométrica Círculo
Realizado por: Guailas, Pedro, 2019

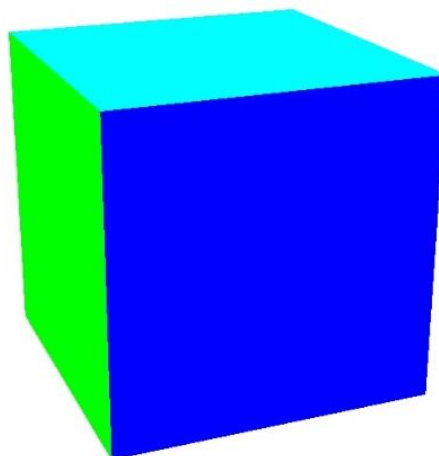


Figura 8-2: Figura geométrica Cubo
Realizado por: Guailas, Pedro, 2019

Sprint 5: En este Sprint se desarrollan una historia de usuario, correspondiente a la visualización de la esfera que es una figura en 3D, de esa manera se resumen en la **Tabla 23-2**, del Sprint 5.

Tabla 23-2: Sprint 5

Sprint 5				
Inicio: 03-12-2018		Fin: 07-12-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU06	Visualizar esfera	40	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Al finalizar la funcionalidad del Sprint 5 perteneciente a la figura geométrica de la esfera, se obtiene la visualización de la figura en la pantalla del dispositivo móvil, la cual se puede visualizar en la **Figura 9-2**.

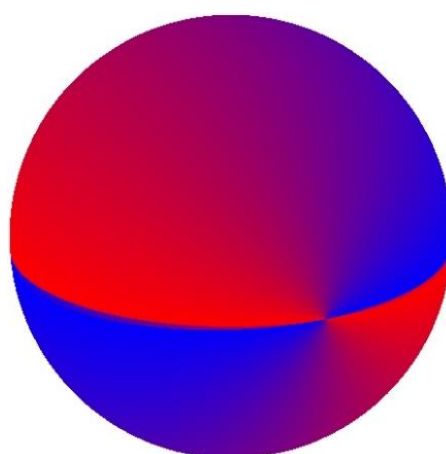


Figura 9-2: Figura geométrica Esfera
Realizado por: Guaitas, Pedro, 2019

Sprint 6: En este Sprint se desarrollan dos historias de usuario, correspondiente a las figuras en 3D, como son: visualización del polígono y la visualización del cilindro, de esa manera se resumen en la **Tabla 24-2**, del Sprint 6.

Tabla 24-2: Sprint 6

Sprint 6				
Inicio: 10-12-2018		Fin: 14-12-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU07	Visualizar polígono	20	Desarrollo	Pedro Guaitas
HU08	Visualizar cilindro	20	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Al finalizar las funcionalidades del Sprint 6 perteneciente a las figuras geométricas del polígono y el cilindro, se obtiene la visualización de las figuras en la pantalla del dispositivo móvil, las cuales se pueden visualizar en la **Figura 10-2** y **Figura 11-2**.

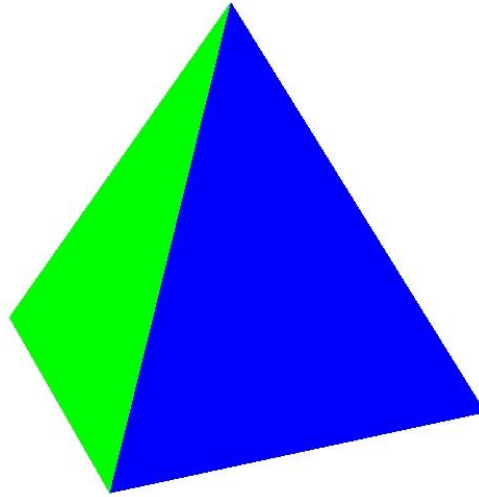


Figura 10-2: Figura geométrica Pirámide
Realizado por: Guailas, Pedro, 2019

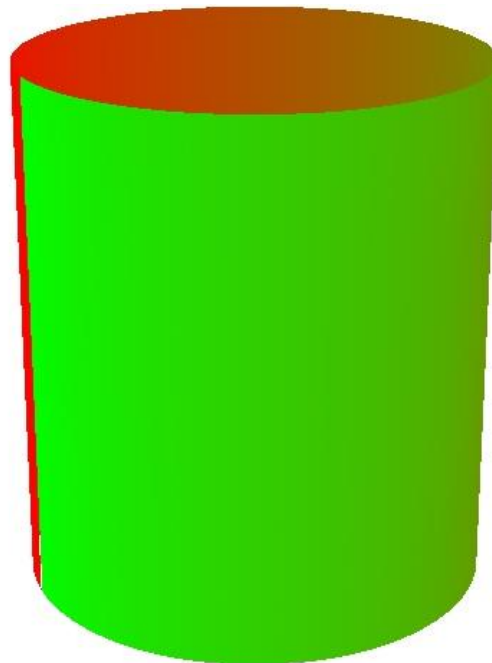


Figura 11-2: Figura geométrica Cilindro
Realizado por: Guailas, Pedro, 2019

Sprint 7: En este Sprint se desarrolla una historia de usuario, la cual permite mover figuras en 2D que corresponde a la línea, cuadrado, triángulo y círculo, de esa manera se especifica en la **Tabla 25-2**, del Sprint 7.

Tabla 25-2: Sprint 7

Sprint 7				
Inicio: 17-13-2018		Fin: 21-12-2018		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU09	Mover figuras en 2D	40	Desarrollo	Pedro Guailas

Realizado por: Guailas, Pedro, 2019

Sprint 8: En este Sprint se desarrolla una historia de usuario, la cual permite mover figuras en 3D que corresponde al polígono, cuadrilátero, esfera y cilindro, de esa manera se detalla en la **Tabla 26-2**, del Sprint 8.

Tabla 26-2: Sprint 8

Sprint 8				
Inicio: 07-01-2019		Fin: 11-01-2019		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU10	Mover figuras en 3D	40	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Sprint 9: En este Sprint se desarrolla una historia de usuario, que es identificar patrones de voz para interactuar la aplicación móvil mediante la voz, de esa manera se detalla en la **Tabla 27-2**, del Sprint 9.

Tabla 27-2: Sprint 9

Sprint 9				
Inicio: 14-01-2019		Fin: 18-01-2019		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HU11	Identificar patrones de voz	40	Desarrollo	Pedro Guaitas

Realizado por: Guaitas, Pedro, 2019

Al finalizar la funcionalidad del Sprint 9, perteneciente a agregar identificadores de patrones de voz, para cada una de las figuras descritas en los anteriores Sprint, se obtiene la visualización de cada uno de las figuras geométricas en la pantalla del dispositivo móvil, mediante la pronunciación del patrones de voz, con el nombre correspondiente a cada figura implementada en el proyecto, el cual se puede visualizar en la **Figura 12-2**, el botón que se presiona antes de pronunciar una figura geométrica.

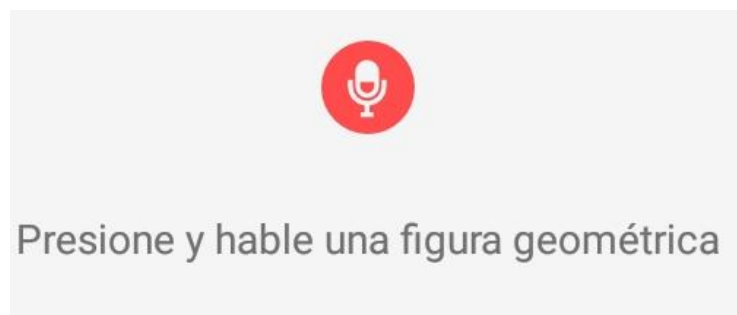


Figura 12-2: Botón para identificar patrones de voz

Realizado por: Guaitas, Pedro, 2019

Sprint 10: En este Sprint se elabora la descripción tanto del manual técnico como de usuario, para tener constancia de las actividades realizadas durante el desarrollo de la aplicación móvil,

además permite conocer a los clientes como interactuar con la aplicación móvil durante su uso, de esa manera se detalla en la **Tabla 28-2**, del Sprint 10.

Tabla 28-2: Sprint 10

Sprint 10				
Inicio: 21-01-2019		Fin: 24-01-2019		Horas: 40
Pila de Sprint				
ID	Descripción	Horas	Tipo	Responsable
HT04	Documentación	40	Desarrollo	Pedro Guaiillas

Realizado por: Guaiillas, Pedro, 2019

2.7.3.7. Reuniones y entregables

Acorde a la metodología ágil Scrum, se lleva a cabo varias reuniones con los docentes de la Unidad Educativa Intercultural Bilingüe Monseñor Leónidas Proaño, con el objetivo de revisar conjuntamente las funcionalidades implementadas en cada Sprint del aplicativo móvil, que se realiza de cada uno de los requerimientos planteados por los docentes en el desarrollo del proyecto. Además los entregables se presentaron cada semana, donde cada Sprint consta de 1 a 2 requerimientos.

En la **Tabla 29-2** se muestra las reuniones que se registran entre el desarrollador y los docentes, para realizar la entrega de los avances del aplicativo móvil de cada Sprint.

Tabla 29-2: Reuniones y entregables

Descripción	Fecha de entrega	Responsables
Reunión para la recolección de requerimientos para el aplicativo móvil.	10/10/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para definir el estándar de interfaz	20/10/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 1	09/11/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 2	16/11/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 3	23/11/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 4	30/11/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 5	07/12/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 6	14/12/2018	Pedro Guaiillas (Desarrollador) Fausto Yépez (Cliente)

Continúa

Reunión para presentar el avance del Sprint 7	21/12/2018	Pedro Guaitas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 8	11/01/2019	Pedro Guaitas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 9	18/01/2019	Pedro Guaitas (Desarrollador) Fausto Yépez (Cliente)
Reunión para presentar el avance del Sprint 10	23/01/2019	Pedro Guaitas (Desarrollador) Fausto Yépez (Cliente)

Realizado por: Guaitas, Pedro, 2019

2.7.3.8. Fase de finalización

En la **Tabla 30-2** se describe las actividades que llevan a la conclusión del aplicativo móvil.

Tabla 30-2: Actividades para la conclusión del aplicativo.

Actividad	Descripción	Responsable
Realizar la documentación del trabajo de titulación	Elaboración del Manuel Técnico de la aplicación móvil.	Desarrollador
Capacitación de los usuarios	Entrega del manual técnico a los docentes.	Desarrollador.

Realizado por: Guaitas, Pedro, 2019

2.7.3.9. Gestión del proyecto

Al finalizar la fase de desarrollo de cada uno de los Sprints planificados para el desarrollo del proyecto las cuales se pueden visualizar en el **Grafico 1-2**, que lleva como nombre Burndown Chart, a través de esta herramienta que proporciona la metodología ágil Scrum, se puede visualizar el avance del aplicativo móvil, además se puede observar si se está cumpliendo con los tiempos determinados en la planificación.

