

REPÚBLICA DEL ECUADOR

INSTITUTO SUPERIOR TECNOLÓGICO TENA



**INSTITUTO SUPERIOR
TECNOLÓGICO TENA**
Tecnología, Innovación y Desarrollo

TECNOLOGÍA SUPERIOR EN

DESARROLLO DE SOFTWARE

**DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID UTILIZANDO
PALABRAS EN EL IDIOMA KICHWA QUE PERMITA EL APRENDIZAJE
DE OPERACIONES MATEMÁTICAS BÁSICAS PARA NIÑOS DE 5 A 10
AÑOS DE LA ESCUELA CENTRO EDUCATIVO CULTURAL BILINGÜE
"PAULINO GREFA" DE LA PARROQUIA MUYUNA CANTÓN TENA DE LA
PROVINCIA DE NAPO.**

Trabajo de Integración Curricular, presentado como requisito parcial para optar por el título de Tecnólogo Superior en Desarrollo de Software.

AUTOR: Fabricio José Cerda Grefa

TUTOR: Ing. Gonzalo Guanipatin.

Tena - Ecuador

2021

APROBACIÓN DEL TUTOR

INGENIERO GONZALO GUANIPATIN, DOCENTE DEL INSTITUTO SUPERIOR TECNOLÓGICO TENA.

CERTIFICA:

En calidad de Tutor del Proyecto Integrador denominado: DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID UTILIZANDO PALABRAS EN EL IDIOMA KICHWA QUE PERMITA EL APRENDIZAJE DE OPERACIONES MATEMÁTICAS BÁSICAS PARA NIÑOS DE 5 A 10 AÑOS DE LA ESCUELA CENTRO EDUCATIVO CULTURAL BILINGÜE "PAULINO GREFA" DE LA PARROQUIA MUYUNA CANTÓN TENA DE LA PROVINCIA DE NAPO, de autoría del señor **FABRICIO JOSÉ CERDA GREFA** con CC. 1500894058, estudiante de la Carrera de Tecnología Superior en Desarrollo de Software del Instituto Superior Tecnológico Tena, CERTIFICO que se ha realizado la revisión prolija del Trabajo antes citado, cumple con los requisitos de fondo y de forma que exigen los respectivos reglamentos e instituciones.

Tena, 30 de julio de 2021

Ing. Gonzalo Guanipatin
TUTOR

CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR

Tena, 06 de agosto del 2021

CERTIFICACIÓN

Los Miembros del Tribunal de Grado abajo firmantes, certificamos que el Trabajo de Titulación denominada: **DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID UTILIZANDO PALABRAS EN EL IDIOMA KICHWA QUE PERMITA EL APRENDIZAJE DE OPERACIONES MATEMÁTICAS BÁSICAS PARA NIÑOS DE 5 A 10 AÑOS DE LA ESCUELA CENTRO EDUCATIVO CULTURAL BILINGÜE "PAULINO GREFA" DE LA PARROQUIA MUYUNA CANTÓN TENA DE LA PROVINCIA DE NAPO**, estudiante de la Carrera de Desarrollo de Software del Instituto Superior Tecnológico Tena, ha sido corregida y revisada; por lo que autorizamos su presentación.

Atentamente;

Mgs. Salomón Quilumba
PRESIDENTE DEL TRIBUNAL

Lic. Gissela Solórzano
MIEMBRO DEL TRIBUNAL

Lic. Inés Gómez
MIEMBRO DEL TRIBUNAL

AUTORÍA

Yo, **FABRICIO JOSÉ CERDA GREFA** con número de cédula N° 1500894058, declaro ser autor del presente Trabajo de Titulación y absuelvo expresamente al Instituto Superior Tecnológico Tena, y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de la misma.

Adicionalmente acepto y autorizo al Instituto Superior Tecnológico Tena, la publicación de mi trabajo de Titulación en el repositorio institucional - biblioteca Virtual.

AUTOR: Fabricio Jesé Cerda Grefa

FIRMA:

CÉDULA: 150089405-8

FECHA: Tena, 30 de julio del 2021.

CARTA DE AUTORIZACIÓN DE INTEGRADOR CURRICULAR PARTE DEL AUTOR PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO

Yo, **FABRICIO JOSÉ CERDA GREFA**, declaro ser autor del trabajo de titulación denominado: **DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID UTILIZANDO PALABRAS EN EL IDIOMA KICHWA QUE PERMITA EL APRENDIZAJE DE OPERACIONES MATEMÁTICAS BÁSICAS PARA NIÑOS DE 5 A 10 AÑOS DE LA ESCUELA CENTRO EDUCATIVO CULTURAL BILINGÜE "PAULINO GREFA" DE LA PARROQUIA MUYUNA CANTÓN TENA DE LA PROVINCIA DE NAPO**, como requisito para la obtención del Título de: **TECNÓLOGO SUPERIOR EN DESARROLLO DE SOFTWARE**. Autorizo al Sistema Bibliotecario del Instituto Superior Tecnológico Tena, para que, con fines académicos, muestre al mundo la producción intelectual del Instituto, a través de la visualización de su contenido que constará en el Repositorio Digital Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio el Instituto.

El Instituto Superior Tecnológico Tena, no se responsabiliza por el plagio o copia del Trabajo de Investigación que realice un tercero. Para constancia de esta autorización, en la ciudad de Tena, 16 días del mes de abril del 2021, firma el autor.

AUTOR: Fabricio José Cerda Grefa

FIRMA:

CÉDULA: 150089405-8

DIRECCIÓN: Parroquia Muyuna - Tena

CORREO ELECTRÓNICO: cerdafabricio1989@gmail.com

CELULAR: 0985118372

DEDICATORIA

En primer lugar agradezco a Dios, por permitirme llegar a este momento tan importante en mi vida. Por los triunfos y los momentos difíciles que me han enseñado a valorarlo cada día más, A mi esposa y mis hijos quienes me han acompañado durante todo mi trayecto estudiantil para convertirme en un profesional. A mis amigos, que gracias al equipo que formamos logramos llegar hasta el final del camino y que hasta el momento seguimos siendo amigos. A mis profesores, gracias por su tiempo, por su apoyo, así como por la sabiduría que me transmitieron en el desarrollo de mi formación profesional.

Fabricio José Cerda Grefa

AGRADECIMIENTO

En primer lugar, doy infinitamente gracias a Dios, por haberme dado fuerza y valor para culminar esta etapa de mi vida.

Agradezco también, la confianza y el apoyo brindado por parte de mi esposa Gricelda, que sin duda alguna en el trayecto de mi vida me ha demostrado su amor, corrigiendo mis faltas y celebrando mi continuidad estudiantil.

A mis hijos: Nahomi y Anthony, que con sus consejos y travesuras me han ayudado a afrontar los retos que se me han presentado a lo largo de mi vida y estudio.

A mi padre, que siempre lo he sentido presente en mi vida y sé que está orgulloso de la persona en la cual me he convertido.

A mis hermanas/os: Jheny, Inés, German, Alexander y Widinson, por darme su apoyo incondicional en el transcurso de mi carrera, por compartir momentos de alegría, tristeza y demostrarme que siempre podré contar con ellos.

A mi director de tesis, Ing. Gonzalo Guanipatin por su esfuerzo y dedicación, quién con sus conocimientos, su experiencia, su paciencia y su motivación ha logrado en mí que pueda terminar mis estudios con éxito.

Finalmente, a mis amigos, porque cada uno con sus valiosas aportaciones hicieron posible este proyecto y por la gran calidad humana que me han demostrado con su amistad.

Fabricio José Cerda Grefa

ÍNDICE DE CONTENIDO

CERTIFICACIÓN DEL DIRECTOR	¡Error! Marcador no definido.
CERTIFICACIÓN DEL TRIBUNAL CALIFICADOR	iii
AUTORÍA.....	iv
CARTA DE AUTORIZACIÓN DE TESIS PARTE DEL AUTOR PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE DE CONTENIDO	viii
ÍNDICE DE GRÁFICOS	xiii
ÍNDICE DE TABLAS	xiv
ÍNDICE DE IMÁGENES	xvi
A. TEMA.....	18
B. RESUMEN	19
ABSTRACT.....	20
C. FUNDAMENTACIÓN DEL TEMA	21
3.1 Introducción.....	21
3.2 Justificación del proyecto.....	22
3.3 Importancia del proyecto.....	22
3.4 Necesidad	23
3.5 Actualidad	24

3.6 Delimitación.....	24
3.6.1 Delimitación Espacial.....	24
3.6.2 Delimitación Temporal.....	24
3.6.3 Delimitación Técnica	25
3.6.4 Unidades de Observación.....	25
3.7 Presentación del problema profesional a responder	25
3.8 Beneficiarios.....	26
3.8.1 Directos	26
3.8.2 Indirectos.....	26
D. OBJETIVOS	27
4.1 Objetivo General	27
4.2 Objetivos Específicos.....	27
E. ASIGNATURAS INTEGRADORAS	28
F. FUNDAMENTACIÓN TEÓRICA.....	29
6.1 Antecedentes nacionales tomamos los siguientes:	29
6.2 Tecnología móvil.....	30
6.3 Sistemas Operativos para móviles.....	32
6.4 Android.....	33
6.4.1 Estructura o componentes de una aplicación Android	35
6.4.2 SDK de Android Studio	37

6.4.3 Google Maps en Android	37
6.4.4 SQLite	37
6.4.5 Dalvik	38
6.5 Plataformas de desarrollo	38
6.5.1 Android SDK	38
6.5.2 Arquitectura de una aplicación	39
6.5.3 Selección del entorno de desarrollo	45
6.6 Android Studio	46
6.6.1 Ventajas y desventajas de Android Studio	47
6.6.2 Herramienta Gradle en Android Studio	47
6.6.3 Ventajas de Gradle	48
6.6.4 Arquitectura de Android Studio	48
G. METODOLOGÍA	53
7.1 Metodología para desarrollo de aplicaciones móviles	53
7.2 Metodología XP (Extreme Programming)	53
7.2.1 Proceso de la metodología XP	56
7.2.1 Fases de la metodología XP	57
7.2.1 Ciclo de vida de un proyecto XP	61
7.3 Especificaciones e instalación de la tecnología de desarrollo	63
7.3.1 Instalar y configurar JDK: Kit de desarrollo de Java	63

7.3.2	Instalar y configurar Android Studio.....	65
7.3.3	Instalar y configurar el SDK de Android	74
7.3.4	Crear y Configurar un Dispositivo virtual Android o AVD.....	78
H.	RESULTADOS	83
8.1	Fase de Planeación	83
8.1.1	Historias de Usuario.....	84
8.1.2	El Plan de Entregas (Release Plan)	87
8.1.3	Plan de Iteraciones (Iteration Plan).....	88
8.1.4	Reuniones Diarias de Seguimiento (Stand – Up Meeting).....	88
8.1.5	Requerimientos funcionales de la aplicación	88
8.1.6	Requerimientos no funcionales de la aplicación	89
8.1.7	Restricciones de la aplicación	90
8.2	Fase de Diseño.....	90
8.2.1	Simplicidad en el diseño.....	91
8.2.2	Diagrama de modelado UML.....	91
8.2.3	Metáfora	97
8.2.4	Tarjetas CRC.....	98
8.3	Fase de Codificación	99
8.3.1	Propiedad Colectiva del Código.....	99
8.3.2	Integraciones Permanentes.....	114

8.4 Fase de Pruebas	114
8.4.1 Pruebas de aceptación	114
8.4.2 Pruebas unitarias.....	118
I. CONCLUSIONES	119
J. RECOMENDACIONES.....	120
K. BIBLIOGRAFÍA.....	121
K. ANEXOS.....	127
Anexo N° 1. Código de clase MainActivity.java.....	127
Anexo N° 2. Código de clase MainActivity_Nivel1.java.....	130
Anexo N° 3. Formato del documento de autorización para el uso de nombre de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa"	134
Anexo N° 4. Capturas de Pantalla en diferentes dispositivos instalados	135

ÍNDICE DE GRÁFICOS

<i>Gráfico 1.</i> Uso de sistemas operativos móviles.	32
<i>Gráfico 2.</i> Fases de la metodología XP.	59

ÍNDICE DE TABLAS

<i>Tabla 1.</i> Matriz de Asignaturas Integradoras	28
<i>Tabla 2.</i> Herramientas del entorno de desarrollo	45
<i>Tabla 3.</i> Involucrados del proyecto	78
<i>Tabla 4.</i> Funciones del aplicativo	79
<i>Tabla 5.</i> Ingreso al sistema.....	79
<i>Tabla 6.</i> Diseño del aplicativo	80
<i>Tabla 7.</i> Contenido del aplicativo	80
<i>Tabla 8.</i> Dificultad del sistema	81
<i>Tabla 9.</i> Disponibilidad del aplicativo	81
<i>Tabla 10.</i> Plan de entregas	82
<i>Tabla 11.</i> Actores de la aplicación	87
<i>Tabla 12.</i> Ingreso a la aplicación.	88
<i>Tabla 13.</i> Navegación en la aplicación.	89
<i>Tabla 14.</i> Tarjeta CRC Ingreso al sistema	92
<i>Tabla 15.</i> Tarjeta CRC Diseño del sistema	93
<i>Tabla 16.</i> Tarjeta CRC Operaciones del sistema	93
<i>Tabla 17.</i> Prueba de aceptación, funciones del aplicativo.	103
<i>Tabla 18.</i> Prueba de aceptación, ingreso al sistema.	104
<i>Tabla 19.</i> Prueba de aceptación, diseño del aplicativo.	104