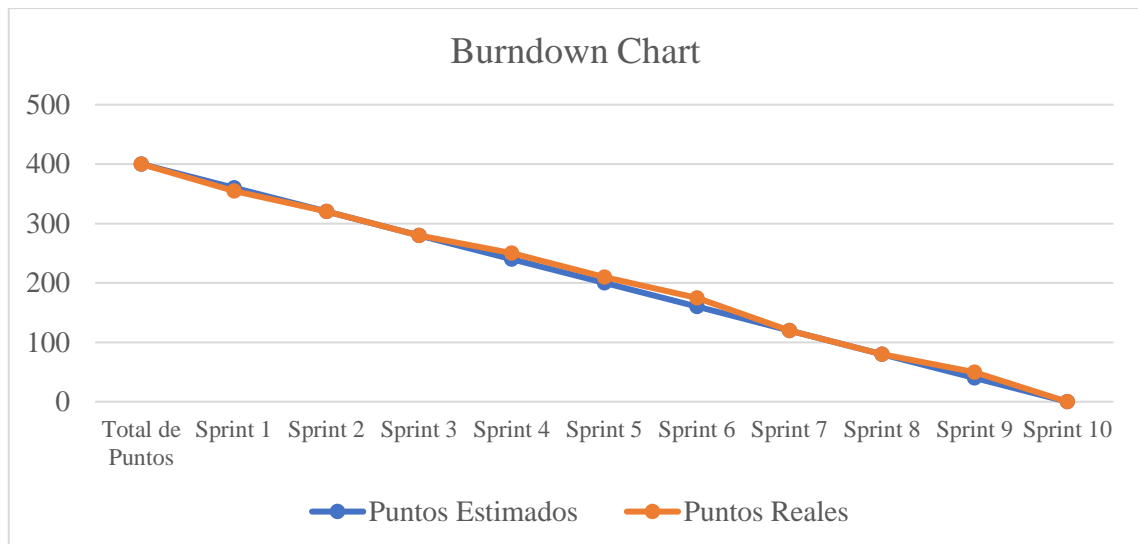


Gráfico 1-2: Burndown Chart

Realizado por: Guayllas, Pedro, 2019

Como resultado del desarrollo del aplicativo, se realizaron 440 puntos reales en un total de 10 Sprints de una semana de duración cada uno, obteniendo como diferencia 40 puntos entre lo planificado y lo real. En el primer Sprint la línea real se encuentra por debajo de la línea ideal, lo que significa que el estándar de codificación fue adaptado a una norma ya establecida, y esto redujo el tiempo. Además, en los Sprint 4, 5, 6 y 9 la línea de trabajo real está por encima de la línea de trabajo ideal, lo que significa que queda más trabajo del previsto originalmente, y el proyecto se retrasa por falta de conocimientos en el uso de la tecnología de desarrollo.

2.7.4. Pruebas

Durante esta etapa se verifica todas las pruebas de aceptación, que se plantean en las historias de usuario, permitiendo conocer el correcto desarrollo de cada uno de los Sprints, y de esa manera tener validez técnica del desarrollo de la aplicación móvil. Todas las pruebas de aceptación se encuentran en el **Anexo C**.

2.7.5. Implantación

La implementación de la aplicación móvil se lleva a cabo en los dispositivos de los docentes y algunos padres de familia, para lo cual se tuvo en cuenta especificaciones mínimas que debe tener el dispositivo móvil como, que tenga una versión de Android 4.0, cumpliendo con este requerimiento, la aplicación móvil puede ejecutar sin interrupciones, y manipular cuando el niño lo requiera.

CAPÍTULO III

3. MARCO DE RESULTADOS Y DISCUSIÓN DE LOS RESULTADOS

En el presente capítulo se analiza la validación del “**DESARROLLO DE UNA APLICACIÓN MÓVIL QUE GENERE Y MUEVA FIGURAS GEOMÉTRICAS, MEDIANTE LA IDENTIFICACIÓN DE PATRONES DE VOZ**”, teniendo en cuenta los principios de la calidad de software, y la libertad que permite definir de acuerdo a las necesidades, se ha optado por evaluar la usabilidad de la aplicación móvil, debido a que se desarrolla una aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz, el cual al pronunciar una figura la aplicación debe realizar dicha figura, el cual influye en la precisión para no confundir entre una figura y otra además, la comodidad de los niños al interactuar con un dispositivo móvil y aceptar que puede realizar las tareas en el dispositivo, han influido para relacionar con las métricas del atributo de calidad, donde la precisión se puede relacionar con la eficacia, y la aceptación con la satisfacción, una vez determinada el atributo de calidad de software también se ha definido el estándar ISO 9241-11, que es un estándar extendido de la ISO 9241 que contiene las mismas métricas de eficacia, eficiencia y satisfacción, pero aplicado a aplicaciones móviles.

Se ha excluido los demás atributos de la calidad de software seguridad, fiabilidad, funcionalidad, portabilidad, mantenibilidad, eficiencia ya que no cumplen la mayoría de las sub-características o métricas haciendo que esto sea complejo durante el proceso de evaluación, o tener que realizar pruebas complejas.

3.1. Población y muestra

3.1.1. Población

Para el desarrollo de la investigación, se lo realizó en la sede número 2 de la Unidad Educativa Intercultural Bilingüe Monseñor Leónidas Proaño, donde se encuentran matriculados 114 niños del nivel inicial, en el periodo septiembre 2018 – julio 2019.

3.1.2. Muestra

Se ha determinado una muestra probabilística utilizando la ecuación del cálculo del tamaño de la muestra conociendo el tamaño de la población que a continuación se plantea.

$$n = \frac{N * Z^2 * p * q}{d^2 * (N - 1) + Z^2 * p * q}$$

En donde:

- N=Es la población finita
- Z=Nivel de confianza
- p=Probabilidad de éxito, o proporción esperada
- q=Probabilidad de fracaso
- d=Precisión (error máximo admisible)

Para el cálculo se tiene los siguientes valores:

- N=114
- Z=95% equivalente a 1.94
- p=0.5
- q=0.5
- d=10% equivalente a 0.1

Remplazando los valores en la ecuación se obtuvo el siguiente resultado:

$$n = \frac{114 * 1.94^2 * 0.5 * 0.5}{0.1^2 * (114 - 1) + 1.94^2 * 0.5 * 0.5} = 51.79 \equiv 52$$

En base al resultado de la muestra 52 niños, para la encuesta se ha tomado 24 niños de tres paralelos y 28 niños de cuatro paralelos que suman la muestra, a los cuales se les proporciono un dispositivo móvil con la aplicación para su respectivo uso, y su posterior evaluación con las encuestas de los criterios establecidos.

3.2. Parámetros de evaluación

Se llevó acabo el análisis de la usabilidad de la aplicación, debido a que es uno de los parámetros más influyentes dentro de los atributos de la calidad del software. Para este análisis se basa en el estándar ISO 9241-11, que es uno de los estándares con enfoque en la usabilidad incluyendo recomendaciones para evaluar aplicaciones móviles. El estándar se fundamenta en tres aspectos de usabilidad que son: eficiencia, efectividad y satisfacción.

3.2.1. Criterios de evaluación

Existen una variedad de criterios de evaluación tanto cuantitativos como cualitativos que permiten extraer resultados en base a un diseño experimental planteado. Para evaluar este estudio se estableció preguntas cerradas dicotómicas en la encuesta, ya que los niños no son tan críticos para determinar grados de decisión cuando se realiza una encuesta, por ende, se plantearon preguntas dicotómicas de si o no.

3.2.2. Análisis de los parámetros de la aplicación móvil

Para la evaluación de la aplicación móvil se tuvo en cuenta el uso de términos más comunes que permitan entender a los niños, como también se planteó preguntas específicas de las tareas que realiza la aplicación móvil, de esa manera los niños pudieron responder a las preguntas planteadas, ya que los niños a temprana edad no utilizan términos muy sofisticados o términos que se utilizan con frecuencia en otras encuestas.

Después de realizar las encuestas a los niños se llegó a obtener los resultados que se muestra en la **Tabla 1-3**, cada pregunta ha obtenido un valor de si o no como se había planteado en los criterios de evaluación.

Tabla 31-3: Resultados de la encuesta

Parámetro	N°	Pregunta	Resultado	
			Si	No
Eficiencia	1.	¿Te enseñé cómo usar la aplicación?	37	15
	2.	¿Te gustaría ver más figuras?	21	31
	3.	¿Necesitas que agregue más botones a la aplicación?	32	20
Efectividad	4.	¿La aplicación muestra la figura que quieres?	32	20
	5.	¿Tenía que repetir el nombre de la figura más de una vez, para que se muestre?	28	24
	6.	¿Se demoraba la aplicación en mostrar las figuras en la pantalla?	50	2
Satisfacción	7.	¿Te gusta los colores de las figuras?	52	0
	8.	¿Te gusta la aplicación?	50	2
	9.	¿Te gusta cómo se mueve las figuras?	48	4

Realizado por: Guailas, Pedro, 2019

Luego del análisis realizado en la **Tabla 1-3**, se determinó que para la evaluación de la usabilidad dentro del aplicativo móvil se tomaron en cuenta las sub-características eficiencia, eficacia y satisfacción planteadas según la norma ISO 9241-11, donde cada sub-característica contiene 3 items dando un total de 9 preguntas, las cuales fueron aplicadas a los estudiantes, donde sus respuestas se evaluaron bajo el criterio de evaluación de preguntas dicotómicas las cuales son respuestas de si o no.

3.3. Análisis de resultados

Una vez realizada la encuesta a los niños, ya se pudo tabular, analizar, graficar y concluir cada una de las características por separado, para determinar el grado de usabilidad que posee la aplicación móvil.

En base a las preguntas de la problematización, se seleccionó la pregunta, **¿Mejoraría el nivel de aprendizaje en los niños con una aplicación móvil?** Para contrastar de esa manera cuanto mejora la aplicación móvil en el aprendizaje de los niños.

3.3.1. Análisis de eficiencia

En la **Tabla 2-3**, se contabiliza las preguntas concernientes a la eficiencia. En cada pregunta la suma de respuestas afirmativas y negativas dan un total del tamaño de la muestra, que son un total de 52 niños.

Tabla 32-3: Análisis de eficiencia.

Pregunta	Pregunta	Respuestas	
		Si	No
1	¿Te enseñó cómo usar la aplicación?	37	15
2	¿Te gustaría ver más figuras?	21	31
3	¿Necesitas que agregue más botones a la aplicación?	32	20

Realizado por: Guailas, Pedro, 2019

De acuerdo con las respuestas obtenidas en la **Tabla 2-3**, se iguala la suma de las respuestas SI con que la aplicación móvil MEJORA, y la respuesta no se iguala con que la aplicación móvil NO MEJORA, de esa manera contrastar la mejoría que tiene la aplicación móvil.

Tabla 33-3: Resultados agrupados de eficiencia.

Pregunta	MEJORA	NO MEJORA
1	37	15
2	21	31
3	32	20
Total	90	66

Realizado por: Guailas, Pedro, 2019

En la **Tabla 3-3**, se muestra las respuestas agrupadas de acuerdo con lo definido por el parámetro que mejora y no mejora, de esa manera se obtuvo el 57.69% que mejora, y el 42.31% que no mejora el nivel de aprendizaje con la utilización de la aplicación móvil, como se visualiza en el **Gráfico 1-3**.

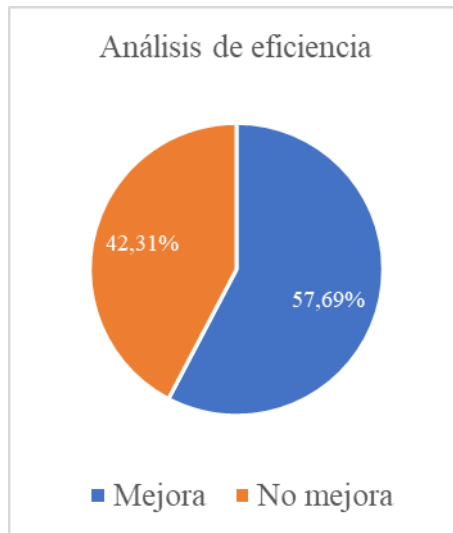


Gráfico 2-3: Análisis de eficiencia.
Realizado por: Guailas Pedro, 2019.

3.3.2. Análisis de eficacia

En la **Tabla 4-3**, se contabiliza las preguntas pertinentes a la eficacia. En cada pregunta la suma de respuestas afirmativas y negativas dan un total del tamaño de la muestra, que son un total de 52 niños.

Tabla 34-3: Análisis de eficacia.