<i>Tabla 20.</i> Prueba de aceptación, contenido del aplicativo.	105
<i>Tabla 21.</i> Prueba de aceptación, dificultad del aplicativo.	105
<i>Tabla 22.</i> Prueba de aceptación, disponibilidad del aplicativo.	106

ÍNDICE DE IMÁGENES

<i>Imagen 1.</i> Logotipo Android.	33
<i>Imagen 2.</i> Pila de software de Android	48
<i>Imagen 3.</i> Flujo de características de la Programación Extrema.	55
<i>Imagen 4.</i> Sitio Web de Oracle para descargar JDK.	63
<i>Imagen 5.</i> Captura de pantalla de la instalación.	63
<i>Imagen 6.</i> Sitio Web Oficial de Android Studio.	64
<i>Imagen 7.</i> Captura de pantalla de proceso de instalación	65
<i>Imagen 8.</i> Captura de pantalla de proceso de instalación	66
<i>Imagen 9.</i> Captura de pantalla de proceso de instalación	66
<i>Imagen 10.</i> Captura de pantalla de proceso de instalación	67
<i>Imagen 11.</i> Captura de pantalla de proceso de instalación	67
<i>Imagen 12.</i> Captura de pantalla de proceso de instalación	68
<i>Imagen 13.</i> Captura de pantalla de proceso de instalación	69
<i>Imagen 14.</i> Captura de pantalla de proceso de instalación	69
<i>Imagen 15.</i> Captura de pantalla de proceso de instalación	70
<i>Imagen 16.</i> Captura de pantalla de proceso de instalación	71
<i>Imagen 17.</i> Captura de pantalla de proceso de instalación	72
<i>Imagen 18.</i> Captura de pantalla de proceso de instalación	73
<i>Imagen 19.</i> Captura de pantalla de proceso de instalación	74

<i>Imagen 20.</i> Captura de pantalla de proceso de instalación	75
<i>Imagen 21.</i> Captura de pantalla de proceso de instalación	76
<i>Imagen 22.</i> Captura de pantalla de proceso de instalación	76
<i>Imagen 23.</i> Captura de pantalla de proceso de instalación	77
<i>Imagen 24.</i> Captura de pantalla de proceso de instalación	77
<i>Imagen 25.</i> Diagrama arquitectónico de la aplicación	86
<i>Imagen 26.</i> Caso de uso general de la aplicación.....	87
<i>Imagen 27.</i> Ingreso a la aplicación.	88
<i>Imagen 28.</i> Visualización del contenido.	89
<i>Imagen 29.</i> Diagrama de relación entre las tablas.....	90
<i>Imagen 30.</i> Diagrama de secuencia, ingresar al sistema.	91
<i>Imagen 31.</i> Diagrama de secuencia, procesos.	91
<i>Imagen 32.</i> Celular Samsung A31, Android 10.	124

A. TEMA

DESARROLLO DE UNA APLICACIÓN MÓVIL ANDROID UTILIZANDO PALABRAS EN EL IDIOMA KICHWA QUE PERMITA EL APRENDIZAJE DE OPERACIONES MATEMÁTICAS BÁSICAS PARA NIÑOS DE 5 A 10 AÑOS DE LA ESCUELA CENTRO EDUCATIVO CULTURAL BILINGÜE "PAULINO GREFA" DE LA PARROQUIA MUYUNA CANTÓN TENA DE LA PROVINCIA DE NAPO.

B. RESUMEN

En la actualidad en la Escuela “Paulino Grefa” de la parroquia Muyuna del cantón Tena de la provincia de Napo, no existe ninguna aplicación móvil que se utilice como una herramienta didáctica, atractiva para niños de la edad de un promedio de 5 a 10 años que ayude en el aprendizaje de las operaciones básicas y a identificar los números del 1 al 10 en el idioma Kichwa.

La metodología XP (Extreme Programming) fue con la cual de desarrollo este proyecto en la cual encontramos cuatro fases: planeación, diseño, desarrollo y pruebas además se trabajó con herramientas tecnológicas como Android Studio, y framework de desarrollo multiplataforma. Asimismo, como resultado se obtuvo un aplicativo móvil que ayudo a la enseñanza y aprendizaje de las operaciones matemáticas, incrementar el interés en niños de la edad de 5 a 10 años con una aplicación adaptada para su fácil comprensión.

Palabras claves: Aplicación Móvil Android, Enseñanza, Operaciones matemáticas básicas.

ABSTRACT

At present, in the “Paulino Grefa” School of the Muyuna Community of Tena Canton in Napo Province there was no mobile application that is used as a didactic tool, attractive for children of the average age of 5 to 10 years who help them learn basic operations and identify numbers from 1 to 10 in the Kichwa language. The XP (Extreme Programming) methodology was used to develop this project in which we found four phases: planning, design, development and testing, as well as working with technological tools such as Android Studio, and a multiplatform development framework, and as a result, a Mobile application that helped the teaching and learning of mathematical operations, increase the interest in children from the age of 5 to 10 years with an application adapted for easy understanding.

Keywords: Android Mobile Application, Teaching, Basic mathematical operations.

Reviewed by

BEd. Gissela Solórzano Intriago
ID. 1313303941
English Teacher of Tena Institute

C. FUNDAMENTACIÓN DEL TEMA

3.1 Introducción

En la actualidad el desarrollo de las aplicaciones móviles está en auge, debido a que el dispositivo móvil llamase a estos teléfonos inteligentes (smartphones) o tabletas (tablets) han aumentado sus prestaciones hasta convertirse el día de hoy en dispositivos que pueden suplantar a un computador fácilmente. Por tal razón, el presente proyecto tiene como propósito el desarrollo de una aplicación móvil para teléfonos inteligentes con sistema operativo Android aprovechando las características que estos dispositivos prestan actualmente. La aplicación se centra en el desarrollo de una herramienta que ayude a la enseñanza y aprendizaje de las operaciones matemáticas básicas en el idioma Kichwa. Esta aplicación aprovechará los beneficios del sistema operativo Android, el cual, por ser un sistema operativo libre y tener una alta gama de dispositivos dentro del mercado se convierte en la mejor opción para desarrollar la aplicación.

Se escogió la metodología ágil XP para el desarrollo de la aplicación llegando a la conclusión que fue la más adecuada para el desarrollo de la misma. La herramienta de desarrollo que por ser la que tiene mayor documentación se eligió a Android Studio y como herramienta de gestor de Base de Datos a SQLite, teniendo así un entorno de desarrollo libre.

En el proyecto se detalla la problemática que presenta la escuela, se presenta además la justificación del desarrollo del presente proyecto, como también, los objetivos a cumplir. Se muestra los antecedentes investigativos encontrados previo a la realización del proyecto y que sustentan la investigación. Se describe el uso de la herramienta de desarrollo de la aplicación como también la metodología seleccionada para este proyecto conjuntamente con sus respectivas fases. De manera siguiente obtener las conclusiones y recomendaciones las cuales se realizan haciendo una síntesis y análisis de todas las etapas del proyecto de titulación. Se implementó en dispositivos móviles de los padres de los estudiantes y en su totalidad y de manera efectiva se validó el impacto pedagógico

mediante el análisis estadístico y descriptivo de los resultados para mejorar del proceso enseñanza – aprendizaje de la suma, resta multiplicación de los estudiantes Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa".

3.2 Justificación del proyecto

Actualmente la sociedad tiene al alcance de su mano la facilidad de tener aplicaciones sobre todo gratis en sus dispositivos móviles. Existen aplicaciones de todo tipo para los smartphones y tablets, apps temáticas de arte, entretenimiento, educación, ciencia, redes sociales, juegos, entre otros. La actual generación está tan acostumbrada a siempre resolver sus dudas, inquietudes y problemas por medio del internet, ingresando rápidamente desde su dispositivo móvil, y también, no está demás mencionar la ventaja más grande, ingresando desde cualquier parte del mundo. (Vega, 2015)

En la actualidad en la Escuela “Paulino Grefa” de la parroquia Muyuna, del cantón Tena, de la provincia de Napo, no existe ninguna aplicación móvil que se utilice como una herramienta didáctica atractiva para niños de la edad de un promedio de 5 a 10 años que ayude el aprendizaje de las operaciones básicas y a identificar los números del 1 al 10 en el idioma Kichwa.

La aplicación móvil prácticamente es un juego que se utilizará como estrategia didáctica y como actividad lúdica en el desarrollo integral del niño permitiendo el aprendizaje de las matemáticas básicas acorde a su edad. El juego se trata de una actividad libre que el niño siempre practica por diversión, pero cuando lo hace por aprender, puede llegar a crear relaciones especiales de interrelación y comunicación para desarrollar su formación integral de manera crítica, analítica y creativa.

3.3 Importancia del proyecto

En la ciudad de Tena, provincia de Napo, parroquia Muyuna, luego de haber realizado un previo análisis y acercamiento hacia la Escuela Centro Educativo Cultural

Bilingüe "Paulino Grefa", se confirma que existe la ausencia de aplicaciones inteligentes que ayuden a la educación y que incentive a los alumnos a practicar el idioma Kichwa.

Otra de las causas, es el desconocimiento del uso de la tecnología móvil como herramientas educativas las cuales al ser más interactivas ayudarán a los alumnos a captar más su atención e interés en dichos temas educativos.

Finalmente, al estudiante con los conocimientos en la Lengua Kichwa ayudará a comprender las operaciones matemáticas básicas mediante una app de uso fácil para niños de 5 a 10 años.

3.4 Necesidad

Actualmente, los usos de dispositivos móviles integrados junto con sus respectivas Apps se han vuelto indispensables en la vida cotidiana, ya que contienen nuevas y novedosas características y funciones que han convertido al dispositivo móvil en una computadora de mano. Los educadores hacen uso del celular como herramienta de comunicación y como un medio de información, pero también, como organizadores personales y para descargar información de internet. Si bien es cierto, los estudiantes estaban obligados a “desconectarse” de sus dispositivos móviles al ingresar a las instituciones educativas, las cuales cada vez más serán percibidas como irrelevantes, aburridas y alejadas de este mundo regido por la tecnología en la que los jóvenes conviven el día a día (Navarra, 2018).

En Irlanda se está llevando a cabo un proyecto de aprendizaje de la lengua original irlandesa a través de la telefonía móvil. El objetivo de la iniciativa era promover el interés de los alumnos por la lengua irlandesa, desarrollar sus habilidades comunicativas e incrementar el uso de las cuatro competencias básicas lectura, escritura, habla y escucha en dicha lengua (Aparice, 2010).

En Ecuador existen diversas recomendaciones que pueden servir para ayudar a sus hijos. Estas van desde temas de salud hasta orientación y aplicaciones relacionadas a la educación como *ABC for Kids*: es una forma divertida para que los niños en edad pre-

escolar aprendan los nombres de las letras del alfabeto y las reconozcan, por lo cual pueden practicar buscando las letras en un contexto (El Telegrafo, 2019).

El aprendizaje móvil promete apoyar la enseñanza virtual por medio de nuevos dispositivos portátiles que fomenten nuevos sistemas de comunicación entre el hombre y la tecnología. Especialmente para los alumnos, las herramientas de aprendizaje móvil resultan más atractivas ya que son novedades tecnológicas que actualmente utilizan como los Smartphone o Tablet. Además, es necesario motivar al estudiante que mejore los conocimientos del lenguaje Kichwa siendo una lengua ancestral y mediante al software adaptado en un dispositivo móvil se lograra el aprendizaje con nuevos métodos tecnológicos.

3.5 Actualidad

Mediante un acercamiento por parte del investigador a la Unidad Educativa Cultural Bilingüe "Paulino Grefa" en la parroquia Muyuna ubicada en el Cantón Tena provincia de Napo, se presenció que la mayoría de alumnos entre 5 a 10 años no aplican el idioma Kichwa como escrita y verbal, por ello, mediante el desarrollo de la aplicación ayudará a mantener valorando el idioma ancestral y por ende será una herramienta más didáctica e interactiva para aprender las cuatro operaciones matemáticas.

3.6 Delimitación

3.6.1 Delimitación Espacial

El Trabajo de Integración Curricular se lo realizará a nivel general en la Unidad Educativa Cultural Bilingüe "Paulino Grefa" del Cantón Tena provincia del Napo.

3.6.2 Delimitación Temporal

El proyecto se lo efectuará en el Periodo Académico Noviembre 2020 – Abril 2021.

3.6.3 Delimitación Técnica

La app para el aprendizaje del estudiante, estará enfocado a la realización de:

- Rendimiento
- Comprensión
- Tiempo

3.6.4 Unidades de Observación

Las unidades de observación que se contemplan para este trabajo están enfocadas en:

- Docentes de la Unidad Educativa Cultural Bilingüe "Paulino Grefa".
- Estudiantes de la Unidad Educativa Cultural Bilingüe "Paulino Grefa".

3.7 Presentación del problema profesional a responder.

En la Unidad Educativa Cultural Bilingüe "Paulino Grefa", según lo mencionado por algunos actores de la misma, los estudiantes tienen dificultades para captar de manera inmediata el aprendizaje en el área Kichwa tanto oral y escrita, es por ello que se realizó la propuesta para el diseño e implementación de una aplicación con la que se pretende mejorar los conocimientos en el aprendizaje.

Campo: Tecnologías de la Información y Comunicación

Área: Informática

Aspecto: Elaboración de una app

Sector: Programación

Línea de investigación: Desarrollo de Software.

3.8 Beneficiarios

3.8.1 Directos

Los beneficiarios directos del trabajo Integrador Curricular son:

- Docentes de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo.