Pregunta	Pregunta	Respuestas	
		Si	No
1	¿La aplicación muestra la figura que quieres?	32	20
2	¿Tenía que repetir el nombre de la figura más de una vez, para que se muestre?	28	24
3	¿Se demoraba la aplicación en mostrar las figuras en la pantalla?	50	2

Realizado por: Guailas, Pedro, 2019

De acuerdo con las respuestas obtenidas en la **Tabla 4-3**, se iguala la suma de las respuestas SI con que la aplicación móvil MEJORA, y la respuesta no se iguala con que la aplicación móvil NO MEJORA de esa manera contrastar la mejoría que tiene la aplicación móvil.

Tabla 35-3: Resultados agrupados de eficacia.

Pregunta	MEJORA	NO MEJORA
1	32	20
2	28	24
3	50	2
Total	110	46

Realizado por: Guailas, Pedro, 2019

En la **Tabla5-3**, se muestra las respuestas agrupadas de acuerdo con lo definido por el parámetro que mejora y no mejora, de esa manera se obtuvo el 70.51% que mejora, y el 29.49% que no mejora el nivel de aprendizaje con la utilización de la aplicación móvil, como se visualiza en el **Gráfico 2-3**.

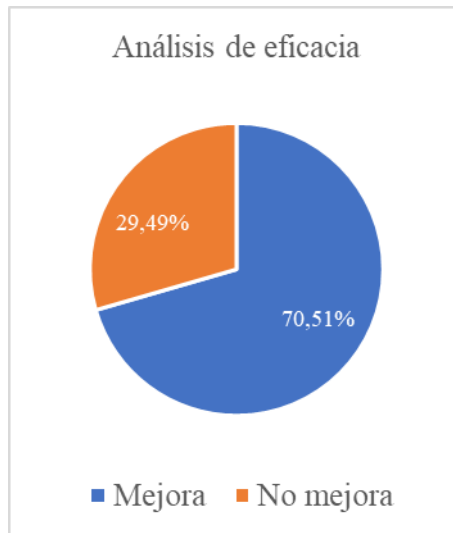


Gráfico 3-3: Análisis de eficacia.
Realizado por: Guailas Pedro, 2019.

3.3.3. Análisis de satisfacción

En la **Tabla 6-3**, se contabiliza las preguntas referentes a la satisfacción. En cada pregunta la suma de respuestas afirmativas y negativas dan un total del tamaño de la muestra, que son un total de 52 niños.

Tabla 36-3: Análisis de satisfacción.

Pregunta	Pregunta	Respuestas	
		Si	No
1	¿Te gusta los colores de las figuras?	52	0
2	¿Te gusta la aplicación?	50	2
3	¿Te gusta cómo se mueve las figuras?	48	4

Realizado por: Guailas, Pedro, 2019

De acuerdo con las respuestas obtenidas en la **Tabla 6-3**, se iguala la suma de las respuestas SI con que la aplicación móvil MEJORA, y la respuesta no se iguala con que la aplicación móvil NO MEJORA, de esa manera contrastar la mejoría que tiene la aplicación móvil.

Tabla 37-3: Resultados agrupados de satisfacción.

Pregunta	MEJORA	NO MEJORA
1	52	0
2	50	2
3	48	4
Total	150	6

Realizado por: Guailas, Pedro, 2019

En la **Tabla 7-3**, se muestra las respuestas agrupadas de acuerdo con lo definido por el parámetro que mejora y no mejora, con respecto al parámetro de satisfacción, de esa manera se obtuvo el 96.15% que mejora, y el 3.85% que no mejora el nivel de aprendizaje con la utilización de la aplicación móvil, como se visualiza en el **Gráfico 3-3**.

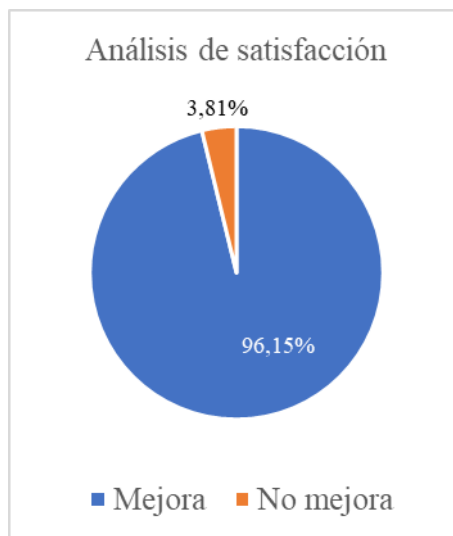


Gráfico 4-3: Análisis de satisfacción.
Realizado por: Guailas Pedro, 2019.

3.4. Prueba de significancia estadística

Para la evaluación del nivel de significancia que tiene este estudio, se tuvo en cuenta la problematización del problema, y de esa manera se plantea la hipótesis a evaluar con un estudio probabilístico del Chi cuadrado.

3.4.1. Determinación de variables

Variable independiente: La aplicación móvil basada en OpenGL ES que genera y mueve figuras geométricas mediante patrones de voz.

Variable dependiente: Mejorará en los niños la identificación de figuras geométricas.

3.4.2. Formulación de la hipótesis

La aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz mejorará en los niños la identificación de las figuras geométricas.

3.4.3. Procedimiento

3.4.3.1. Hipótesis nula

La aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz no mejorará en los niños la identificación de figuras geométricas.

3.4.3.2. Hipótesis de investigación

La aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz mejorará en los niños la identificación de figuras geométricas.

3.4.3.3. Prueba Chi cuadrado

Chi cuadrado establece una diferencia entre un valor esperado y su valor nula, que para nuestro estudio comprende la diferencia entre la hipótesis de investigación y la hipótesis nula, donde se establece el nivel de significancia de un 0.05 hasta un 0.01, para de esa manera rechazar o aceptar la hipótesis planteada. La prueba Chi cuadrado plantea la siguiente ecuación.

$$x^2 = \sum \frac{(\text{observado} - \text{esperado})}{\text{esperado}}$$

3.4.3.4. Frecuencia observada

Luego de analizar cada parámetro de la usabilidad, se obtuvo valores cuantitativos según la tabulación de los datos, los cuales se pueden observar en la **Tabla 8-3**, donde se obtiene un resultado igual al tamaño de la muestra por cada parámetro evaluado, que corresponden a la eficiencia, eficacia y satisfacción.

Tabla 38-3: Frecuencia observada.

Evento	Eficiencia	Eficacia	Satisfacción	Total
No mejora	66	46	6	118
Mejora	90	110	150	350
Total	156	156	156	468

Realizado por: Guailas, Pedro, 2019

3.4.3.5. Frecuencia esperada

Dado que se quiere conocer la relación que existen entre dos valores, se establece valores esperados que consisten a la correspondencia que puede existir entre dos elementos nominativos. La frecuencia esperada es el resultado de la multiplicación del total de cada fila por el valor de la columna total dividido para el total de las frecuencias observadas. La frecuencia esperada plantea la siguiente ecuación.

$$FE = \frac{Tf * Tc}{N}$$

Al remplazar cada valor en la ecuación anterior se obtuvo el siguiente resultado que se muestra a continuación:

$$FE \text{ primera fila} = \frac{156 * 118}{468} = 39$$

$$FE \text{ segunda fila} = \frac{156 * 350}{468} = 117$$

Después de obtener cada uno de los valores de las frecuencias esperadas, por cada fila, se ubicó cada uno de los valores que se esperaba obtener, que se muestra en la **Tabla 9-3**.

Tabla 39-3: Frecuencia esperada.

Evento	Eficiencia	Eficacia	Satisfacción	Total
No mejora	39	39	39	118
Mejora	117	117	117	350
Total	156	156	156	468

Realizado por: Guayllas, Pedro, 2019

3.4.3.6. Grados de libertad

Los grados de libertad es el resultado de la multiplicación entre el número de filas menos uno por el número de columnas menos uno, que a continuación se visualiza la siguiente ecuación y el cálculo al remplazar ya lo valores.

$$Gl = (f - 1) * (c - 1) = (2 - 1) * (3 - 1) = 2$$

3.4.3.7. Cálculo del Chi cuadrado

A continuación, se procedió a realizar cálculos matemáticos para obtener el valor del Chi cuadrado (χ^2), de acuerdo con la siguiente ecuación:

$$\chi^2 = \sum \frac{(\text{observado} - \text{esperado})^2}{\text{esperado}}$$

$$\chi^2 = \sum \frac{(66 - 39)^2}{39} + \frac{(46 - 39)^2}{39} + \frac{(6 - 39)^2}{39} + \frac{(90 - 117)^2}{117} + \frac{(110 - 117)^2}{117} + \frac{(150 - 117)^2}{117}$$

$$\chi^2 = 63.83$$

Después de efectuar los cálculos matemáticos se obtuvo el valor de Chi cuadro igual a 63.83.

3.4.3.8. Análisis de la prueba de Chi cuadrado

Una vez obtenida el resultado del Chi cuadrado, se procedió a verificar el valor del Chi cuadrado, en la tabla de distribuciones de Chi cuadrado, teniendo presente el parámetro de significancia del 0.05 que corresponde al 5% que determina mayor certeza en la seguridad de los resultados, y con los grados de libertad establecidos. En la **Figura 1-3**, se muestra la tabla de distribuciones de Chi cuadrado.

DISTRIBUCION DE χ^2

Grados de libertad	Probabilidad										
	0,95	0,90	0,80	0,70	0,50	0,30	0,20	0,10	0,05	0,01	0,001
1	0,004	0,02	0,06	0,15	0,46	1,07	1,64	2,71	3,84	6,64	10,83
2	0,10	0,21	0,45	0,71	1,39	2,41	3,22	4,60	5,99	9,21	13,82
3	0,35	0,58	1,01	1,42	2,37	3,66	4,64	6,25	7,82	11,34	16,27
4	0,71	1,06	1,65	2,20	3,36	4,88	5,99	7,78	9,49	13,28	18,47
5	1,14	1,61	2,34	3,00	4,35	6,06	7,29	9,24	11,07	15,09	20,52
6	1,63	2,20	3,07	3,83	5,35	7,23	8,56	10,64	12,59	16,81	22,46
7	2,17	2,83	3,82	4,67	6,35	8,38	9,80	12,02	14,07	18,48	24,32
8	2,73	3,49	4,59	5,53	7,34	9,52	11,03	13,36	15,51	20,09	26,12
9	3,32	4,17	5,38	6,39	8,34	10,66	12,24	14,68	16,92	21,67	27,88
10	3,94	4,86	6,18	7,27	9,34	11,78	13,44	15,99	18,31	23,21	29,59
	No significativo								Significativo		

Figura 1-3: Distribución Chi cuadrado.

Fuente: <https://cristina92sm.wordpress.com/2011/05/15/ejercicio-del-seminario-nueve-chi-cuadrado/>

Estableciendo un nivel de significancia 0.05 que equivale al 5% y con grado de libertad 2, se obtiene el valor de Chi cuadrado 63.83, y evaluado ese valor en la tabla de Chi cuadrado, se tiene que equivale a 5.99 que es un valor menor de 63.83, de esa manera se rechaza la hipótesis nula planteada **la aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz no mejorará en los niños la identificación de figuras geométricas**, y se acepta la hipótesis alternativa planteada, **la aplicación móvil que genera y mueve figuras geométricas mediante patrones de voz mejorará en los niños la identificación de figuras geométricas**.

CONCLUSIONES

Para el desarrollo de la aplicación móvil se usa el lenguaje de programación Kotlin, la librería OpenGL ES para visualizar y mover las figuras geométrica en la pantalla, logrando de esa manera desarrollar la aplicación móvil de acuerdo con los requerimientos planteados.

En cuanto al análisis de las figuras geométricas recomendadas para identificar en los niños del nivel inicial, los docentes han manifestado que no existe ciertas figuras específicas recomendadas, si no que interviene sus habilidades cognitivas para identificar figuras geométricas, por ende, hemos llega a un acuerdo con los docentes de la institución generar las siguientes figuras geométricas línea, triángulo, cuadrado, rectángulo, círculo, cubo esfera, polígono y cilindro, para la aplicación móvil.