3.8.2 Indirectos

Los beneficiarios indirectos del trabajo de titulación son:

- Estudiantes de la Escuela centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo.

D. OBJETIVOS

4.1 Objetivo General

- Desarrollar un aplicativo para dispositivos móviles Android utilizando palabras en el idioma Kichwa que facilite el proceso de enseñanza-aprendizaje de operaciones matemáticas básicas enfocada a niños de 5 a 10 años de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo.

4.2 Objetivos Específicos

- Aplicar metodologías de desarrollo de software seguro para el desarrollo del aplicativo móvil Android.
- Diseñar el aplicativo que facilite el aprendizaje de las operaciones matemáticas básicas para niños de 5 a 10 años de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo.
- Implementar la aplicación en dispositivos móviles (Tablets, celulares) que posean sistema operativo Android.

E. ASIGNATURAS INTEGRADORAS

Para el desarrollo del Trabajo de Integración Curricular se ha considerado apoyarse en la siguiente metodología de estudios que contempla la malla curricular de la carrera de Desarrollo de Software.

Tabla 1.

Matriz de Asignaturas Integradoras

ASIGNATURAS DE LA MALLA CURRICULAR ASOCIADAS AL PROYECTO INTEGRADOR CURRICULAR			
Asignatura	Aplicación		Resultados de Aprendizaje
	Directa	Indirecta	
Base de datos	X		Aplica alternativas de solución a problemas identificados
Programación de aplicaciones Android	X		Crea aplicación en lenguaje nativo del sistema operativo.
Análisis y Diseño de Sistemas	X		Aplica alternativas de solución a problemas identificados.
Metodología de desarrollo de software	X		Aplica una metodología de desarrollo de software durante el ciclo de vida de una aplicación desarrollada.

Legislación informática	X	Desarrollar sus actividades con respeto de las normas jurídicas vigentes en la legislación ecuatoriana e Internacional.
-------------------------	---	---

Elaborado por: Cerda Fabricio, 2021.

F. FUNDAMENTACIÓN TEÓRICA

Este literal nos ayudara a clarificar y comprender algunos conceptos teóricos y prácticos, con el objetivo de investigar y recolectar información, que pudiera documentar y crear referentes sobre el trabajo que se propone hacer, se han encontrado diversas fuentes bibliográficas que reposan en bibliotecas, es así como también se revisaron fuentes de información en bibliotecas, revistas virtuales, artículos científicos y académicos que tienen relación directa con el tema a desarrollarse.

6.1 Antecedentes nacionales tomamos los siguientes:

A nivel mundial ya se evidencia que utilizan muy a menudo las Aplicaciones Móviles y se hallan en tiendas virtuales, páginas web, redes sociales entre otras, en querencia de estas tecnologías se realizan publicidades, para adquirir más demandantes y dejando un enlace para su apropiada descarga de la aplicación para que los usuarios la puedan instalar en su Smartphone (González M., 2014).

Esta iniciativa ha sido de gran éxito debido a que tiene una respuesta rápida y permite suministrar eficazmente la dirección web. Además, se vuelve interactivo para el usuario quien lo utilice, de esa forma facilitando el acceso a recursos requeridos de una manera fácil, oportuna y sin muchos medios, para la obtención de informaciones pertinentes (Campos, 2014).

En el Ecuador las Aplicaciones Móviles apuntan a ser poderosas en el medio de información de manera eficaz en los procesos de virtualización. Es así que ya algunas páginas web han optado por implementar las Aplicaciones Móviles para la vista previa del sitio (El Comercio, 2018).

Con el objetivo de buscar información, que pudiera documentar y crear referentes sobre el trabajo que se propone hacer, se han encontrado diversas fuentes bibliográficas que reposan en bibliotecas, así como también se analizaron fuentes de información en revistas digitales, bibliotecas virtuales, artículos científicos y académicos que tienen relación directa con el tema de investigación:

Como un ejemplo claro de su importancia en Guatemala, marzo del 2015; La App “Duolingo” ha sido usada en escuelas públicas como un proyecto piloto "Duolingo Challenge", llevado a cabo por el gobierno, y el cual intenta que los estudiantes de los novenos grados aprendan inglés usando los recursos que nos brinda esta aplicación móvil (Fernando, 2016).

También se puede destacar el aporte de la App Duolingo que en enero de 2015 hizo el lanzamiento del proyecto llamado "Duolingo para Escuelas", cuyo fin es el de proveer a los docentes de un módulo de control a través de su portal con el cual se puede hacer un seguimiento del progreso de sus alumnos.

6.2 Tecnología móvil

La tecnología móvil se vincula directamente a la comunicación o telefonía móvil y es a la que nos vamos a referir para nuestro estudio.

Dentro de las tecnologías móviles esta las comunicaciones móviles que sin duda alguna han experimentado un enorme crecimiento desarrollándose diversas tecnologías y sistemas para dar servicios de comunicación inalámbrica. “En Ecuador el servicio móvil celular logra incluirse a finales de 1993 con la entrada en el mercado de CONECEL S.A. (Porta Celular, luego CLARO) y OTECEL S.A. (al inicio Bellsouth y actualmente denominada Movistar), manteniéndose el dominio de estas 2 empresas hasta el año 2003 cuando entró en operación una tercera operadora TELECSA (al inicio Alegro actualmente CNT E.P.)”. (Super intendencia de Telecomunicaciones, 2012)

En el pasado las empresas de telecomunicaciones brindaban un solo servicio: telefonía, audio y video por suscripción, servicios portadores y servicios de valor agregado. “En la actualidad los servicios han convergido de tal manera que un mismo proveedor de servicios dentro de una misma infraestructura de telecomunicaciones, puede brindar múltiples servicios. Esta convergencia tecnológica de un dispositivo electrónico que proporciona funcionalidades de telefonía y ofrece asistencias iguales a las de una computadora, hace posible el concepto de Smartphone (teléfonos inteligentes) con los que se puede realizar compras, enviar y recibir correos

electrónicos, escuchar música, ver videos, acceder a redes sociales, además de hablar por teléfono, es posible con sólo un dispositivo. La gran aceptación del mercado hacia estos aparatos ha permitido que las empresas líderes en tecnología vean atractiva esta plaza, direccionando su trabajo al desarrollo de aplicaciones personales y empresariales que satisfagan al usuario”. (Super intendencia de Telecomunicaciones, 2012)

Definitivamente la posibilidad de acceso a internet (internet móvil) es el factor que más ha incidido para que los Smartphone logren tener el nivel de penetración alcanzado en el mercado, la sensación de conectividad que ofrecen las redes sociales, así como el acceso a mensajería instantánea hace que los usuarios se vean tentados por estos dispositivos. Por otra parte, las numerosas aplicaciones disponibles para teléfonos inteligentes le dan un valor agregado para quienes buscan entretenimiento en dispositivos móviles.

Todas estas aplicaciones se ejecutan dentro de un eco sistema existiendo varios factores que lo afectan como son: la infraestructura de la aplicación, el sistema operativo, los métodos de entrada de información, los propios usuarios, los canales de distribución de la aplicación, etc. “En el caso de las aplicaciones móviles, el ecosistema es aún más heterogéneo que en el resto de desarrollos. Pueden ejecutarse en diferentes tipos de dispositivo, ya sea en un móvil antiguo o bien en un actualizado ya sea un Smartphone o una Tablet los cuales se encuentran diseñados bajo una plataforma informática y

dependiendo de la marca, en diferentes sistemas operativos, por lo tanto, para conseguir un desarrollo satisfactorio es ideal conocer los más profundamente”. (R. Vique, 2012)

6.3 Sistemas Operativos para móviles

Se detalla algunos conceptos tomados de un sistema operativo móvil:

Un sistema operativo (SO) móvil controla un dispositivo móvil al igual que los PC utilizan Windows o Linux entre otros. Sin embargo, los sistemas operativos móviles son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos. Algunos de los sistemas operativos utilizados en los dispositivos móviles están basados en el modelo de capas. (Bustamante, 2016)

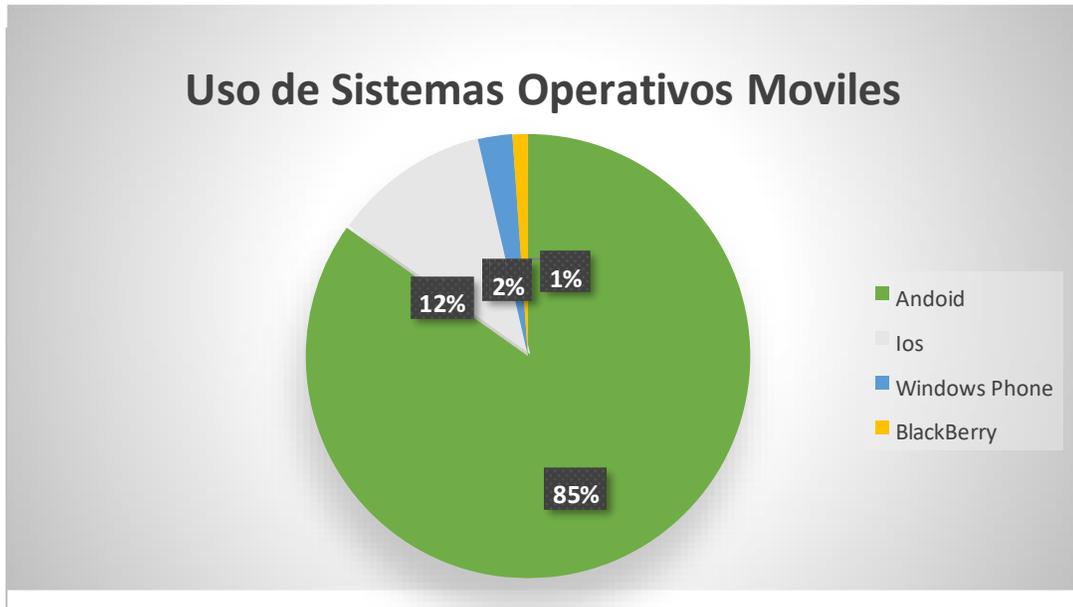
De igual manera Juan José Bustamante B (2016) dice: “El Sistema Operativo (SO) móvil de un teléfono o tableta realiza la interacción real con lo que podemos hacer a partir de las capacidades del hardware que conforman un equipo. A manera de traductor, esta plataforma interpreta lo que el usuario quiere que la terminal realice y cada vez, lo ejecuta con mayor inteligencia. Una de las cualidades más atractivas de un sistema operativo móvil es la rapidez con la que en general se desempeña”.

Según el servicio de estadísticas NetMarketShare, la cuota de mercado de sistemas operativos móviles a principios de 2019 es el siguiente:

- Android 84,71 % (en países como España las diferencias son más significativas, donde Android tiene más del 90,8% de la cuota de mercado (Acuña, 2019))
- iOS 11,55 %
- Windows Phone 2,51 %
- BlackBerry OS 1,22 %. (Acuña, 2019)

Gráfico 2.

Uso de sistemas operativos móviles.



Elaborado propio por: Cerda Fabricio, 2021.

Android tiene la mayor cuota, desde enero 2011, con más de la mitad del mercado, experimentó un creciente aumento y en solo diez años (2009 a comienzos de 2019) ha pasado a ser el SO móvil más utilizado. (Zona, 2019)

6.4 Android

Android es una solución completa de software de código libre (GNU Linux7) para teléfonos y dispositivos móviles. Es un paquete que engloban un sistema operativo, un "Runtime" de ejecución basado en Java, un conjunto de librerías de bajo y medio nivel y un conjunto inicial de aplicaciones destinadas al usuario final. Android se distribuye bajo una licencia libre que permite la integración con soluciones de código propietario. Android surge como resultado de la "Open Handset Alliance" un consorcio de 48 empresas distribuidas por todo el mundo con intereses diversos en la telefonía móvil y un compromiso de comercializar dispositivos móviles con este sistema

operativo. El desarrollo viene avalado principalmente por Google (tras la compra de Android Inc. en 2005). (Universidad Politécnica de Madrid, 2012)



*Imagen 1. Logotipo Android.
Fuente: (Android Studio, 2021)*

Como se había descrito anteriormente, Android está basado en Linux, disponiendo de un Kernel en este sistema y utilizando una máquina virtual sobre este Kernel que es la responsable de convertir el código escrito en Java de las aplicaciones a código capaz de comprender el Kernel. Las aplicaciones para Android se escriben y desarrollan en Java aunque con unas APIS (Application Programming Interfaces) propias por lo que programas realizados en Java para PC y demás plataformas no son compatibles con Android. (Tecnología Fácil, 2018)

La cualidad más grande de este sistema operativo es que es de código abierto y se distribuye bajo dos tipos de licencias, una que abarca todo el código del Kernel y que es GNU GPLv2, Google también por supuesto tiene otra licencia para el resto de componentes del sistema que se licencia bajo APACHE v2. (Bustamante, 2016)

Vamos a detallar sus principales características:

- Alta calidad de gráficos y sonido: gran variedad de formatos soportados
- Las aplicaciones escritas en Java pueden ser compiladas y ejecutadas en la máquina virtual Dalvik, máquina virtual diseñada para uso en dispositivos móviles.

- Soporte para hardware adicional (cámaras de vídeo, pantallas táctiles, GPS, acelerómetros, etc.).
- Entorno de desarrollo (emulador, herramientas de depuración, perfiles de memoria y funcionamiento, plugin para Android Studio).
- Plataforma realmente abierta al ser basada en Linux y de código libre. Se puede usar y adecuar el sistema sin pagar algún tipo de licencia.
- Portabilidad asegurada: Al desarrollar las aplicaciones en Java, y gracias al concepto de máquina virtual, las aplicaciones podrán ser ejecutadas en gran variedad de dispositivos tanto actuales como futuros.
- Las interfaces se hacen en formato xml, lo que permite el uso de una misma interfaz en dispositivos de distintos tamaños de pantallas.
- Gran cantidad de servicios incorporados: Reconocimiento y síntesis de voz, localización basada en GPS, potentes bases de datos, etc.
- Nivel de seguridad: Los programas se encuentran separados unos de otros. Cada aplicación dispone distintos tipos de permisos que limitan su rango de actuación.

6.4.1 Estructura o componentes de una aplicación Android

Las aplicaciones en Android están basadas en componentes y son los elementos básicos con los que se construyen un proyecto. Una aplicación Android será una combinación de uno o más de estos componentes y se deben declarar en el fichero `AndroidManifest.xml` donde se definen todos los componentes de la aplicación, así como los permisos que requiere, o los recursos y librerías que utiliza. (Vico, 2011)

Activity: Son el componente visual de una aplicación, es decir forman la interfaz y se podría decir que son las pantallas o ventanas que ve el usuario.

Service: Son componentes sin interfaz gráfica, se ejecutan en segundo plano y realizan múltiples acciones.

Content Provider: Es la forma que tiene Android de compartir datos entre aplicaciones, compartiendo datos sin la necesidad de dar de detalles sobre su almacenamiento.

Broadcast Receiver: Este componente se encarga de detectar y reaccionar a los eventos generales como pueden ser: batería baja, entrada de llamada, es decir es el encargado de recibir las llamadas que son enviados a cualquier aplicación que esté “escuchando” (Vico, 2011).

También debemos tener muy presente el manejo de las actividades y fragmentos dentro de la construcción de una aplicación móvil.

Actividad: Una actividad puede contener 0 o múltiples números de fragmentos en función del tamaño de la pantalla. Un fragmento puede ser reutilizado en múltiples actividades, por lo que actúa como un componente reutilizable en las actividades.

La clase Activity es un componente clave de una app para Android, y la forma en que se inician y se crean las actividades es una parte fundamental del modelo de aplicación de la plataforma. A diferencia de los paradigmas de programación en los que las apps se inician con un método main (), el sistema Android inicia el código en una instancia de Activity invocando métodos de devolución de llamada específicos que corresponden a etapas específicas de su ciclo de vida (Android Studio, 2020) .

Fragmento: Un Fragment representa un comportamiento o una parte de la interfaz de usuario en una FragmentActivity. Se puede combinar varios fragmentos en una sola actividad para crear una IU (superposición de software desarrollada por Samsung Electronics para sus dispositivos Android) multi panel y volver a usar un fragmento en diferentes actividades. Su puede pensar en un fragmento como una sección modular de una actividad que tiene un ciclo de vida propio, que recibe sus propios eventos de entrada y que se podría agregar o quitar mientras la actividad se esté ejecutando (algo así como una "subactividad" que se puede volver a usar en diferentes actividades) (Android Studio, 2020).

6.4.2 SDK de Android Studio

Como se observó anteriormente, Android es una plataforma de software libre, por lo que cuenta con Kit de desarrollo de Software o SDK disponible para todo desarrollador o para alguien que desee experimentar que incluye: el conjunto completo del API soportado por este sistema para crear, probar y depurar aplicaciones para Android. (Baña, 2016)

6.4.3 Google Maps en Android

Google Maps nos proporciona un servicio de cartografía online que podremos utilizar en nuestras aplicaciones Android. A diferencia de Android, Google Maps no es un software libre, por lo que está limitado a una serie de condiciones de servicio. Podemos usarlo de forma gratuita siempre que nuestra aplicación no solicite más de 15.000 codificaciones geográficas al día. Podemos incluir propaganda en los mapas. (Universidad Carlos III de Madrid, 2012)

6.4.4 SQLite

SQLite es un sistema gestor de base de datos relacional (RDBMS18). Lo que hace único a SQLite es que se considera una solución embebida. La mayoría de los sistemas de gestión de bases de datos como Oracle, MySQL, y SQL Server son procesos de servidor autónomos que se ejecutan independientemente. SQLite es en realidad una librería que está enlazada dentro de las aplicaciones. Todas las operaciones de base de datos se manejan dentro de la aplicación mediante llamadas y funciones contenidas en la librería SQLite, hace mucho más fácil tratar grandes bases de datos en comparación con otras soluciones más convencionales de base de datos. (Muradas M, 2018)

SQLite se basa en el Lenguaje Estructurado de Consultas (SQL); el mismo lenguaje que utilizan la mayoría de RDBMS. Es una gran ventaja si se conoce SQL ya que la mayoría de las sentencias se formulan de la misma forma. Android tiene integrado en el propio sistema una API completa que nos permite manejar la Base de Datos en SQLite.

6.4.5 Dalvik

Dalvik es el nombre de la máquina virtual que utiliza (DalvikVM), la cual está basada en registro, diseñada y escrita por "Dan Bornstein" y algunos otros profesionales de Google. En ella podemos encontrar una gran diferencia con respecto a la máquina virtual Java (JVM). Dalvik VM es un intérprete que sólo ejecuta archivos con formato dex (Dalvik executable). Este formato está optimizado para el almacenamiento eficiente de la memoria, lo cual consigue delegando en el kernel la gestión de hilos (multithreading), de memoria y de procesos. (Universidad Politécnica de Madrid, 2012)

6.5 Plataformas de desarrollo

Para el desarrollo de aplicaciones Android, hay varias opciones disponibles ya sea herramientas de desarrollo oficial y herramientas de terceros, pero generalmente se utiliza el software oficial "Android Studio" que utiliza un lenguaje de programación Java y el conjunto de herramientas de desarrollo SDK (Software Development Kit) que es una herramienta oficial, y la cual utilizaremos en nuestro proyecto para la parte de la aplicación.

6.5.1 Android SDK

SDK responde a las siglas Software Development Kit, es decir un kit de desarrollo de software. Con él podremos desarrollar aplicaciones y ejecutar un emulador del sistema Android de la versión que sea. Todas las aplicaciones Android se desarrollan en lenguaje Java con este kit. (NextPit, 2012)

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad. (NextPit, 2012)

6.5.2 Arquitectura de una aplicación

Como ya se ha mencionado, Android es una plataforma para dispositivos móviles que contiene una pila de software donde se incluye un sistema operativo, middleware y aplicaciones básicas para el usuario que a continuación indicaremos:

Aplicaciones: Este nivel contiene, tanto las incluidas por defecto de Android como aquellas que el usuario vaya añadiendo posteriormente, ya sean de terceras empresas o de su propio desarrollo. Todas estas aplicaciones utilizan los servicios, las API y librerías de los niveles anteriores.

Framework de Aplicaciones: Representa fundamentalmente el conjunto de herramientas de desarrollo de cualquier aplicación. Toda aplicación que se desarrolle para Android, ya sean las propias del dispositivo, las desarrolladas por Google o terceras compañías, o incluso las que el propio usuario cree, utilizan el mismo conjunto de API y el mismo "framework", representado por este nivel

Entre las API más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **Activity Manager:** Conjunto de API que gestiona el ciclo de vida de las aplicaciones en Android.
- **Window Manager:** Gestiona las ventanas de las aplicaciones y utiliza la librería Surface Manager.
- **Telephone Manager:** Incluye todas las API vinculadas a las funcionalidades propias del teléfono (llamadas, mensajes, etc.)
- **Content Provider:** Permite a cualquier aplicación compartir sus datos con las demás aplicaciones de Android. Por ejemplo, gracias a esta API la información de contactos, agenda, mensajes, etc. será accesible para otras aplicaciones.
- **View System:** Proporciona un gran número de elementos para poder construir interfaces de usuario (GUI), como listas, mosaicos, botones, "check-boxes", tamaño de ventanas, control de las interfaces mediante teclado, etc. Incluye también algunas vistas estándar para las funcionalidades más frecuentes.

- **Location Manager:** Posibilita a las aplicaciones la obtención de información de localización y posicionamiento.
- **Notification Manager:** Mediante el cual las aplicaciones, usando un mismo formato, comunican al usuario eventos que ocurran durante su ejecución: una llamada entrante, un mensaje recibido, conexión Wi-Fi disponible, ubicación en un punto determinado, etc. Si llevan asociada alguna acción, en Android denominada **Intent**, (por ejemplo, atender una llamada recibida) ésta se activa mediante un simple clic.
- **XMPP Service:** Colección de API para utilizar este protocolo de intercambio de mensajes basado en XML.

Librerías: La siguiente capa se corresponde con las librerías utilizadas por Android. Éstas han sido escritas utilizando C/C++ y proporcionan a Android la mayor parte de sus capacidades más características. Junto al núcleo basado en Linux, estas librerías constituyen el corazón de Android.

Entre las librerías más importantes ubicadas aquí, se pueden encontrar las siguientes:

- **Librería libc:** Incluye todas las cabeceras y funciones según el estándar del lenguaje C. Todas las demás librerías se definen en este lenguaje.
- **Librería Surface Manager:** Es la encargada de componer los diferentes elementos de navegación de pantalla. Gestiona también las ventanas pertenecientes a las distintas aplicaciones activas en cada momento.
- **OpenGL/SL y SGL:** Representan las librerías gráficas y, por tanto, sustentan la capacidad gráfica de Android. OpenGL/SL maneja gráficos en 3D y permite utilizar, en caso de que esté disponible en el propio dispositivo móvil, el hardware encargado de proporcionar gráficos 3D. Por otro lado, SGL proporciona gráficos en 2D, por lo que será la librería más habitualmente utilizada por la mayoría de las aplicaciones. Una característica importante de la capacidad gráfica de

Android es que es posible desarrollar aplicaciones que combinen gráficos en 3D y 2D.

- **Librería Media Libraries:** Proporciona todos los códecs necesarios para el contenido multimedia soportado en Android (vídeo, audio, imágenes estáticas y animadas, etc.)
- **FreeType:** Permite trabajar de forma rápida y sencilla con distintos tipos de fuentes.
- **Librería SSL:** Posibilita la utilización de dicho protocolo para establecer comunicaciones seguras.
- **Librería SQLite:** Creación y gestión de bases de datos relacionales.
- **Librería WebKit:** Proporciona un motor para las aplicaciones de tipo navegador y forma el núcleo del actual navegador incluido por defecto en la plataforma Android.

Tiempo de ejecución de Android: Al mismo nivel que las librerías de Android se sitúa el entorno de ejecución. Éste lo constituyen las Core Libraries, que son librerías con multitud de clases Java y la máquina virtual Dalvik.

Núcleo Linux: Android utiliza el **núcleo de Linux 2.6** como una capa de abstracción para el hardware disponible en los dispositivos móviles. Esta capa contiene los drivers necesarios para que cualquier componente hardware pueda ser utilizado mediante las llamadas correspondientes. Siempre que un fabricante incluye un nuevo elemento de hardware, lo primero que se debe realizar para que pueda ser utilizado desde Android es crear las librerías de control o drivers necesarios dentro de este kernel de Linux embebido en el propio Android (Software de Comunicaciones, 2016).

La parte final del desarrollo de aplicaciones móviles se trata de publicar las Apps de forma pública en los market places de Google y de Apple.

Para ello se debe disponer de:

- **Cuenta Google Play.** Licencia para publicar Apps Android. (\$25.00/año aprox)

- **Cuenta Apple Store.** Licencia para publicar Apps iOS. (\$100.00 año aprox o \$300.00/año para Enterprise Account de esta forma se pueden publicar apps privadas sin tener que pasar por el Store). (Coditramuntana, 2019)

Las arquitecturas implementadas en el desarrollo de aplicaciones móviles, se debe a el tipo de información el cual se vaya a impartir; recordando que, a la hora de desarrollar aplicaciones móviles, hay que tener lo que la escalabilidad y todas sus ventajas. las aplicaciones móviles según la información que imparten se dividen en dos grupos, aplicaciones móviles auto contenidas y las aplicaciones con conexión a Internet.

a. Aplicaciones móviles auto contenidas

En este tipo de aplicaciones el contenido es estático, sus imágenes, su información, sus menús, casi nunca cambia o rara vez lo hace. Esto es debido a que todo su contenido se encuentra auto contenido dentro de la misma aplicación, no necesita valerse de una herramienta o un servicio para funcionar.

Este tipo de aplicaciones son del tipo nativo, debido a que este brinda una seria de herramientas y posibilidades a la hora de desarrollar la aplicación móvil de este tipo, un ejemplo de este tipo de aplicaciones seria por ejemplo una calculadora desarrollada para ya sea Android, iOS, Windows Phone, entre otros; la aplicación no tiene necesidad de cambiar los números debido a que los números son los mismos siempre, la información puede ser suministrada a través de una manual en alguna función u opción del menú, y si necesita brindar herramientas como tema puede predeterminar colores tamaños y demás cosas al respecto.

- **Ventajas**

Su desarrollo puede resultar sencillo, no hay que hacer gastos en lo que son servicios de hosting, bases de datos, entre otros.

- **Desventajas**

Si un día decide actualizar la información en dicha aplicación, debe compilar el código nuevamente y subirla a la plataforma la cual estaba siendo distribuida.

Si la información que es utilizada por la aplicación es demasiada, puede ser que la aplicación necesite cambios periódicos lo que podría volverla inviable. (Zenva Pty Ltd, 2020)

b. Aplicaciones móviles con conexión a Internet

En este tipo de aplicaciones la información está alojada en un back-end, el cual cuenta con una base de datos la cual contiene la información, que, accedida a través de un API de servicios web, entre los cuales los más usados suelen ser SOAP o REST, los cuales comunican a la base de datos con el front-end de la aplicación, que es donde se puede ver la información solicitada.