Mediante el empleo de la metodología ágil SCRUM en el desarrollo de la aplicación móvil, se separan las tareas por Sprint, presentando así avances al cliente en tiempos cortos, logrando validar la funcionalidad que esta tenga y adaptar posibles cambios que pudieran ocurrir en el desarrollo del proyecto. Mediante la aplicación de esta metodología se desarrollan, un total de 4 historias técnicas, 11 historias de usuario, 12 tareas de ingeniería y 27 pruebas de aceptación, las cuales fueron todas exitosas.

Aplicando la norma ISO 9241-11, hemos obtenido como resultado en los parámetros de eficiencia un valor porcentual de 57,66%; en el parámetro de eficiencia un valor porcentual de 70,51%; y en el parámetro de satisfacción un valor porcentual de 96,15%; esto permite deducir que el resultado final del proyecto ha sido satisfactorio.

RECOMENDACIONES

Utilizar el lenguaje de programación Kotlin ya que corrige errores comunes de otros lenguajes en el caso de punteros nulos, verbosidad de código, por otra parte, tiene ventajas en ser interoperable, versátil, seguro y fácil de aprender a demás, usar la librería OpenGL ES, ya que es una librería abierta para el aporte de las otras empresas en el mantenimiento de esta, que ayudan en la generación y movimiento de las figuras en un dispositivo móvil, con muy pocos recursos.

Utilizar la metodología ágil SCRUM en el desarrollo de proyectos informáticos, esta metodología proporciona recomendaciones y buenas prácticas durante el desarrollo de estos tipos de proyectos, como reuniones constantes entre el usuario, cliente, y desarrollador, también busca una buena relación de comunicación entre todos los stakeholders; haciendo que se desarrolle los Sprints en tiempos establecidos y no exista retrasos.

Se recomienda utilizar la norma ISO 9241-11, para evaluar la usabilidad del software ya que sus métricas son: eficiencia, eficacia y satisfacción, estos criterios permiten conocer la calidad que tiene la aplicación móvil directamente con el cliente, quien opina al respecto sobre el producto desarrollado.

BIBLIOGRAFÍA

APOLO DÍAZ, K. B., & COBA CASTILLO, N. J. Implementación de un módulo de reconocimiento de voz para niños mediante el procesamiento de señales aplicado en un caso práctico [En línea] (Trabajo de titulación). (Ingeniería). Escuela Superior Politécnica de Chimborazo, FIE, Escuela de ingeniería en electrónica, telecomunicaciones y redes. Riobamba - Ecuador. 2017, pp. 12–13. [Consulta: 31 de agosto del 2018]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/6365>

ARTHANA, I. K. R., PRADNYANA, I. M. A. & DANTES, G. R. " Usability testing on website wadaya based on ISO 9241-11" Journal of Physics: Conference Series [En línea], 2019, (Indonesia) 1165(1), p. 4. [Consulta: 12 de febrero del 2020]. ISSN 012-012. Disponible en: [10.1088/1742-6596/1165/1/012012](https://doi.org/10.1088/1742-6596/1165/1/012012).

BAZ ALONSO, A., FERREIRA ARTIME, I., ÁLVAREZ RODRÍGUEZ., M. & GARCÍA BANIELLO, R. "Dispositivos móviles". E.P.S.I.G (2011), (España) p. 1.

BUSTOS, M., PEREZ, N., & BERÓN, M. "PLATAFORMAS PARA EL DESARROLLO DE APLICACIONES MÓVILES". Universidad Nacional de San Luis (2015), (Argentina) 1, p. 5.

BRUCE, S. *Mathwords: Geometric Figure* [blog]. [Consulta: 5 de febrero 2019]. Disponible en: http://www.mathwords.com/g/geometric_figure.htm

CALLEJAS CUERVO, M., ALARCÓN ALDAN, A., & ÁLVAREZ CARREÑO, A. "Software quality models, a state of the art". Entramado [En línea], 2017, (Colombia) 13(1), pp. 236–250. [Consulta: 12 de febrero del 2020]. Disponible en: [10.18041/entramado.2017v13n1.25125](https://doi.org/10.18041/entramado.2017v13n1.25125)

CHOUDHARI, W., & RANJANKAR, R. " A Survey Paper on OpenGL ES". International Journal on Recent and Innovation Trends in Computing and Communication [En línea], 2015, (India) 3(4), pp. 1755–1758. [Consulta: 31 de agosto del 2018]. Disponible en: <https://ijritcc.com/index.php/ijritcc/article/view/4121>.

DEBUIRE ENRÍQUEZ, B. *ANÁLISIS Y ESTUDIO DE LA ILUMINACIÓN Y COLOR EN OPENGL* (Trabajo de titulación) (Ingeniería). UNIVERSIDAD TÉCNICA DEL NORTE, Facultad de Ingeniería en Ciencias Aplicadas, Escuela de Ingeniería en Sistemas Computacionales, Ibarra - Ecuador. 2008. pp. 127.

FRANCIS, M. "A Developer's First Look At Android". *Linux for you* (2008), pp. 48–50.

GONZALEZ, T., DIAZ-HERRERA, J. AND TUCKER, A. (2014) *Computing Handbook: Computer Science and Software Engineering* [En línea]. 3ª ed. Allen Tucker, 2014. [Consulta: 9 de diciembre del 2018]. Disponible en: <https://books.google.com.ec/books?id=wyHSBQAAQBAJ&pg=SA32-PA1&dq=geometric+primitives+in+computer+graphics&hl=es-419&sa=X&ved=0ahUKEwiH9JPQ54bfAhXktlkKHScrC6sQ6AEIVzAF#v=onepage&q=geometric primitives in computer graphics&f=false>

GUAMÁN CAMPOVERDE, V. & CUVI OCAÑA, N. Implementación de una Aplicación Móvil de Realidad Aumentada para el Proceso Enseñanza - Aprendizaje de la Flora del Campus Epoch. Caso Práctico: Escuela de Ingeniería Forestal (Trabajo de titulación). (Ingeniería) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2015. p. 26. [Consulta: 29 de agosto del 2018]. Disponible en: <http://dspace.esepoch.edu.ec/handle/123456789/4340>

INUKOLLU, V., KESHAMON, D., KANG, T., & INUKOLLU, M. "Factors Influencing Quality of Mobile Apps: Role of Mobile App Development Life Cycle". *International Journal of Software Engineering & Applications* [En línea], 2014, (USA), 5(5), pp. 15–34. [Consulta: 28 de enero del 2019]. ISSN 0962-1083. Disponible en: doi: 10.5121/ijsea.2014.5502

JANGID, M. "Kotlin The unrivalled android programming language lineage". *Imperial Journal of Interdisciplinary Research (IJIR)*, n° 3(8), (India) pp. 256–259

JUNGHYUN, H. *Introduction to Computer Graphics with OpenGL ES* [En línea]. Boca Taton Nueva York - USA. CRC Press, 2018. [Consulta: 30 de agosto del 2018]. Disponible en: <https://books.google.com.ec/books?id=EP5cDwAAQBAJ&lpg=PR4&ots=FZOcxGsBnF&dq=978-1-4987-4892-6&pg=PR4#v=onepage&q=978-1-4987-4892-6&f=false>

KAISER, A., YBANEZ ZEPEDA, J. A. & BOUBEKEUR, T. "A Survey of Simple Geometric Primitives Detection Methods for Captured 3D Data", *Computer Graphics Forum* [En línea]. 2018, (Francia) [Consulta: 10 de diciembre del 2018]. Disponible en: doi: 10.1111/cgf.13451

LEIVA, A. *Kotlin for Android Developers Learn Kotlin the easy way while developing an Android App*. 7ª ed, 2018 pp. 44

LISANDRO, D., GALDAMEZ, N., THOMAS, P., CORBALAN, L., & PESADO, P. "Análisis

Experimental de desarrollo de Aplicaciones Móviles Multiplataforma", XX Congreso Argentino de Ciencias de la Computación [En línea], 2014, (Argentina) [Consulta: 27 de noviembre del 2018]. ISSN 0001-5385. Disponible en: http://sedici.unlp.edu.ar/bitstream/handle/10915/42355/Documento_completo.pdf?sequence=1

LOSA, G., RISTANOVIĆ, D., RISTANOVIĆ, D., ZALETEL, I., & BELTRAMINELLI, S. "From Fractal Geometry to Fractal Analysis", Applied Mathematics. Scientific Research Publishing [En línea], 2016, (Serbia) 07(04), pp. 346–354. [Consulta: 2 de febrero del 2019]. ISSN 2152-7385. Disponible en: doi: 10.4236/am.2016.74032

MADUEÑO ORTEGA, M. Control Teleoperado del robot RV-M1 mediante dispositivo móvil y Realidad Aumentada [En línea], (Trabajo de titulación). (Ingeniería). Universitat Politècnica de Catalunya. Ingeniería de Sistemas, Automat. e Informática Ind. Catalunya - España, 2013. pp. 53. [Consulta: 11 de febrero del 2020]. Disponible en: <https://upcommons.upc.edu/handle/2099.1/19144>

MARCIN MOSKALA, I. *Android Development with Kotlin* [En línea]. Birmingham - Inglaterra. Published by Packt Publishing Ltd. 2017. [Consulta: 27 de noviembre del 2018]. Disponible en: <https://books.google.com.ec/books?id=PJZGDwAAQBAJ&lpg=PA8&dq=kotlin> is a&pg=PP1#v=onepage&q=kotlin is a&f=false

MASCHERONI, A., GREINER, C., PETRIS, R., DAPOZO, G., & ESTAYNO, M. "Calidad de software e ingeniería de usabilidad", XIV Workshop de de Investigadores en Ciencias de la Computación [En línea], 2012, (Argentina) (1), pp. 656–659. [Consulta: 12 de febrero del 2020]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/19202>

MCREYNOLDS, T. AND BLYTHE, D. "Geometry Representation and Modeling", Advanced Graphics Programming Using OpenGL [En línea]. 2005, pp. 3–17. [Consulta: 29 de agosto del 2018]. Disponible en: doi: 10.1016/B978-155860659-3.50003-2.

MONTE GALIANO, J. *Implantar scrum con éxito*. Riobamba - Ecuador: UOC, 2016 pp. 13–18

MORALES MONTERO, H., & YÁNEZ JÁCOME, C. Diseño e implementación de un sistema inalámbrico que permita interactuar con el entorno a personas con limitación motriz utilizando el movimiento de ojos y comandos de voz. (Trabajo de titulación). (Ingeniería) Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2018. pp. 46. [Consulta: 31 de agosto del 2018]. Disponible en: <http://dspace.espace.edu.ec/handle/123456789/8448>

MORALE, A. & ROSAS, L. "Una propuesta para el desarrollo de modelos geométricos en las Educadoras de Párvulos: El caso del polígono", *Estudios pedagógicos (Valdivia)* [En línea]. 2016 (Valdivia) 42(2), pp. 247–267. [Consulta: 5 de febrero del 2019]. ISSN 0718-0705. Disponible en: doi: 10.4067/S0718-07052016000200014

MOSCOSO DOMINGUÉZ, M. "Control doméstico por voz desde Android Desarrollo de aplicaciones de software libre", 2014, (Catalunya) p. 63. Disponible en: <http://hdl.handle.net/10609/34161>

MOUNTH, D. *Computer Graphics*. Maryland - USE. University of Maryland, 2013, p. 11.

ONTARIO, E. *A Guide to Effective Instruction in Mathematics*. 3ª ed. Ontario - Francia: Queen's Printer for Ontario, 2005, pp. 7-8.

POUDEL, A. MOBILE APPLICATION DEVELOPMENT FOR ANDROID OPERATING SYSTEM. (Trabajo de titulación). (Ingeniería). TURKU UNIVERSITY OF APPLIED SCIENCES. Information Technology. Turku - Finlandia. 2013. pp. 6

RASHID, R. et al. "Security system using biometric technology: Design and implementation of Voice Recognition System (VRS)", 2008 International Conference on Computer and Communication Engineering. IEEE [En línea], 2008, (Kuala Lumpur) pp. 898–902. [Consulta: 31 de agosto del 2018]. Disponible en: doi: 10.1109/ICCCE.2008.4580735

ROHAUT, S. *LINUX Dominar la administración del sistema, Journal of Visual Languages & Computing*. 3ª ed. Barcelona: Ediciones ENI, 2015. pp. 46.