Un ejemplo claro es las aplicaciones de las redes sociales, como por ejemplo facebook y/o twitter, aplicaciones de mensajería instantánea como lo son Line, Viber, WeChat, Wasap; necesitan el servicio de Internet para poder acceder a información específica o realizar alguna acción. Dentro de este tipo de aplicaciones, se encuentran dos tipos, los cuales son con descarga de información estática y con información dinámica. (Zenva Pty Ltd, 2020)

c. Aplicaciones móviles con descarga de información estática.

Cuenta con un archivo en formato XML u otro formato que se utilice, que se modifica de forma manual cada que se requiera cambiar información.

- **Ventajas**

No hay que publicar una versión nueva cada que se requiera dar a conocer una nueva información.

- **Desventajas**

Requiere un gasto para implementar una base de datos, hosting, entre otros. Si la información modificada es errónea, entonces se tendría que corregir dicha información hasta que sea la correcta. (Zenva Pty Ltd, 2020)

d. Aplicaciones móviles con información dinámica.

Este tipo de aplicaciones se llama así, debido a que la información esta almacenada en una base de datos la cual está alojada en un servidor y que se accede a esta a través de servicios API los cuales comunican a la base con la interfaz de la aplicación.

- **Ventajas**

Los cambios se realizan de manera fácil, solo contando con conocimientos básicos. Permite editar la información en la base de datos a través de una plataforma especifica. Cualquier persona con permisos especiales, puede realizar tareas desde cualquier lugar.

- **Desventajas**

El coste suele ser mayor, debido a que hay que contratar personal para el desarrollo web. Existe un coste también a la hora de contratar alguna empresa de hosting. El desarrollo de la aplicación es más complejo (Zenva Pty Ltd, 2020).

6.5.3 Selección del entorno de desarrollo

Para el desarrollo de la aplicación que permita el aprendizaje de operaciones matemáticas básicas para niños de 5 a 10 años de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la Provincia de Napo, a la cual se ha denominado "AppKichwa", se usará herramientas oficiales para el desarrollo de aplicaciones Android, y el entorno de desarrollo integrado llamado también IDE que se detalla a continuación. *Ver Tabla 2.*

Tabla 2.

Herramientas del entorno de desarrollo.

Herramientas	Descripción	Enlace de descarga o Instalación
Android Studio	Android Studio es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar.	https://developer.android.com/studio
Java JDK	Java Development Kit, es un software que provee herramientas de Desarrollo para la creación de programas en Java. Puede instalarse en una Computadora local o en una unidad de Red	https://www.oracle.com/java/technologies/javasedownloads.html
Android SDK	Las siglas SDK se refieren a "Software Development Kit" (Kit de Desarrollo de Software). Este Kit incluye un conjunto de herramientas para facilitar la tarea a los desarrolladores: Un depurador de código Biblioteca Emulador de Android Documentación Ejemplos de código Tutoriales	https://developer.android.com/studio/

SQLITE SQLite es una <https://www.sqlite.org/download.html> herramienta de software libre, que permite almacenar información en dispositivos empotrados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular.

Elaborado propio por: Cerda Fabricio, 2021.

6.6 Android Studio

Android Studio es un IDE, una interfaz de desarrollo. En realidad, es una especie de escritorio de trabajo para un desarrollador. Allí se encuentra nuestro proyecto, las carpetas del mismo, los archivos que hay en él, y todo lo necesario para acabar creando la aplicación. Lo mejor de Android Studio es que ha sido creado por Google y fue presentado hace tan solo unos meses, por lo que no hablamos de una herramienta antigua y nada depurada, sino de un programa muy moderno que encima ha sido creado por los mismos que han creado el sistema operativo. (Android Studio, 2020)

Entre otras cosas, cuenta con algunas herramientas que nos facilitarán mucho el desarrollo de las aplicaciones, como por ejemplo el poder pre visualizar las aplicaciones en diferentes Smartphone y Tablets para saber cómo está quedando el código que nosotros estamos editando, y cómo se ve en los diferentes tipos de pantalla que existen. Este nuevo programa, además, nos ayudará a dejar nuestra aplicación lista para publicar. Aunque las aplicaciones de Android se escriben en lenguaje Java, lo cierto es que después hay que compilarlas para que quede un único archivo .apk. Este último paso es muy simple con Android Studio. Digamos, que el nuevo IDE ahora es mucho más

familiar y fácil de usar de lo que era el SDK con el que contaba Google antes. (Jiménez, 2013)

6.6.1 Ventajas y desventajas de Android Studio

Ventajas:

- Compilación rápida.
- Ejecución de la app en tiempo real gracias al emulador.
- Ejecución de la app directamente desde el móvil.
- No soporta el desarrollo para NDK, pero IntelliJ con el plugin Android sí.
- Tiene renderizado en el tiempo real, layouts y puede hacer uso de parámetros tools.
- Funciona bien (sobre todo si se usa en versiones estables).
- Contiene todo lo necesario para desarrollar cualquier IDE.
- Es capaz de asociar automáticamente carpetas y archivos con su papel en la aplicación, la creación de nuevas carpetas, borrado de archivos en valores.

Desventaja

- Los requisitos son un poco elevados (se debería tener una buena máquina para que te funcione bien el emulador). Pero esto hace que sea el mejor entorno para programar en Android, por lo que es necesario. (Baña, 2016)

6.6.2 Herramienta Gradle en Android Studio

Gradle es un sistema de compilación que reúne en uno solo las mejores prestaciones de otros sistemas de compilación. Está basado en JVM (Java Virtual Machine), lo que significa que puedes escribir tu propio script en Java, y que Android Studio lo entenderá y lo usará. Lo mejor de Gradle es que es un plugin, lo que facilita su actualización y su exportación de un proyecto a otro. Esto significa que se puede tener un propio lenguaje de programación y automatizar el proceso de compilación en un solo paquete (de la misma manera que un jar en caso de Java) y poder distribuirlo al resto del mundo.

Google ha creado uno de los sistemas de compilación más avanzados del mercado para permitir a todos los usuarios escribir sus propios scripts sin necesidad de aprender ningún nuevo lenguaje, disminuyendo así la curva de aprendizaje y permitiendo llegar a un mayor público la programación en Android. (Baña, 2016)

6.6.3 Ventajas de Gradle

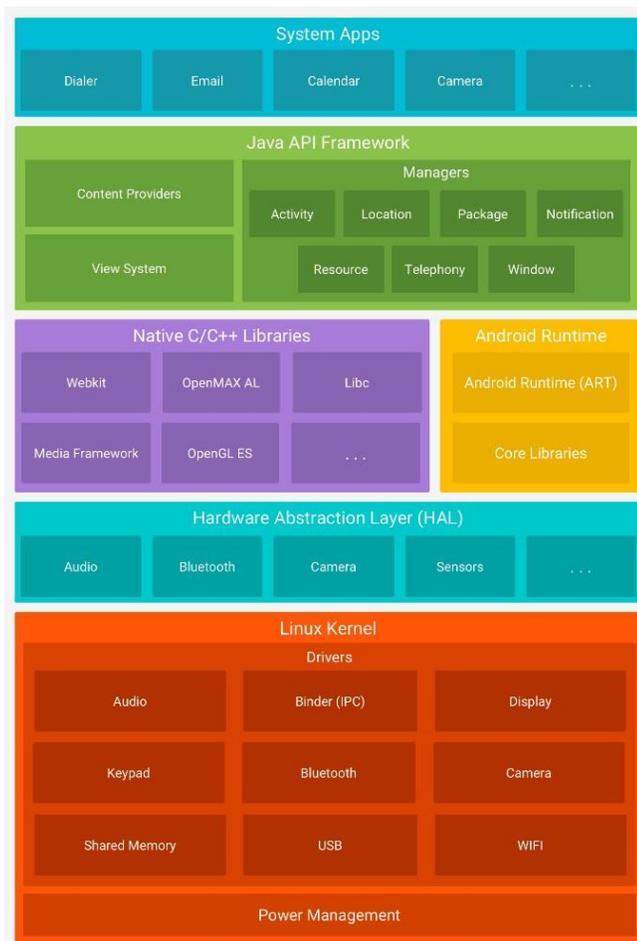
- Permite reutilizar fácilmente código.
- Hace sencilla la tarea de configurar y personalizar la compilación.
- Permite la distribución sencilla de código al resto del mundo, y fomenta el trabajo en equipo.
- Gestiona las dependencias de forma potente y cómoda (está basado en Maven).
- Permite la compilación desde consola, lo que nos puede hacer más sencilla la tarea de compilación en sistemas sin el entorno de desarrollo montado.
- Lo más importante es que hace increíblemente fácil la creación de diferentes versiones de la aplicación, por ejemplo, para hacer múltiples versiones para móviles o tablets, versiones de pago o gratuitas, etc. (Baña, 2016)

6.6.4 Arquitectura de Android Studio

Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma. En el siguiente diagrama, se muestran los componentes principales de la plataforma Android. (Android Studio,2021).

Ver imagen 2

Imagen 2. Pila de software de Android Studio



Android *Fuente:* (Android Studio, 2021).

a. Kernel de Linux

La base de la plataforma Android es el kernel de Linux. Por ejemplo, el tiempo de ejecución de Android (ART) se basa en el kernel de Linux para funcionalidades subyacentes, como la generación de subprocessos y la administración de memoria de bajo nivel.

El uso del kernel de Linux permite que Android aproveche funciones de seguridad claves y, al mismo tiempo, permite a los fabricantes de dispositivos desarrollar controladores de hardware para un kernel conocido. (Android Studio, 2021)

b. Capa de abstracción de hardware (HAL)

La capa de abstracción de hardware (HAL) brinda interfaces estándares que exponen las capacidades de hardware del dispositivo al marco de trabajo de la API de Java de nivel más alto. La HAL consiste en varios módulos de biblioteca y cada uno de estos implementa una interfaz para un tipo específico de componente de hardware, como el módulo de la cámara o de Bluetooth. Cuando el marco de trabajo de una API realiza una llamada para acceder a hardware del dispositivo, el sistema Android carga el módulo de biblioteca para el componente de hardware en cuestión. (Android Studio, 2021)

c. Tiempo de ejecución de Android

Para los dispositivos con Android 5.0 (nivel de API 21) o versiones posteriores, cada app ejecuta sus propios procesos con sus propias instancias del tiempo de ejecución de Android (ART). El ART está escrito para ejecutar varias máquinas virtuales en dispositivos de memoria baja ejecutando archivos DEX, un formato de código de bytes diseñado especialmente para Android y optimizado para ocupar un espacio de memoria mínimo. Crea cadenas de herramientas, como Jack, y compila fuentes de Java en código de bytes DEX que se pueden ejecutar en la plataforma Android.

Estas son algunas de las funciones principales del ART:

- Compilación ahead-of-time (AOT) y just-in-time (JIT)
- Recolección optimizada de elementos no utilizados (GC)
- En Android 9 (nivel de API 28) y versiones posteriores, se convierten los archivos de formato ejecutable (DEX) de un paquete de aplicaciones a un código de máquina más compacto
- Esto mejora la compatibilidad con la depuración, el generador de perfiles de muestras dedicado, las excepciones de diagnóstico detalladas y los informes de

fallos, y la capacidad de establecer puntos de control para supervisar campos específicos.

Antes de Android 5.0 (nivel de API 21), Dalvik era el entorno de ejecución del sistema operativo. Si la app se ejecuta bien en el ART, también debe funcionar en Dalvik, pero es posible que no suceda lo contrario.

En Android, también se incluye un conjunto de bibliotecas de entorno de ejecución centrales que proporcionan la mayor parte de la funcionalidad del lenguaje de programación Java; se incluyen algunas funciones del lenguaje Java 8, que usa el marco de trabajo de la API de Java. (Android Studio, 2021)

d. Bibliotecas C/C++ nativas

Muchos componentes y servicios centrales del sistema Android, como el ART y la HAL, se basan en código nativo que requiere bibliotecas nativas escritas en C y C++. La plataforma Android proporciona API del marco de trabajo de Java para exponer la funcionalidad de algunas de estas bibliotecas nativas a las apps. Por ejemplo, se puede acceder a OpenGL ES a través de la API de OpenGL de Java del marco de trabajo de Android para agregar a la app compatibilidad con los dibujos y la manipulación de gráficos 2D y 3D.

Si al desarrollar una app que requiera C o C++, se puede usar el NDK de Android para acceder a algunas de estas bibliotecas de plataformas nativas directamente desde el código nativo. (Android Studio, 2021)

e. Marco de trabajo de la API de Java

Todo el conjunto de funciones del SO Android está disponible mediante API escritas en el lenguaje Java. Estas API son los cimientos que se necesita para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, como los siguientes:

- Un sistema de vista enriquecido y extensible que puedes usar para compilar la IU de una app; se incluyen listas, cuadrículas, cuadros de texto, botones e incluso un navegador web integrable.
- Un administrador de recursos que te brinda acceso a recursos sin código, como strings localizadas, gráficos y archivos de diseño.
- Un administrador de notificaciones que permite que todas las apps muestren alertas personalizadas en la barra de estado.
- Un administrador de actividad que administra el ciclo de vida de las apps y proporciona una pila de retroceso de navegación común.
- Proveedores de contenido que permiten que las apps accedan a datos desde otras apps, como la app de Contactos, o compartan sus propios datos.