ROJAS ÁLVAREZ, C. J. *Introducción a la geometría* [En línea]. Universidad del Norte. Ibarra-Ecuador: ProQuest Ebook Central, 2016 [Consulta: 15 de Diciembre 2018]. Disponible en: <https://ebookcentral.proquest.com/lib/epochsp/detail.action?docID=4183562>

SANTAMARÍA PUERTO, G. A. & HERNÁNDEZ RINCÓN, E. "Aplicaciones médicas móviles: Definiciones, beneficios y riesgos", *Salud Uninorte* [En línea], 2015, (Barranquilla) 31(3), pp. 599–607. [Consulta: 28 de enero 2019]. ISSN 0120-5552. Disponible en: doi: 10.14482/sun.31.3.7662.

SCRUM MANAGER. [blog], 2016. [Consulta: 6 noviembre 2019]. Disponible en: <https://doi.org/1607208414838>.

SOBALVARRO CHAVARRÍA, L. M. & CAMACHO ÁLVAREZ, M. M. "El aprendizaje de la noción de objeto según la forma en niños de educación preescolar: Propuesta geometría en movimiento", *Revista Educación* [En línea], 2018, (Costa Rica) 42(2), p. 12. [Consulta: 14 de diciembre 2018]. ISSN 2215-2644. Disponible en: doi: 10.15517/revedu.v42i2.28195.

WALDON, R. ESTUDIO DE LOS MÉTODOS DE REPRESENTACIÓN DE OBJETOS GRÁFICOS PARA APLICACIONES ANDROID EN DISPOSITIVOS MÓVILES. CASO SISTEMA FAMILIOGRAMA DEL MCDS (Trabajo de titulación). (Ingeniería). Escuela Superior Politécnica de Chimborazo, FIE, Carrera de Ingeniería en Sistemas, Riobamba-Ecuador, 2015, pp. 124.

YONGXIN ZHUANG, Y. L. "Display technique of mobile video monitor on android". *International Symposium on Computers & Informatics*, 2015 pp. 2475–2482.

ZÚÑIGA, O. Estrategias activas en la iniciación a la matemática en los niños y niñas de 5 a 6 años en la Unidad Educativa Julio María Matovelle sector norte [En línea] (Trabajo de titulación). (Ingeniería) Universidad Central del Ecuador, Quito, Ecuador. 2016. pp. 50. [Consulta: 14 December 2018] Disponible en: <http://www.dspace.uce.edu.ec/handle/25000/12268>.

ANEXOS

Anexo A: Hoja de gestión de riesgos

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R_01		FECHA: 4/abril/2019	
Probabilidad: Alta Valor: 3	Impacto: Alto Valor: 3	Exposición: Alta Valor: 9	Prioridad: 1
DESCRIPCIÓN: Cambio constante de los requerimientos			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Mala comunicación entre el desarrollar con los clientes • Desacuerdo entre los docentes de la escuela en exponer sus requerimientos 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso en el proyecto • Aumento de los costos en el desarrollo 			
REDUCCIÓN:			
<ul style="list-style-type: none"> • Comunicación constante con el cliente • Pedir que lleguen a acuerdos al exponer requerimientos por parte de los docentes 			
SUPERVISION:			
<ul style="list-style-type: none"> • Analizar el tiempo que van a tener estos cambios. • Establecer nuevas estrategias de comunicación con el cliente. 			
GESTIÓN:			
<ul style="list-style-type: none"> • Llevar un seguimiento continuo en el desarrollo de cada requerimiento. • Reconocer el riesgo y actuar rápidamente para que el daño no sea mayor. 			
ESTADO ACTUAL:			
<ul style="list-style-type: none"> • Fase de reducción iniciada ✓ • Fase de supervisión iniciada • Gestionando el riesgo 			
RESPONSABLE:			
<ul style="list-style-type: none"> • Pedro Guayllas 			

Realizado por: Guayllas, Pedro, 2019

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R_02		FECHA: 4/abril/2019	
Probabilidad: Media Valor: 2	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 3
DESCRIPCIÓN: Falta de conocimiento en el uso de la tecnología para el desarrollo			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • El proyecto requiere el uso de tecnologías más eficientes • Se requiere incorporar el uso de nuevas tecnologías en desarrollo de proyecto 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso en el proyecto • Aumento de los costos en el desarrollo 			
REDUCCIÓN:			
<ul style="list-style-type: none"> • Asesoramiento previo en la incorporación de nuevas tecnologías 			
SUPERVISION:			
<ul style="list-style-type: none"> • Analizar el tiempo que va a requerir incorporar en el uso de una nueva tecnología • Fomentar el aprendizaje continuo en uso de la tecnología 			

GESTIÓN:
<ul style="list-style-type: none"> • Evaluar conocimientos previos de la tecnología a utilizar • Reconocer el riesgo y actuar rápidamente para que el daño no sea mayor.
ESTADO ACTUAL:
<ul style="list-style-type: none"> • Fase de reducción iniciada ✓ • Fase de supervisión iniciada • Gestionando el riesgo
RESPONSABLE:
<ul style="list-style-type: none"> • Pedro Guailas

Realizado por: Guailas, Pedro, 2019

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R_03		FECHA: 4/abril/2019	
Probabilidad: Baja Valor: 1	Impacto: Alto Valor: 3	Exposición: Alta Valor: 6	Prioridad: 4
DESCRIPCIÓN: Daños en Hardware utilizado para el desarrollo			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • Los equipos estén deteriorados • Incorporar programas que utilizan mayor los recursos del computador 			
Consecuencias:			
<ul style="list-style-type: none"> • Retraso en el proyecto • Aumento de los costos en el desarrollo 			
REDUCCIÓN:			
<ul style="list-style-type: none"> • Tener en cuenta los requerimientos del Hardware a la hora de instalar las herramientas de desarrollo. • Contar con Hardware estable 			
SUPERVISION:			
<ul style="list-style-type: none"> • Analizar los requerimientos de hardware • Analizar la estabilidad de los equipos. 			
GESTIÓN:			
<ul style="list-style-type: none"> • Evaluar conocimientos previos de la tecnología a utilizar • Reconocer el riesgo y actuar rápidamente para que el daño no sea mayor. 			
ESTADO ACTUAL:			
<ul style="list-style-type: none"> • Fase de reducción iniciada ✓ • Fase de supervisión iniciada • Gestionando el riesgo 			
RESPONSABLE:			
<ul style="list-style-type: none"> • Pedro Guailas 			

Realizado por: Guailas, Pedro, 2019

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R_04		FECHA: 4/abril/2019	
Probabilidad: Media Valor: 2	Impacto: Crítico Valor: 4	Exposición: Alta Valor: 8	Prioridad: 2
DESCRIPCIÓN: Mala planificación			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • No establece todos los requerimientos al inicio del proyecto 			

<ul style="list-style-type: none"> • Mala asignación de horas a las tareas <p>Consecuencias:</p> <ul style="list-style-type: none"> • Presión por cumplir con la tarea • Aumento de los costos en el desarrollo
<p>REDUCCIÓN:</p> <ul style="list-style-type: none"> • Trata de obtener todos los requerimientos al inicio del proyecto • Asignar horas extras para cumplir a tiempo
<p>SUPERVISION:</p> <ul style="list-style-type: none"> • Analizar las horas para cada requerimiento requerimientos • Analizar el avance del proyecto para acelerar los tiempos.
<p>GESTIÓN:</p> <ul style="list-style-type: none"> • Evaluar el nivel de desempeño cuando se avanza el proyecto • Reconocer el riesgo y actuar rápidamente para que el daño no sea mayor.
<p>ESTADO ACTUAL:</p> <ul style="list-style-type: none"> • Fase de reducción iniciada ✓ • Fase de supervisión iniciada • Gestionando el riesgo
<p>RESPONSABLE:</p> <ul style="list-style-type: none"> • Pedro Guailas

Realizado por: Guailas, Pedro, 2019

HOJA DE GESTIÓN DEL RIESGO			
ID. DEL RIESGO: R_05		FECHA: 4/abril/2019	
Probabilidad: Media Valor: 2	Impacto: Baja Valor: 1	Exposición: Baja Valor: 2	Prioridad: 5
DESCRIPCIÓN: Mal diseño de la interfaz de usuario			
REFINAMIENTO:			
Causas:			
<ul style="list-style-type: none"> • No contar con conocimientos de diseño • No tener presente las recomendaciones del cliente 			
Consecuencias:			
<ul style="list-style-type: none"> • No es atractivo para el usuario • Menor uso de la aplicación 			
REDUCCIÓN:			
<ul style="list-style-type: none"> • Pedir ayuda a los diseñadores • Hacer caso las recomendaciones de los clientes 			
SUPERVISION:			
<ul style="list-style-type: none"> • Analizar el diseño de la interfaz • Analizar las recomendaciones del usuario 			
GESTIÓN:			
<ul style="list-style-type: none"> • Evaluar el diseño de la interfaz con los usuarios • Reconocer el riesgo y actuar rápidamente para que el daño no sea mayor. 			
ESTADO ACTUAL:			
<ul style="list-style-type: none"> • Fase de reducción iniciada ✓ • Fase de supervisión iniciada • Gestionando el riesgo 			
RESPONSABLE:			
<ul style="list-style-type: none"> • Pedro Guailas 			

Realizado por: Guailas, Pedro, 2019

Anexo B: Definición del estándar de codificación

Estándar para Clases

Reglas para la definición de clases

- Usar palabras completas, evite abreviaturas o acrónimos, a menos que sean mucho más usadas que el nombre completo.
- Los nombres de clase deben empezar con una letra mayúscula y el resto de las letras deben estar escritas en minúscula. En el caso que pueda tener más de una palabra, las primeras letras de cada palabra interna deben ser con mayúscula.
- Los nombres de clase no pueden contener espacios ni caracteres especiales, sólo son permitidas las letras de la “a” a la “z” y los números del 0 al 9.

Ejemplo:

```
class Cube{  
}
```

Estándar para Atributos

Reglas para definición de atributos de clases

- Los nombres de atributos deben estar escritas en minúsculas.
- Los nombres de atributo no pueden contener espacios, sólo son permitidos las letras de la “a” a la “z” y los números del 0 al 9. Separados por el guion bajo.
- Los nombres de atributo no pueden ser verbos.

Ejemplo:

```
var lado=5
```

Estándar para Métodos

Reglas para definir los métodos

- Los nombres de métodos deben estar escritas en minúsculas.
- Los nombres de los métodos deben ser verbos o palabras que identifiquen de manera general el objetivo del método.
- Los nombres de los métodos no pueden contener espacios ni caracteres especiales, sólo son permitidas las letras de la “a” a la “z” y los números del 0 al 9.
- Si el nombre de método requiere estar compuesto por más de una palabra, cada palabra adicional debe empezar con mayúscula.

Ejemplo:

```
fun Dibujar(){  
}
```

Estándar para Constantes

Reglas para definir constantes

- Los nombres de constantes deben estar escritas en mayúscula.
- Los nombres de constantes deben identifiquen de manera general el objetivo de la constante.
- Los nombres de las constantes no pueden contener espacios ni caracteres especiales, sólo son permitidas las letras de la “A” a la “Z” y los números del 0 al 9.

- Si el nombre de la constante requiere estar compuesto por más de una palabra, cada palabra adicional debe estar separado por un guion bajo.

Ejemplo:

const final COIORS_PER_VERTEX = 4

Definición de archivos

Los archivos que utilizan en el proyecto se ejemplifican en la siguiente tabla.

Definición	Ejemplo
Código Kotlin	Cube.kt
Imagen	Microfono.png
Audio	Intro.mp3

Realizado por: Guaitas, Pedro, 2019

Anexo C: Manual Técnico

Historias de usuario y tareas de ingeniería con sus respectivas pruebas de aceptación.

HISTORIA TÉCNICA	
Número: HT01	Nombre de la historia: Diseño de la interfaz.
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como desarrollador deseo poder diseñar todas las interfaces de la aplicación, para su posterior desarrollo.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento. <ul style="list-style-type: none">• Verificar que los colores de diseño estén de acuerdo con los especificados por el cliente.	

Realizado por: Guayllas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-01.HT01	Historia de Usuario: HT01 Diseño de la interfaz
Nombre: Verificar que los colores de diseño estén de acuerdo con los especificados por el cliente.	
Responsable: Pedro Guayllas	Fecha: 7-11-2018
Descripción: Se verificará que el diseño de las interfaces cuente con los colores establecidos por el cliente.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Que las interfaces no estén diseñadas.	
Pasos de ejecución: <ol style="list-style-type: none">1. Tener instaladas las herramientas necesarias.2. Diseñar cada una de las interfaces para la aplicación móvil.3. Comprobar que cumplan con los colores ya establecidos.	
Resultado esperado: Las interfaces se diseñaron de forma correcta.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guayllas, Pedro, 2019

HISTORIA TÉCNICA	
Número: HT02	Nombre de la historia: Definición de estándar de codificación
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 1
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como desarrollador deseo poder definir el estándar de codificación, para el desarrollo de la aplicación móvil.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Aplicar el estándar de codificar definido por el desarrollador. 	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-02. HT02	Historia de Usuario: HT02 Definición de estándar de codificación.
Nombre: Aplicar el estándar de codificar definido por el desarrollador.	
Responsable: Pedro Guailas	Fecha: 9-11-2018
Descripción: Se verificará que se aplique el estándar de codificar definido por el desarrollador y el cliente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el estándar de codificación no esté definido. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Revisar los tipos de estándar para desarrollo de software. 2. Escoger un estándar de codificación para el desarrollo de aplicaciones móviles. 3. Implementar el estándar de codificación en el desarrollo. 	
Resultado esperado: Que el estándar de codificación establecido se aplique durante el desarrollo de la aplicación móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

HISTORIA TÉCNICA	
Número: HT03	Nombre de la historia: Definición de la arquitectura de software
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como desarrollador deseo poder definir la arquitectura de software, para el desarrollo de la aplicación móvil.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Aplicar la arquitectura de software definida el desarrollador. 	

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-03. HT03	Historia de Usuario: HT03 Definición de la arquitectura de software
Nombre: Aplicar la arquitectura de software definida el desarrollador	
Responsable: Pedro Guaitas	Fecha: 14-11-2018
Descripción: Se verificará que se aplique la arquitectura de software durante el desarrollo y la documentación de la aplicación móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Establecer una arquitectura de software para el desarrollo del sistema. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Realizar un estudio de las diferentes arquitecturas de software que existen en el mercado. 2. Establecer una arquitectura de software. 3. Aplicar la arquitectura de software durante el desarrollo de la aplicación móvil. 	
Resultado esperado: Que la arquitectura de software sea aplicada durante todo el desarrollo de la aplicación móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guaitas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU01	Nombre de la historia: Visualizar línea
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 2
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar una línea, para que los niños reconozcan e identifiquen una línea.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que la línea se grafique correctamente en la pantalla del dispositivo móvil. 	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-04. HU01	Historia de Usuario: HU01 Visualizar línea
Nombre: Que la línea se grafique correctamente en la pantalla del dispositivo móvil.	
Responsable: Pedro Guailas	Fecha: 16-11-2018
Descripción: Se verificará que la línea se grafique correctamente en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que la línea este graficada. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Selección la línea en la aplicación móvil. 3. Visualizar que se grafique dentro del rango de la pantalla del dispositivo móvil. 	
Resultado esperado: Que la línea se grafique correctamente en la pantalla del dispositivo móvil	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

TAREA DE INGENIERÍA
Historia de Usuario: HU01 Visualizar línea

Número de Tarea: TI-01. HU01	Nombre de Tarea: Realizar el método de codificación para la visualización de una línea.
Tipo de Tarea: Desarrollo	Puntos Estimados: 20
Fecha Inicio: 14-11-2018	Fecha Fin: 16-11-2018
Programador Responsable: Pedro Guailas	
Descripción: Crear el método para realizar la codificación para visualizar una línea en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que el método visualizar línea este correctamente desarrollado.	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-01.TI-01	Tarea de Ingeniería: TI-01 Realizar el método de codificación para la visualización de una línea.
Nombre: Verificar que el método visualizar línea este correctamente desarrollado.	
Responsable: Pedro Guailas	Fecha: 16-11-2018
Descripción: Se verificará que el método de codificación visualizar línea este desarrollado correctamente para graficar la línea en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar el método de visualizar línea. 2. Verificar que el método este desarrollado correctamente. 3. Que el método grafique correctamente la línea. 4. Visualizar la línea en la pantalla del dispositivo móvil. 	
Resultado esperado: Que la funcionalidad graficar línea funcione correctamente y muestre en la pantalla del dispositivo móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU02	Nombre de la historia: Visualizar triángulo
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 3
Prioridad en el Negocio: Alta	Puntos Estimados: 20

Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un triángulo, para que los niños reconozcan e identifiquen la figura geométrica de un triángulo.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que el triángulo este diseñado de acuerdo al estándar de interfaz establecido anteriormente. 	

Realizado por: Guayllas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-05.HU02	Historia de Usuario: HU02 Visualizar triángulo
Nombre: Que el triángulo este diseñado de acuerdo al estándar de interfaz establecido anteriormente.	
Responsable: Pedro Guayllas	Fecha: 21-11-2018
Descripción: Se verificará que el triángulo este diseñado de acuerdo al estándar de interfaz establecido anteriormente.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el triángulo se encuentre diseñado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Seleccionar el triángulo en la aplicación móvil. 3. Visualizar el triángulo en la aplicación móvil. 4. Verificar que los colores sean los correctos. 	
Resultado esperado: Que el triángulo este diseñado de acuerdo al estándar de interfaz establecido anteriormente.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guayllas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU02 Visualizar triángulo	
Número de Tarea: TI-02. HU02	Nombre de Tarea: Realizar el método de codificación para la visualización de un triángulo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 20
Fecha Inicio: 19-11-2018	Fecha Fin: 21-11-2018
Programador Responsable: Pedro Guayllas	
Descripción: Crear y desarrollar el método para graficar y visualizar un triángulo en la pantalla del dispositivo móvil.	

(Reverso) Pruebas de Aceptación

Verificar que el método visualizar triángulo este correctamente desarrollado.

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-02.TI-02

Tarea de Ingeniería: TI-02 Realizar el método de codificación para la visualización de un triángulo.

Nombre: Verificar que el método visualizar triángulo este correctamente desarrollado.

Responsable: Pedro Guaitas

Fecha: 21-11-2018

Descripción: Se verificará que el método de codificación visualizar triángulo este desarrollado correctamente para graficar el triángulo en la pantalla del dispositivo móvil.

Condiciones de Ejecución:

- Que el método este creado y desarrollado.

Pasos de ejecución:

1. Ejecutar el método de visualizar triángulo.
2. Verificar que el método este desarrollado correctamente.
3. Que el método grafique correctamente el triángulo.
4. Visualizar el triángulo en la pantalla del dispositivo móvil.

Resultado esperado: Que la funcionalidad graficar triángulo funcione correctamente y muestre en la pantalla del dispositivo móvil.

Evaluación de la prueba: Exitosa.

Realizado por: Guaitas, Pedro, 2019

HISTORIA DE USUARIO

Número: HU03

Nombre de la historia: Visualizar cuadrado

Modificación de historia de usuario:

Usuario: Cliente

Sprint Asignada: 3

Prioridad en el Negocio: Alta

Puntos Estimados: 20

Riesgo en el Desarrollo: Media

Puntos Reales: 20

Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un cuadrado, para que los niños reconozcan e identifiquen la figura geométrica de un cuadrado.

Observaciones:

(Reverso) Pruebas de Aceptación

Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.

- Que el cuadrado se visualice de forma correcta en la pantalla del dispositivo móvil.

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-06.HU03

Historia de Usuario: HU03 Visualizar cuadrado

Nombre: Que el cuadrado se visualice de forma correcta en la pantalla del dispositivo móvil.

Responsable: Pedro Guailas

Fecha: 23-11-2018

Descripción: Se verificará que el cuadrado se visualice de forma correcta en la pantalla del dispositivo móvil.

Condiciones de Ejecución:

- Que la figura del cuadrado se encuentre diseñado.

Pasos de ejecución:

1. Abrir la aplicación móvil.
2. Seleccionar la figura del cuadrado en la aplicación móvil.
3. Visualizar el cuadrado en la pantalla de la aplicación móvil.
4. Verificar que se visualice de forma correcta.

Resultado esperado: Que el cuadrado se visualice de forma correcta en la pantalla del dispositivo móvil.

Evaluación de la prueba: Exitosa.

Realizado por: Guailas, Pedro, 2019

TAREA DE INGENIERÍA

Historia de Usuario: HU03 Visualizar cuadrado

Número de Tarea: TI-03. HU03

Nombre de Tarea: Realizar el método de codificación para la visualización de un cuadrado.

Tipo de Tarea: Desarrollo

Puntos Estimados: 20

Fecha Inicio: 21-11-2018

Fecha Fin: 23-11-2018

Programador Responsable: Pedro Guailas

Descripción: Crear y desarrollar el método para graficar y visualizar un cuadrado en la pantalla del dispositivo móvil.

(Reverso) Pruebas de Aceptación

Verificar que el método visualizar cuadrado este correctamente desarrollado.

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-03.TI-03	Tarea de Ingeniería: TI-03 Realizar el método de codificación para la visualización de un cuadrado.
Nombre: Verificar que el método visualizar cuadrado este correctamente desarrollado	
Responsable: Pedro Guailas	Fecha: 23-11-2018
Descripción: Se verificará que el método de codificación visualizar cuadrado este desarrollado correctamente para graficar la figura del cuadrado en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 5. Ejecutar el método de visualizar cuadrado. 6. Verificar que el método este desarrollado correctamente. 7. Que el método grafique correctamente el cuadrado. 8. Visualizar el cuadrado en la pantalla del dispositivo móvil. 	
Resultado esperado: Que la funcionalidad graficar cuadrado funcione correctamente y muestre en la pantalla del dispositivo móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU04	Nombre de la historia: Visualizar círculo
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un círculo, para que los niños reconozcan e identifiquen la figura geométrica de un círculo.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que el círculo cuente con los colores que se establecieron en el estándar de interfaz. 	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-07.HU04	Historia de Usuario: HU04 Visualizar círculo
Nombre: Que el círculo cuente con los colores que se establecieron en el estándar de interfaz.	
Responsable: Pedro Guaitas	Fecha: 28-11-2018
Descripción: Se verificará que el círculo cuente con los colores que se establecieron en el estándar de interfaz.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que la figura del círculo se encuentre diseñado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Seleccionar la figura del círculo en la aplicación móvil. 3. Visualizar el círculo en la pantalla de la aplicación móvil. 4. Verificar que los colores estén acorde al estándar de interfaz establecido. 	
Resultado esperado: Que el círculo este diseñado con los colores establecidos en el estándar de interfaz.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guaitas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU04 Visualizar círculo	
Número de Tarea: TI-04. HU04	Nombre de Tarea: Realizar el método de codificación para la visualización de un círculo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 20
Fecha Inicio: 26-11-2018	Fecha Fin: 28-11-2018
Programador Responsable: Pedro Guaitas	
Descripción: Crear y desarrollar el método para graficar y visualizar un círculo en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que el método visualizar círculo este correctamente desarrollado.	