Los desarrolladores tienen acceso total a las mismas API del marco de trabajo que usan las apps del sistema Android. (Android Studio, 2021)

f. Apps del sistema

En Android se incluye un conjunto de apps centrales para correo electrónico, mensajería SMS, calendarios, navegación en Internet y contactos, entre otros elementos. Las apps incluidas en la plataforma no tienen un estado especial entre las apps que el usuario elige instalar; por ello, una app externa se puede convertir en el navegador web, el sistema de mensajería SMS o, incluso, el teclado predeterminado del usuario (existen algunas excepciones, como la app Settings del sistema).

Las apps del sistema funcionan como apps para los usuarios y brindan capacidades claves a las cuales los desarrolladores pueden acceder desde sus propias apps. Por ejemplo, si en tu app se intenta entregar un mensaje SMS, no es necesario que compiles esa funcionalidad tú mismo; como alternativa, puedes invocar la app de SMS que ya está instalada para entregar un mensaje al receptor que especifiques.

(Android Studio, 2021)

G. METODOLOGÍA

En este tema se analizará la metodología a utilizarse en el desarrollo del proyecto, la plataforma de desarrollo, seleccionando el entorno de programación para finalmente indicar la instalación y configuración de cada una de las herramientas necesarias para el desarrollo de una aplicación y detallando los pasos a seguir.

7.1 Metodología para desarrollo de aplicaciones móviles

Las metodologías de software seguro pueden lograr la construcción de un sistema informático eficiente, que cumpla con los requerimientos planteados, es una tarea realmente intensa y sobre todo difícil de cumplir. Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo.

Cabe destacar que todas las metodologías tienen puntos a favor como en contra, por lo que es muy importante conocerlas y ver cuáles la que brinda mejor soporte al desarrollo de una aplicación móvil, por lo que a continuación se analizara la metodología Extreme Programming (XP) ya que es la más adecuada para este proyecto (EcuRed, 2003).

7.2 Metodología XP (Extreme Programming)

La programación extrema o eXtreme Programming (XP) es una metodología de desarrollo de la ingeniería de software formulada por Kent Beck. Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de la XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más

realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos. (Wells D. , 2013)

Wells (2013) dice también que se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software. Esta metodología busca eliminar actividades relacionadas con la elaboración de algunos documentos de especificaciones que no tienen relación directa con el resultado final del software.

Se basa en los siguientes valores principales:

1. **Comunicación:** La necesidad de los desarrolladores de intercambiar ideas e información sobre el proyecto ya sea con los directores del proyecto o los usuarios de forma confiable, y fácil. La comunicación debe ser continua y rápida.
2. **Sencillez:** Cuando se tenga que elegir entre varias opciones, en lo posible elegir soluciones simples, sin que esto signifique aplicar enfoques simplistas; la programación extrema define un diseño simple en el que se realice el menor número posible de clases, métodos y que no tenga código duplicado.
3. **Retroalimentación:** Debe ser rápida en todos los niveles, principalmente se consigue ejecutando y probando el código, por lo que las entregas tempranas y frecuentes son muy importantes.
4. **Valor:** Todas las personas que participen en el proyecto deben tener la capacidad de expresar su valoración sobre el proyecto. Deberían ser abiertos y dejar que todos revisen e incluso modifiquen su trabajo. Los cambios no deberían ser vistos con terror y los desarrolladores deberían tener el valor de encontrar mejores soluciones y modificar el código siempre que sea necesario y factible.
5. **Respeto:** Debe manifestarse en diversas formas y situaciones, son la base para una buena relación y cooperación entre todos los componentes del equipo de trabajo. (Wells D., 2013)

La Programación Extrema se basa en:

- *Desarrollo en iteraciones*: En cada iteración se agregan nuevas funcionalidades, o se corrigen errores generando distintas versiones.
- *Pruebas unitarias continuas*: Estas pruebas están orientadas a comprobar que la aplicación mantenga sus funcionalidades.
- *Programación en parejas*: Se hace esto con el fin de que el código se discuta y revise mientras se desarrolla el programa, basado en que los dos programadores pueden complementarse, generando código de mejor calidad con menos errores.
- *Interacción entre los desarrolladores y el usuario*: Se minimizará el esfuerzo de ambas partes, pues se podrá tener una mejor comprensión de los problemas o necesidades de los usuarios y las soluciones que puedan brindar los desarrolladores.
- *Refactorización del código*: Busca hacer el código más fácil y mantenible, pero debe garantizar su correcto funcionamiento manteniendo las pruebas unitarias.
- *Propiedad del código compartida*: Busca que cualquier integrante del proyecto pueda colaborar modificando código hecho por otro. La existencia de errores se comprueba mediante las pruebas.
- *Simplicidad del diseño*: Los diseños simples pero funcionales permiten que posteriores funcionalidades se puedan agregar de manera fácil y rápida (Wells D. , 2013). Ver imagen 3.



Imagen 3. Flujo de características de la Programación Extrema.

Fuente: (Wells D. , 2013).

7.2.1 Proceso de la metodología XP

Un proyecto XP tiene éxito cuando el cliente selecciona el valor de negocio a implementar basado en la habilidad del equipo para medir la funcionalidad que puede entregar a través del tiempo (Extreme Programming Explained, 2000).

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo tanto el cliente como el programador aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos.

7.2.1 Fases de la metodología XP

Para el desarrollo de un proyecto tenemos 4 fases:

a. Planeación:

La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Una vez obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Los Conceptos básicos de la planificación son:

Las Historias de Usuarios, las cuales son descritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar.

El Plan de Entregas (Release Plan), establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto.

Plan de Iteraciones (Iteration Plan), las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.

Reuniones Diarias de Seguimiento (Stand – Up Meeting), el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones (Joskowicz, 2008).

b. Diseño:

La Metodología XP estimula el uso de las tarjetas CRC (clase-responsabilidad colaborador) o el diagrama de clases como un mecanismo eficaz para el diseño e incremento del software. El diseño ocurre tanto antes como después de que comienza la codificación. En esta fase se sugiere que los diseños sean simples y sencillos.

En la fase de diseño se debe desarrollar lo siguiente:

Simplicidad en el diseño. Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione. Se sugiere nunca adelantar la implementación de funcionalidades que no correspondan a la iteración en la que se esté trabajando.

Diagrama de modelado UML. Es un lenguaje unificado de modelado compuesto por diversos elementos gráficos que se combinan para conformar diagramas. Debido a que el UML es un lenguaje, cuenta con reglas para combinar tales elementos. La finalidad de los diagramas es presentar diversas perspectivas de un sistema, a las cuales se les conoce como modelo. Recordemos que un modelo es una representación simplificada de la realidad; el modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema.

Metáfora. Una “metáfora” es algo que todos entienden, sin necesidad de mayores explicaciones. La metodología XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, y guiar la estructura y arquitectura del mismo.

Tarjetas CRC. Las tarjetas C.R.C (Class, Responsibilities and Colaboración) es una técnica de diseño orientado a objetos propuesta por Kent Beck. Las principales características de las tarjetas son:

- Identificación de clases y asociaciones que participan del diseño del sistema.
- Obtención de las responsabilidades que debe cumplir cada clase.

- Establecimiento de cómo una clase colabora con otras clases para cumplir con sus responsabilidades.

La técnica CRC propone una forma de trabajo, preferentemente grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y cómo colaboran con otros para realizar tareas. Esta técnica utiliza las llamadas tarjetas CRC, las cuáles registran el nombre de las clases, sus responsabilidades y las otras clases con la que colaboran (Joskowicz, 2008).

c. Codificación:

La codificación constituye la fase más importante en el ciclo de vida de Programación Extrema (XP). La XP da prioridad a la programación sobre todas las otras tareas como la documentación para asegurar que el cliente reciba algún valor sustancial al terminar el proyecto. Los estándares relacionados a la programación incluyen:

Propiedad Colectiva del Código, En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, una pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re codificar.

Programación en Pares, XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y, por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

Integraciones Permanentes, Todos los desarrolladores necesitan trabajar siempre con la “última versión”. Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar

errores, solo una pareja de desarrolladores puede integrar su código a la vez (Joskowicz, 2008).

d. Pruebas:

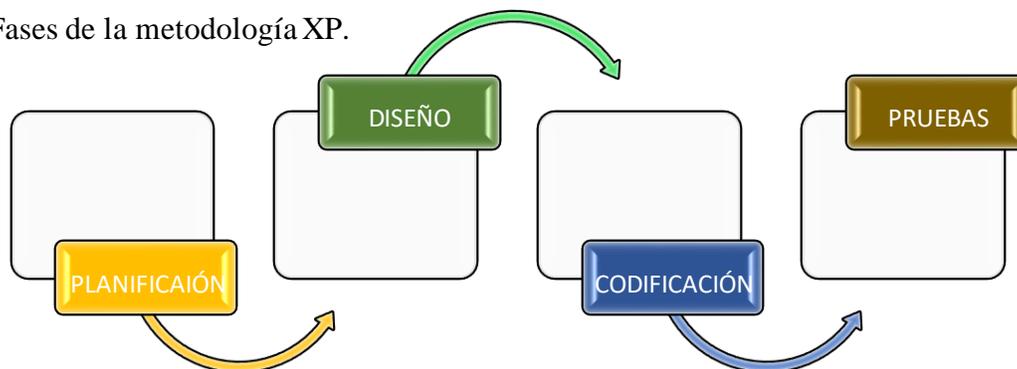
XP integra las pruebas junto a la fase de codificación en lugar de al final del desarrollo. Todos los códigos tienen las siguientes pruebas para eliminar errores, el código debe pasar todas las pruebas unitarias antes del lanzamiento.

Pruebas de Aceptación, Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación.

Pruebas Unitarias, Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-Driven Programmng”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código. (Joskowicz, 2008).

Gráfico 2.

Fases de la metodología XP.



Elaborado por: Cerda Fabricio, 2021.

7.2.1 Ciclo de vida de un proyecto XP

El ciclo de vida ideal de XP consiste de seis fases: Exploración, Planificación de la Entrega, Iteraciones, Producción, Mantenimiento y Muerte del Proyecto (Wells D. , 2001).

Fase I: Exploración: En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las

posibilidades de la arquitectura del sistema construyendo un prototipo. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología.

Fase II: Planificación de la Entrega: En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias.

Fase III: Iteraciones: Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Al final de la última iteración el sistema estará listo para entrar en producción.

Fase IV: Producción: La fase de producción requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente.

Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase. Las ideas que han sido propuestas y las sugerencias son documentadas para su posterior implementación (por ejemplo, durante la fase de mantenimiento).

Fase V: Mantenimiento: Mientras la primera versión se encuentra en producción, el proyecto XP debe mantener el sistema en funcionamiento al mismo tiempo que desarrolla nuevas iteraciones. Para realizar esto se requiere de tareas de soporte para el cliente. De esta forma, la velocidad de desarrollo puede bajar después de la puesta del sistema en producción. La fase de mantenimiento puede requerir nuevo personal dentro del equipo y cambios en su estructura.

Fase VI: Muerte del Proyecto: Es cuando el cliente no tiene más historias para ser incluidas en el sistema. Esto requiere que se satisfagan las necesidades del cliente en otros aspectos como rendimiento y confiabilidad del sistema. Se genera la documentación final del sistema y no se realizan más cambios en la arquitectura. La muerte del proyecto también ocurre cuando el sistema no genera los beneficios esperados por el cliente o cuando no hay presupuesto para mantenerlo. (Wells D. , 2001)

Cabe mencionar que no existe una metodología universal para aplicarla con éxito a cualquier proyecto de desarrollo de software. Toda metodología debe ser adaptada a todo el contexto del proyecto como son: recursos técnicos y humanos, tiempo de desarrollo, tipo de sistema, etc.

Sin embargo, la Programación Extrema proporciona una serie de ventajas para la realización del proyecto ya que ayudan al ahorro de tiempo y recursos.

- Permite la entrega de productos usables con mayor rapidez lo que permite la entrega de versiones continuas al usuario.
- Presenta un código más simple y fácil de entender que permite la reducción del número de errores en su desarrollo.

- Capacidad de respuesta ante cambios permitiendo la adaptación a nuevos requisitos de la empresa u organización.

Por estas y muchas otras ventajas es la Metodología que más se adecua a nuestro proyecto de desarrollo ya que pertenece a las “Metodologías Ágiles de Desarrollo” y como su nombre lo indica nos agiliza su desarrollo.

7.3 Especificaciones e instalación de la tecnología de desarrollo

Para el desarrollo de una aplicación Android, lo primero que se necesita es definir bien un entorno de desarrollo. Por lo que se explicara detalladamente los pasos necesarios para construir el entorno de desarrollo de aplicaciones Android. El entorno se puede implementar tanto en Windows como en Linux. Los requisitos para cada sistema operativo se pueden consultar en la web de Android, aunque el proceso es prácticamente idéntico en otros sistemas operativos. (Android Studio, 2021)

El proceso consta de los siguientes pasos:

1. Instalar y configurar JDK: Kit de desarrollo de Java.
2. Instalar y configurar Android Studio: Herramientas para desarrollar software.
3. Instalar y configurar el SDK de Android: Kit de desarrollo de aplicaciones Android, agregar APIs y componentes al SDK: descarga del resto de componentes necesarios para desarrollar aplicaciones.
4. Crear y Configurar un Dispositivo virtual Android o AVD (Android Virtual Device) para poder ejecutar la aplicación.
5. Instalar SQLite: Herramienta para administrar bases de datos.