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-04.TI-04	Tarea de Ingeniería: TI-04 Realizar el método de codificación para la visualización de un círculo.
Nombre: Verificar que el método visualizar círculo este correctamente desarrollado.	
Responsable: Pedro Guaitas	Fecha: 28-11-2018

Descripción: Se verificará que el método de codificación visualizar círculo este desarrollado correctamente para graficar la figura del círculo en la pantalla del dispositivo móvil.
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que el método este creado y desarrollado.
Pasos de ejecución: <ol style="list-style-type: none"> 1. Ejecutar el método de visualizar círculo. 2. Verificar que el método este desarrollado correctamente. 3. Que el método grafique correctamente el círculo. 4. Visualizar el círculo en la pantalla del dispositivo móvil.
Resultado esperado: Que la funcionalidad graficar círculo funcione correctamente y muestre en la pantalla del dispositivo móvil.
Evaluación de la prueba: Exitosa.

Realizado por: Guayllas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU05	Nombre de la historia: Visualizar cubo.
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 4
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un cubo, para que los niños reconozcan e identifiquen la figura geométrica de un cubo.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento. <ul style="list-style-type: none"> • Que el cubo se visualice de forma correcta en la pantalla del dispositivo móvil. 	

Realizado por: Guayllas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-08.HU05	Historia de Usuario: HU04 Visualizar cubo
Nombre: Que el cubo se visualice de forma correcta en la pantalla del dispositivo móvil.	
Responsable: Pedro Guayllas	Fecha: 30-11-2018

Descripción: Se verificará que el cubo se visualice de forma correcta en la pantalla del dispositivo móvil.
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que la figura del cubo se encuentre diseñado.
Pasos de ejecución: <ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Seleccionar la figura del cubo en la aplicación móvil. 3. Visualizar el cubo en la pantalla de la aplicación móvil. 4. Verificar que la figura del cubo se visualice de forma correcta en la pantalla del dispositivo móvil.
Resultado esperado: Que la figura se visualice de forma correcta en la pantalla.
Evaluación de la prueba: Exitosa.

Realizado por: Guailas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU05 Visualizar cubo	
Número de Tarea: TI-05. HU05	Nombre de Tarea: Realizar el método de codificación para la visualización de un cubo.
Tipo de Tarea: Desarrollo	Puntos Estimados: 20
Fecha Inicio: 28-11-2018	Fecha Fin: 30-11-2018
Programador Responsable: Pedro Guailas	
Descripción: Crear y desarrollar el método para graficar y visualizar un cubo en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que el método visualizar cubo este correctamente desarrollado.	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-05.TI-05	Tarea de Ingeniería: TI-05 Realizar el método de codificación para la visualización de un cubo
Nombre: Verificar que el método visualizar cubo este correctamente desarrollado.	
Responsable: Pedro Guailas	Fecha: 30-11-2018
Descripción: Se verificará que el método de codificación visualizar cubo este desarrollado correctamente para graficar la figura en la pantalla del dispositivo móvil.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	

<ol style="list-style-type: none"> 1. Ejecutar el método de visualizar cubo. 2. Verificar que el método este desarrollado correctamente. 3. Que el método grafique correctamente el cubo. 4. Visualizar el cubo en la pantalla del dispositivo móvil.
Resultado esperado: Que la funcionalidad graficar cubo funcione correctamente y muestre en la pantalla del dispositivo móvil.
Evaluación de la prueba: Exitosa.

Realizado por: Guaitas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU06	Nombre de la historia: Visualizar esfera.
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 5
Prioridad en el Negocio: Alta	Puntos Estimados: 40
Riesgo en el Desarrollo: Media	Puntos Reales: 40
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar una esfera, para que los niños reconozcan e identifiquen la figura geométrica de una esfera.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que la esfera cuente con los colores que se establecieron en el estándar de interfaz. 	

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-09.HU06	Historia de Usuario: HU06 Visualizar esfera
Nombre: Que la esfera cuente con los colores que se establecieron en el estándar de interfaz.	
Responsable: Pedro Guaitas	Fecha: 7-12-2018
Descripción: Se verificará que la esfera cuente con los colores que se establecieron en el estándar de interfaz.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que la figura de la esfera se encuentre diseñado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 	

<ol style="list-style-type: none"> 2. Seleccionar la figura de la esfera en la aplicación móvil. 3. Visualizar la esfera en la pantalla de la aplicación móvil. 4. Verificar que la figura de la esfera contenga los colores establecidos por el cliente.
Resultado esperado: Que los colores de la figura sean los que pidió el cliente.
Evaluación de la prueba: Exitosa.

Realizado por: Guailles, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU06 Visualizar esfera	
Número de Tarea: TI-06. HU06	Nombre de Tarea: Realizar el método de codificación para la visualización de una esfera.
Tipo de Tarea: Desarrollo	Puntos Estimados: 40
Fecha Inicio: 3-12-2018	Fecha Fin: 7-12-2018
Programador Responsable: Pedro Guailles	
Descripción: Crear y desarrollar el método para graficar y visualizar una esfera en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que el método visualizar esfera este correctamente desarrollado.	

Realizado por: Guailles, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-06.TI-06	Tarea de Ingeniería: TI-06 Realizar el método de codificación para la visualización de una esfera.
Nombre: Verificar que el método visualizar esfera este correctamente desarrollado.	
Responsable: Pedro Guailles	Fecha: 7-12-2018
Descripción: Se verificará que el método de codificación visualizar esfera este desarrollado correctamente para graficar la figura en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar el método de visualizar esfera. 2. Verificar que el método este desarrollado correctamente. 3. Que el método grafique correctamente la esfera. 4. Visualizar la esfera en la pantalla del dispositivo móvil. 	
Resultado esperado: Que la funcionalidad graficar esfera funcione correctamente y muestre en la pantalla del dispositivo móvil.	

Evaluación de la prueba: Exitosa.

Realizado por: Guayllas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU07	Nombre de la historia: Visualizar polígono
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 6
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un polígono, para que los niños reconozcan e identifiquen la figura geométrica de un polígono.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento. <ul style="list-style-type: none">• Que el polígono se visualice de forma correcta en la pantalla del dispositivo móvil.	

Realizado por: Guayllas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-10.HU07	Historia de Usuario: HU07 Visualizar polígono
Nombre: Que el polígono se visualice de forma correcta en la pantalla del dispositivo móvil.	
Responsable: Pedro Guayllas	Fecha: 12-12-2018
Descripción: Se verificará que el polígono se visualice de forma correcta en la pantalla del dispositivo móvil.	
Condiciones de Ejecución: <ul style="list-style-type: none">• Que la figura del polígono se encuentre diseñado.	
Pasos de ejecución: <ol style="list-style-type: none">1. Abrir la aplicación móvil.2. Seleccionar la figura de polígono en la aplicación móvil.3. Visualizar el polígono en la pantalla de la aplicación móvil.4. Verificar que la figura se visualice de forma correcta en la pantalla del móvil.	
Resultado esperado: Que la figura se visualice de forma correcta en la pantalla de los dispositivos móviles.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guayllas, Pedro, 2019

TAREA DE INGENIERÍA

Historia de Usuario: HU07 Visualizar polígono

Número de Tarea: TI-07. HU07

Nombre de Tarea: Realizar el método de codificación para la visualización de un polígono.

Tipo de Tarea: Desarrollo

Puntos Estimados: 20

Fecha Inicio: 10-12-2018

Fecha Fin: 12-12-2018

Programador Responsable: Pedro Guailas

Descripción: Crear y desarrollar el método para graficar y visualizar un polígono en la pantalla del dispositivo móvil.

(Reverso) Pruebas de Aceptación

Verificar que el método visualizar polígono este correctamente desarrollado.

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-07.TI-07

Tarea de Ingeniería: TI-07 Realizar el método de codificación para la visualización de un polígono.

Nombre: Verificar que el método visualizar polígono este correctamente desarrollado.

Responsable: Pedro Guailas

Fecha: 12-12-2018

Descripción: Se verificará que el método de codificación visualizar polígono este desarrollado correctamente para graficar la figura en la pantalla del dispositivo móvil.

Condiciones de Ejecución:

- Que el método este creado y desarrollado.

Pasos de ejecución:

1. Ejecutar el método de visualizar polígono.
2. Verificar que el método este desarrollado correctamente.
3. Que el método grafique correctamente el polígono.
4. Visualizar el polígono en la pantalla del dispositivo móvil.

Resultado esperado: Que la funcionalidad graficar polígono funcione correctamente y muestre en la pantalla del dispositivo móvil.

Evaluación de la prueba: Exitosa.

Realizado por: Guailas, Pedro, 2019

HISTORIA DE USUARIO

Número: HU08

Nombre de la historia: Visualizar cilindro

Modificación de historia de usuario:

Usuario: Cliente	Sprint Asignada: 6
Prioridad en el Negocio: Alta	Puntos Estimados: 20
Riesgo en el Desarrollo: Media	Puntos Reales: 20
Descripción: Como cliente quiero que la aplicación móvil me permita visualizar un cilindro, para que los niños reconozcan e identifiquen la figura geométrica de un cilindro.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Que el cilindro se visualice de forma correcta en la pantalla del dispositivo móvil. 	

Realizado por: Guayllas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-11.HU08	Historia de Usuario: HU08 Visualizar cilindro
Nombre: Que el cilindro se visualice de forma correcta en la pantalla del dispositivo móvil.	
Responsable: Pedro Guayllas	Fecha: 14-12-2018
Descripción: Se verificará que el cilindro se visualice de forma correcta en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que la figura del cilindro se encuentre diseñado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Seleccionar la figura del cilindro en la aplicación móvil. 3. Visualizar el cilindro en la pantalla de la aplicación móvil. 4. Verificar que la figura se visualice de forma correcta en la pantalla del móvil. 	
Resultado esperado: Que la figura se visualice de forma correcta en la pantalla de los dispositivos móviles.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guayllas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU08 Visualizar cilindro	
Número de Tarea: TI-08. HU08	Nombre de Tarea: Realizar el método de codificación para la visualización de un cilindro.
Tipo de Tarea: Desarrollo	Puntos Estimados: 20

Fecha Inicio: 12-12-2018	Fecha Fin: 14-12-2018
Programador Responsable: Pedro Guaitas	
Descripción: Crear y desarrollar el método para graficar y visualizar un cilindro en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que el método visualizar cilindro este correctamente desarrollado.	

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-08.TI-08	Tarea de Ingeniería: TI-08 Realizar el método de codificación para la visualización de un cilindro.
Nombre: Verificar que el método visualizar cilindro este correctamente desarrollado	
Responsable: Pedro Guaitas	Fecha: 14-12-2018
Descripción: Se verificará que el método de codificación visualizar cilindro este desarrollado correctamente para graficar la figura en la pantalla del dispositivo móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar el método de visualizar cilindro. 2. Verificar que el método este desarrollado correctamente. 3. Que el método grafique correctamente el cilindro. 4. Visualizar el cilindro en la pantalla del dispositivo móvil. 	
Resultado esperado: Que la funcionalidad graficar cilindro funcione correctamente y muestre en la pantalla del dispositivo móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guaitas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU09	Nombre de la historia: Mover figuras 2D.
Modificación de historia de usuario:	
Usuario: Cliente	Sprint Asignada: 7
Prioridad en el Negocio: Alta	Puntos Estimados: 40
Riesgo en el Desarrollo: Media	Puntos Reales: 40
Descripción: Como cliente quiero que la aplicación móvil permita mover las figuras en 2D, para que los niños puedan interactuar de una mejor manera con las distintas figuras en 2D.	

Observaciones:
(Reverso) Pruebas de Aceptación
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento. <ul style="list-style-type: none"> • Que las figuras se puedan mover en 2D.