7.3.1 Instalar y configurar JDK: Kit de desarrollo de Java

Las librerías del Java JDK (Java Development Kit) son indispensables para ejecutar el emulador de Android y algunas herramientas de depuración. Para instalar el entorno de desarrollo JDK 8 en Windows, se debe seguir los siguientes pasos:

1. Descargar el JDK del sitio web de Oracle

<https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>. En esta página se encuentran las diversas versiones, en este caso se utiliza la última versión que es la 8u111 / 8u112, para Windows 64bits. *Ver Imagen 4.*

Imagen 4.

Sitio Web de Oracle para descargar JDK.



Elaborado por: Cerda Fabricio, 2021.

2. Una vez descargado el JDK se procede a instalar en el computador, se escoge las características a instalar y se selecciona siguiente. *Ver imagen 5.*

Imagen 5.

Captura de pantalla de la instalación.



Elaborado por: Cerda Fabricio, 2021.

7.3.2 Instalar y configurar Android Studio

Android Studio es un paquete bastante completo e incluye emuladores de sistema operativo y aplicaciones especializadas para controlar y monitorear todos los elementos de una aplicación móvil. Para su instalación se seguirá los siguientes pasos:

1. Se descarga el paquete el cual se lo hace desde el sitio oficial en esta dirección:
<https://developer.android.com/studio>. *Ver imagen 6.*

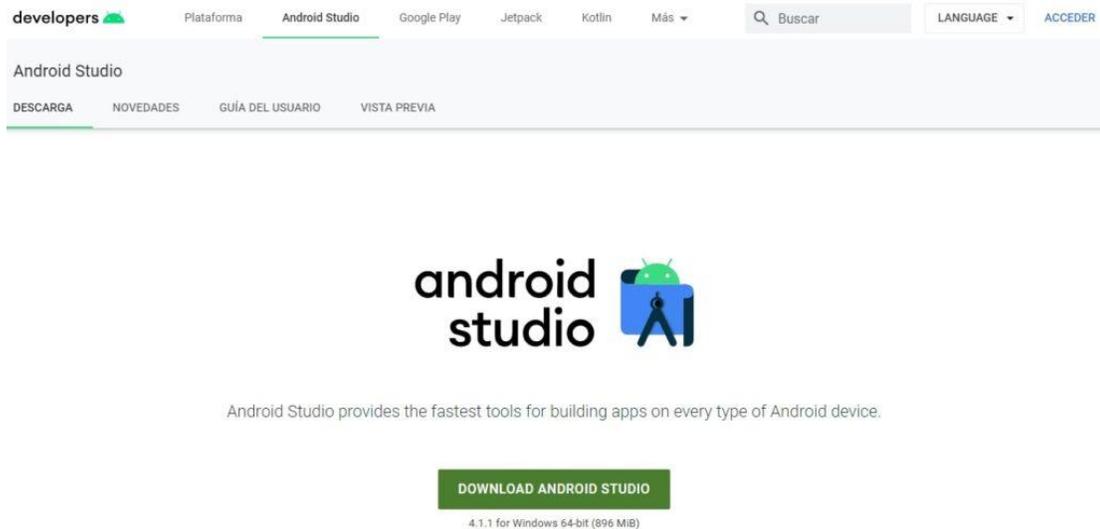


Imagen 6. Sitio Web Oficial de Android Studio.
Fuente: (Android Studio, 2021).

Dependiendo de la velocidad de la conexión a Internet, tardará más o menos, así que hay que tener paciencia. En cuanto termine, se da clic en el archivo descargado para iniciar la instalación.

2. Se inicia la instalación, aunque está disponible únicamente en inglés, lo cierto es que no hay mucha dificultad pues en la mayoría de pantallas solo se necesita pulsar Next, o siguiente. *Ver imagen 7,8.*

Imagen 7.

Captura de pantalla de proceso de instalación

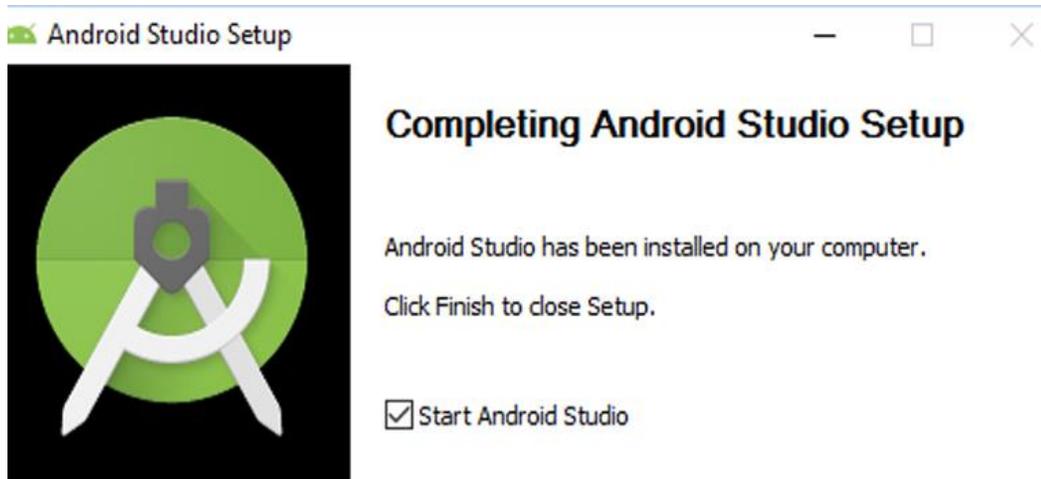


Elaborado propio por: Cerda Fabricio, 2021.

El único lugar donde se puede tener algo más de dudas es en la configuración sobre la memoria RAM máxima que se dedicará al emulador de Android. Lo cierto es que se puede cambiar esto más tarde y la configuración recomendada es... pues eso, la recomendada, así que en esta ventana lo más sensato es pulsar de nuevo Next. Por lo demás no hay mayor dificultad en la instalación y pronto comenzará el proceso real. Puede tardar lo suyo si el PC va con la potencia justa, así que debemos tener paciencia. En cuanto termine, pulsa Finish para que se abra Android Studio por primera vez. *Ver imagen 8.*

Imagen 8.

Captura de pantalla de proceso de instalación

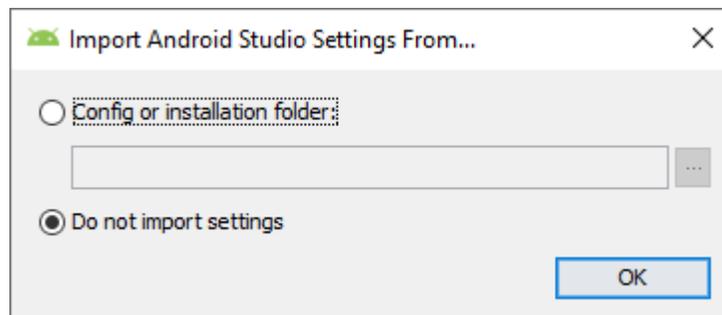


Elaborado propio por: Cerda Fabricio, 2021.

3. Se hace la configuración inicial, lo primero que se verá al abrir Android Studio por primera vez es una pantalla como la siguiente. En ella simplemente se pregunta si se quiere importar la configuración de una versión anterior. Generalmente Android Studio detectará la configuración por sí mismo, y si no tiene ninguna configuración anterior guardada se marcará Do not import settings, o no importar configuración. *Ver Imagen 9.*

Imagen 9.

Captura de pantalla de proceso de instalación

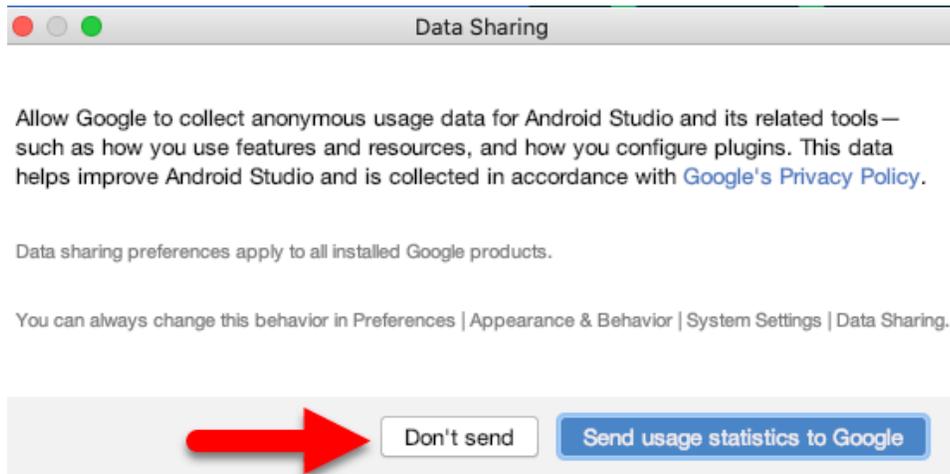


Elaborado por: Cerda Fabricio, 2021.

La siguiente ventana que aparecerá pide permiso para enviar datos de uso y estadísticas a Google, con el objetivo de que sean usados para mejorar la aplicación. Es totalmente opcional, así que se acepta con Send usage statistics to Google o no, en este caso haciendo clic en Don't send. *Ver imagen 10.*

Imagen 10.

Captura de pantalla de proceso de instalación



Elaborado propio por: Cerda Fabricio, 2021.

Tras este pre-asistente de configuración viene el asistente de configuración. En verdad son solo un par de ventanas con las opciones de configuración más importantes y que, en cualquier caso, siempre se podrá cambiar más tarde desde los ajustes. *Ver imagen 11.*

Imagen 11.

Captura de pantalla de proceso de instalación

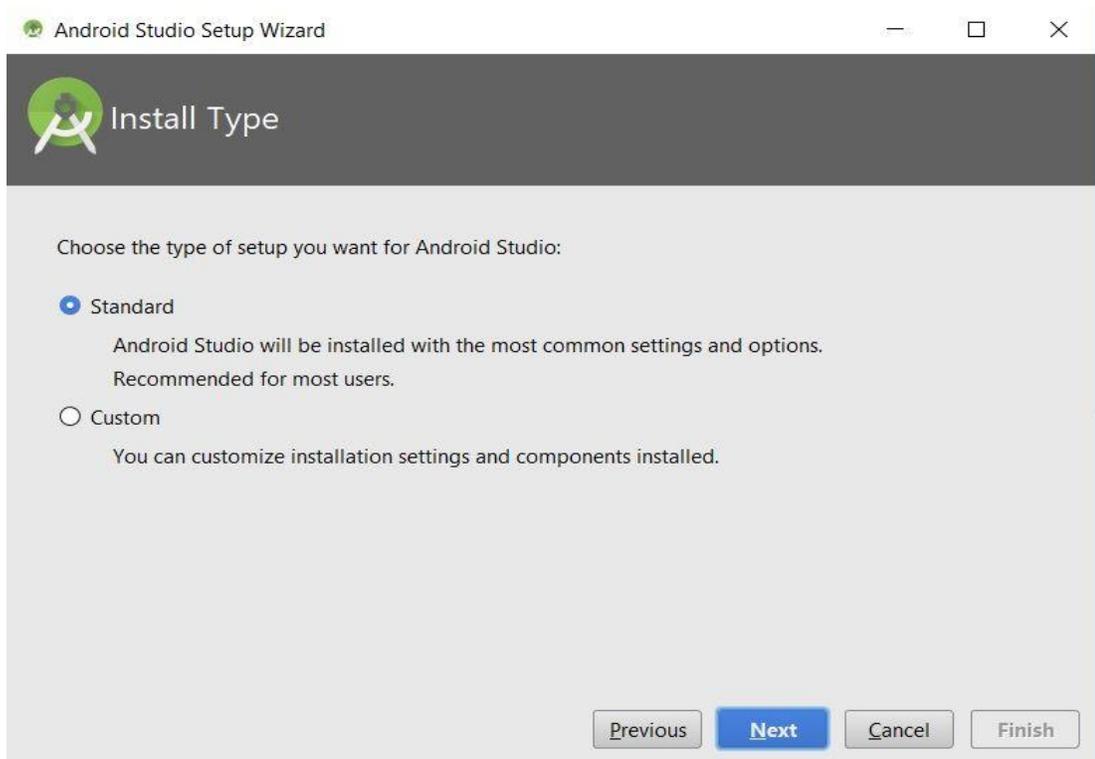


Elaborado propio por: Cerda Fabricio, 2021.

Lo primero que se pregunta es si se quiere hacer una instalación estándar o personalizada de Android Studio. En la gran mayoría de los casos, la instalación estándar es suficiente y ahorrará tiempo. *Ver imagen 12.*

Imagen 12.