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-12.HU09	Historia de Usuario: HU09 Mover figuras 2D.
Nombre: Que las figuras se puedan mover en 2D.	
Responsable: Pedro Guaitas	Fecha: 21-12-2018
Descripción: Se verificará que las figuras en 2D se puedan mover en la pantalla del dispositivo móvil.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que todas las figuras se encuentren diseñadas. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Seleccionar una figura en 2D en la aplicación móvil. 3. Visualizar la figura 2D en la pantalla de la aplicación móvil. 4. Hacerla girar a la figura 2D. 5. Verificar que la figura 2D gire de forma uniforme. 	
Resultado esperado: Que la figura 2D gire de forma uniforme en la pantalla de los dispositivos móviles.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guaitas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU09 Mover figuras 2D.	
Número de Tarea: TI-09. HU09	Nombre de Tarea: Realizar el método de codificación que permita mover las figuras en 2D en la pantalla del dispositivo móvil.
Tipo de Tarea: Desarrollo	Puntos Estimados: 40
Fecha Inicio: 17-12-2018	Fecha Fin: 21-12-2018
Programador Responsable: Pedro Guaitas	
Descripción: Crear y desarrollar el método que permita mover figuras en 2D en la pantalla del dispositivo móvil.	

(Reverso) Pruebas de Aceptación

Verificar que el método mover figuras en 2D este correctamente desarrollado.

Realizado por: Guaitillas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-09.TI-09

Tarea de Ingeniería: TI-09 Realizar el método de codificación que permita mover las figuras en 2D en la pantalla del dispositivo móvil.

Nombre: Verificar que el método mover figuras en 2D este correctamente desarrollado.

Responsable: Pedro Guaitillas

Fecha: 21-12-2018

Descripción: Se verificará que el método de codificación mover las figuras en 2D este desarrollado correctamente para mover todas las figuras en 2D en la pantalla del dispositivo móvil.

Condiciones de Ejecución:

- Que el método este creado y desarrollado.

Pasos de ejecución:

1. Ejecutar el método de para mover las figuras en 2D.
2. Verificar que el método este desarrollado correctamente.
3. Que el método mover las figuras en 2D se ejecute correctamente.
4. Que las figuras 2D se muevan en la pantalla del dispositivo móvil.

Resultado esperado: Que la funcionalidad mover las figuras en 2D funcione correctamente y las figuras se muevan en la pantalla del dispositivo móvil.

Evaluación de la prueba: Exitosa.

Realizado por: Guaitillas, Pedro, 2019

HISTORIA DE USUARIO

Número: HU10

Nombre de la historia: Mover figuras 3D

Modificación de historia de usuario:

Usuario: Cliente

Sprint Asignada: 8

Prioridad en el Negocio: Alta

Puntos Estimados: 40

Riesgo en el Desarrollo: Media

Puntos Reales: 40

Descripción: Como cliente quiero que la aplicación móvil permita mover las figuras en 3D, para que los niños puedan interactuar de mejor manera con las distintas figuras en 3D.

Observaciones:

(Reverso) Pruebas de Aceptación

Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.

- Que las figuras se puedan mover en 3D.

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-13.HU10

Historia de Usuario: HU10 Mover figuras 3D.

Nombre: Que las figuras se puedan mover en 3D.

Responsable: Pedro Guailas

Fecha: 11-01-2019

Descripción: Se verificará que las figuras en 3D se puedan mover en la pantalla del dispositivo móvil.

Condiciones de Ejecución:

- Que todas las figuras se encuentren diseñadas.

Pasos de ejecución:

1. Abrir la aplicación móvil.
2. Seleccionar una figura en 3D en la aplicación móvil.
3. Visualizar la figura 3D en la pantalla de la aplicación móvil.
4. Hacerla girar a la figura 3D.
5. Verificar que la figura 3D gire de forma uniforme.

Resultado esperado: Que la figura 3D gire de forma uniforme en la pantalla de los dispositivos móviles.

Evaluación de la prueba: Exitosa.

Realizado por: Guailas, Pedro, 2019

TAREA DE INGENIERÍA

Historia de Usuario: HU10 Mover figuras 3D.

Número de Tarea: TI-10. HU10

Nombre de Tarea: Realizar el método de codificación que permita mover las figuras en 3D en la pantalla del dispositivo móvil.

Tipo de Tarea: Desarrollo

Puntos Estimados: 40

Fecha Inicio: 7-01-2019

Fecha Fin: 11-01-2019

Programador Responsable: Pedro Guailas

Descripción: Crear y desarrollar el método que permita mover las figuras en 3D en la pantalla del dispositivo móvil.

(Reverso) Pruebas de Aceptación

Verificar que el método mover figuras en 3D este correctamente desarrollado.

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-10.TI-10	Tarea de Ingeniería: TI-10 Realizar el método de codificación que permita mover las figuras en 3D en la pantalla del dispositivo móvil.
Nombre: Verificar que el método mover figuras en 3D este correctamente desarrollado.	
Responsable: Pedro Guailas	Fecha: 11-01-2019
Descripción: Se verificará que el método de codificación mover las figuras en 3D este desarrollado correctamente para mover todas las figuras en 3D en la pantalla del dispositivo móvil.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Ejecutar el método de para mover las figuras en 3D. 2. Verificar que el método este desarrollado correctamente. 3. Que el método mover las figuras en 3D se ejecute correctamente. 4. Que las figuras 3D se muevan en la pantalla del dispositivo móvil. 	
Resultado esperado: Que la funcionalidad mover las figuras en 3D funcione correctamente y las figuras se muevan en la pantalla del dispositivo móvil.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

HISTORIA DE USUARIO	
Número: HU11	Nombre de la historia: Identificar patrón de voz
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 9
Prioridad en el Negocio: Alta	Puntos Estimados: 40
Riesgo en el Desarrollo: Media	Puntos Reales: 40
Descripción: Como cliente quiero que la aplicación móvil permita identificar las distintas figuras por medio del patrón de voz, para que los niños puedan interactuar con las figuras mediante su voz.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento. <ul style="list-style-type: none"> • Verificar que las figuras se puedan visualizar por medio de comandos de voz. 	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-14.HU11	Historia de Usuario: HU11 Identificar patrón de voz
Nombre: Verificar que las figuras se puedan visualizar por medio de comandos de voz.	
Responsable: Pedro Guaitas	Fecha: 18-01-2019
Descripción: Se verificará que las figuras se grafiquen de forma correcta por medio de comando de voz a la hora de pronunciar su respectivo nombre.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Que todas las figuras se encuentren diseñadas. • Que todas las figuras estén asignadas el patrón de voz con su respectivo nombre. 	
Pasos de ejecución: <ol style="list-style-type: none"> 1. Abrir la aplicación móvil. 2. Tocar el asistente de voz. 3. Pronunciar el nombre de una figura en específico. 4. La figura se graficará en la pantalla del dispositivo móvil. 5. Verificar que la figura coincida con el comando de voz. 	
Resultado esperado: Que las figuras graficadas coincidan con el nombre asignados al asistente de voz.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guaitas, Pedro, 2019

TAREA DE INGENIERÍA	
Historia de Usuario: HU11 Identificar patrón de voz.	
Número de Tarea: TI-11. HU11	Nombre de Tarea: Realizar el método de codificación que permita reconocer las figuras por medio de comandos de voz.
Tipo de Tarea: Desarrollo	Puntos Estimados: 40
Fecha Inicio: 14-01-2019	Fecha Fin: 18-01-2019
Programador Responsable: Pedro Guaitas	
Descripción: Crear y desarrollar el método que permita reconocer todas las figuras diseñadas por medio de comandos de voz y que se visualicen en la pantalla del dispositivo móvil.	
(Reverso) Pruebas de Aceptación Verificar que el método Identificar patrón de voz este correctamente desarrollado.	

Realizado por: Guaitas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-11.TI-11	Tarea de Ingeniería: TI-11 Realizar el método de codificación que permita reconocer las figuras por medio de comandos de voz.

Nombre: Verificar que el método Identificar patrón de voz este correctamente desarrollado.	
Responsable: Pedro Guailas	Fecha: 18-01-2019
Descripción: Se verificará que el método de codificación reconocimiento de las figuras por medio de patrones de voz este desarrollado correctamente para mostrar las figuras en la pantalla del dispositivo móvil solo por patrones de voz.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el método este creado y desarrollado. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Ejecutar el método de reconocimiento de patrones de voz. 2. Verificar que el método este desarrollado correctamente. 3. Que el método de reconocimiento de patrones de voz se ejecute correctamente. 4. Que todas las figuras se visualicen correctamente en la pantalla del dispositivo móvil por medio de patrones de voz. 	
Resultado esperado: Que la funcionalidad de reconocimiento de patrones de voz. funcione correctamente y las figuras se muestren en la pantalla del dispositivo móvil a la hora de decir el comendo de voz de la figura.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

HISTORIA TÉCNICA	
Número: HT04	Nombre de la historia: Documentación.
Modificación de historia de usuario:	
Usuario: Desarrollador	Sprint Asignada: 10
Prioridad en el Negocio: Media	Puntos Estimados: 40
Riesgo en el Desarrollo: Baja	Puntos Reales: 40
Descripción: Como desarrollador deseo poder documentar todo el proceso de diseño y desarrollo de la aplicación móvil, para tener evidencia de lo realizado y para futuras actualizaciones.	
Observaciones:	
(Reverso) Pruebas de Aceptación	
Pruebas consensuadas entre el cliente y el desarrollador y que el código debe superar para dar como finalizada la implementación del requerimiento.	
<ul style="list-style-type: none"> • Documentar de acuerdo con la arquitectura de desarrollo. 	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN	
Código: PA-15. HT04	Historia de Usuario: HT04 Documentación.

Nombre: Documentar de acuerdo a la arquitectura de desarrollo.	
Responsable: Pedro Guailas	Fecha: 23-01-2019
Descripción: Se verificará que la documentación este acorde a la arquitectura de desarrollo del proyecto.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que el proyecto cuente con su respectiva documentación. 	
Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Revisar la documentación del proyecto. 2. Revisar la arquitectura de desarrollo escogida para el proyecto. 3. Verificar que este bien documentado el desarrollo del proyecto. 	
Resultado esperado: Que la documentación este realizada de acuerdo con la arquitectura escogida y sin faltas ortográficas.	
Evaluación de la prueba: Exitosa.	

Realizado por: Guailas, Pedro, 2019

TAREA DE INGENIERÍA

Historia de Usuario: HT04 Documentación	
Número de Tarea: TI-12. HT04	Nombre de Tarea: Realizar la documentación respectiva del proyecto
Tipo de Tarea: Documentación	Puntos Estimados: 40
Fecha Inicio: 21-01-2019	Fecha Fin: 23-01-2019
Responsable: Pedro Guailas	
Descripción: Realizar toda la documentación correspondiente al desarrollo de la aplicación móvil.	
(Reverso) Pruebas de Aceptación	
Verificar que la documentación este correcta y sin faltas de ortográficas.	

Realizado por: Guailas, Pedro, 2019

PRUEBA DE ACEPTACIÓN

Código: PA-12.TI-12	Tarea de Ingeniería: TI-12 Realizar la documentación respectiva del proyecto
Nombre: Verificar que la documentación este correcta y sin faltas de ortográficas.	
Responsable: Pedro Guailas	Fecha: 23-01-2019
Descripción: Se verificará que la documentación realizada este acorde al desarrollo del proyecto de la aplicación móvil.	
Condiciones de Ejecución:	
<ul style="list-style-type: none"> • Que la documentación esté concluida. 	

Pasos de ejecución:

1. Abrir el documento que contenga toda la documentación del proyecto.
2. Revisar que la documentación sea la correcta.
3. Revisar que no existan faltas ortográficas en el documento.

Resultado esperado: Que la documentación este realizada correctamente y no contenga faltas ortográficas.

Evaluación de la prueba: Exitosa.

Realizado por: Guailas, Pedro, 2019

Se realizaron: 4 Historias técnicas, 11 Historias de usuario, 12 tareas de Ingeniería y 27 pruebas de aceptación.