Captura de pantalla de proceso de instalación

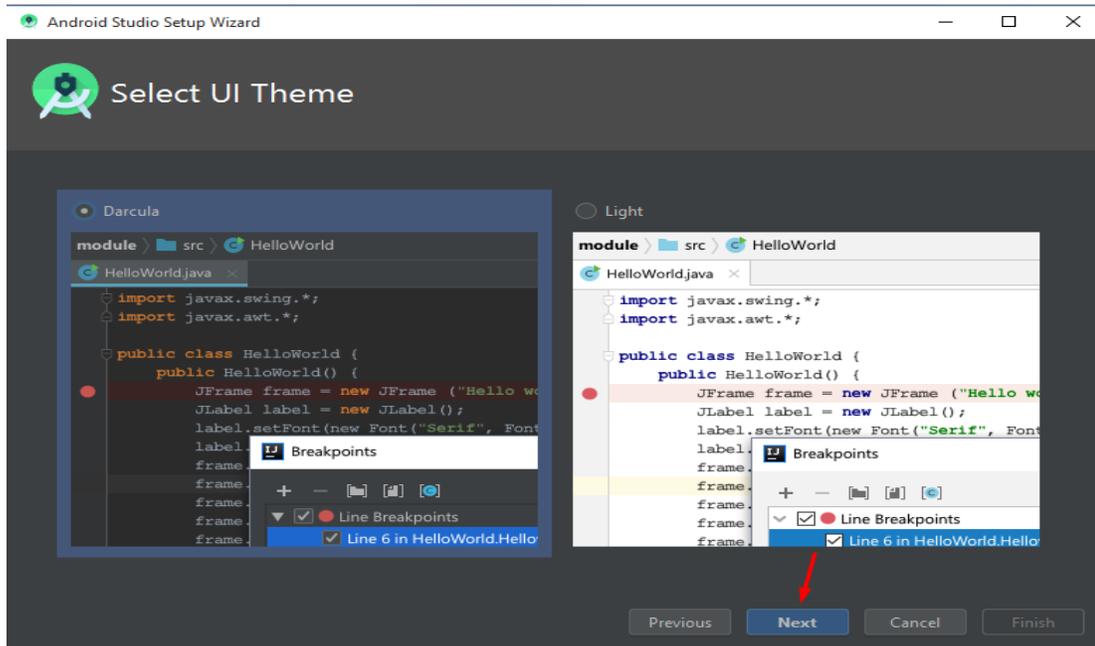


Elaborado por: Cerda Fabricio, 2021.

A continuación, se elige el tema a usar en el editor de código. De fábrica trae dos: el tema claro Light y el tema oscuro Drácula. Elegir el de preferencia, aunque siempre podremos cambiar más tarde desde las opciones, así como personalizar todos los colores y fuentes si así lo deseamos. *Ver imagen 13.*

Imagen 13.

Captura de pantalla de proceso de instalación

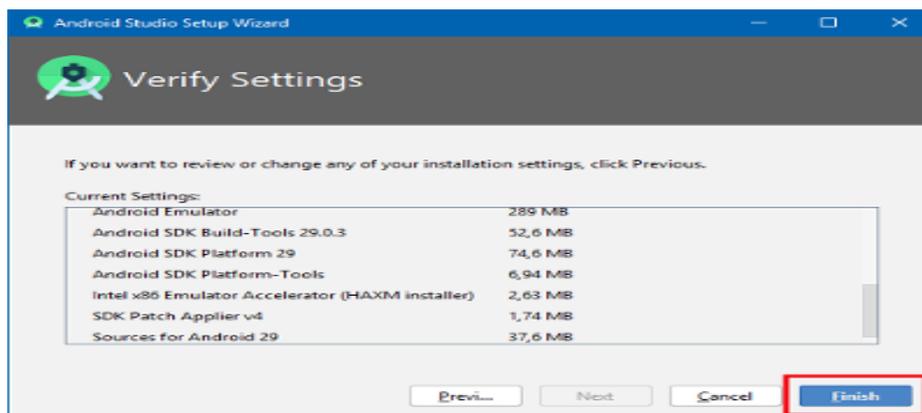


Elaborado por: Cerda Fabricio, 2021.

4. Se descarga los componentes, si se eligió la instalación estándar, Android Studio elige por nosotros los componentes que necesita descargar. Ver imagen 14.

Imagen 14.

Captura de pantalla de proceso de instalación



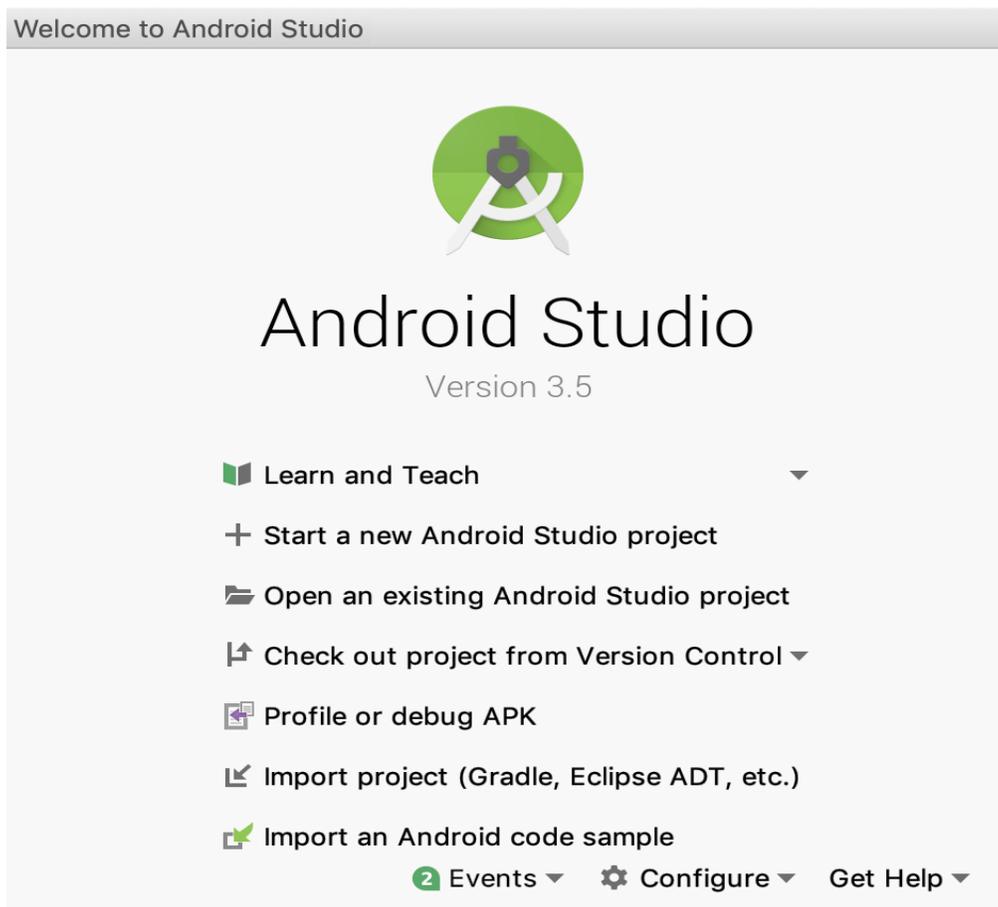
Elaborado por: Cerda Fabricio, 2021.

Una vez más la descarga seguramente supere los 500 MB, así que nos aseguramos de estar conectado a una red Wi-Fi y de tener suficiente espacio en el disco. Se deja que el PC haga el trabajo por nosotros y esperamos a que termine la descarga e instalación. Cuando termine, pulsamos Finish.

5. Android Studio, ya está instalado en el sistema y se mostrara la pantalla de bienvenida. Para iniciar con el primer proyecto solo se necesita hacer clic en Start a new Android Studio project. *Ver Imagen 15.*

Imagen 15.

Captura de pantalla de proceso de instalación



Elaborado por: Cerda Fabricio, 2021.

7.3.3 Instalar y configurar el SDK de Android

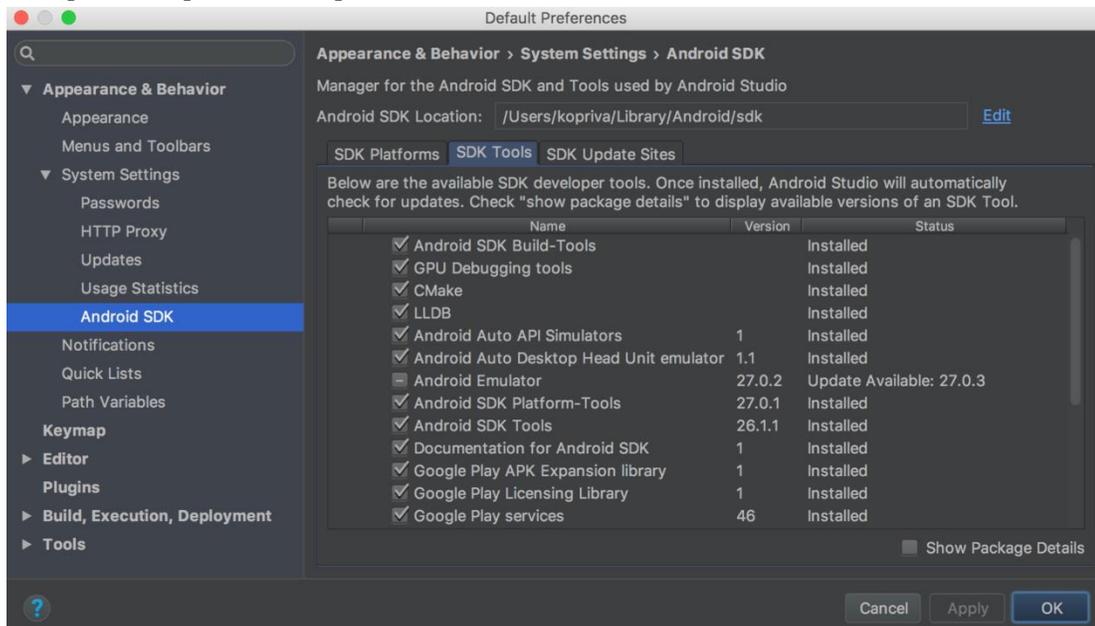
Ya se ha instalado Android Studio y el Android SDK, pero esto no es todo. Para aprovechar este software se necesita instalar una serie de herramientas, componentes y paquetes especializados para dotar de potencia a este entorno de programación. Para ello se descarga estos archivos, pero no hay de qué preocuparse ya que todo se realiza desde los mismos menús.

Para el siguiente paso, se vuelve abrir Android Studio, se da clic en configuración y allí el SDK Manager abrirá un menú con multitud de opciones, de entre todas ellas debemos quedarnos en la pestaña de Android SDK.

1. Si se quiere exprimir al máximo el SDK se debe descargar algunos paquetes adicionales. *Ver imagen 16.*

Imagen 16.

Captura de pantalla de proceso de instalación



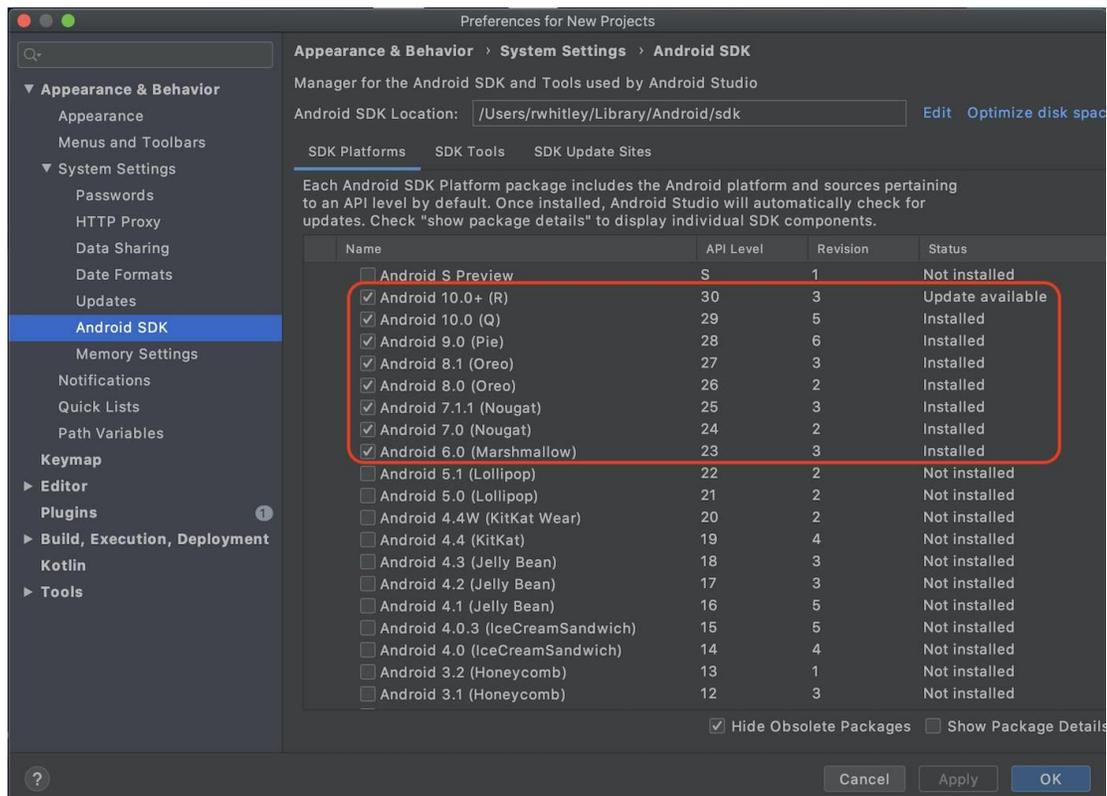
Elaborado por: Cerda Fabricio, 2021.

Una vez aquí hay que seleccionar el Android SDK Platform y las APIs de la versión de Android para la que se quiera programar. Para quienes quieran probar lo último, se recomienda descargar la última API disponible (en 2020 es la API level 27). Una vez que se selecciona estas versiones y sus archivos, dar clic a OK.

2. Cuando se instala el SDK de Android también hay que seleccionar y descargar herramientas como el Android SDK Tools, el Android SDK Platform-tools y la última versión del Android SDK Build-tools, todos ellos imprescindibles cuando se quiera instalar ROMs e imágenes de fábrica como las de las preview de Android. *Ver imagen 17.*

Imagen 17.

Captura de pantalla de proceso de instalación



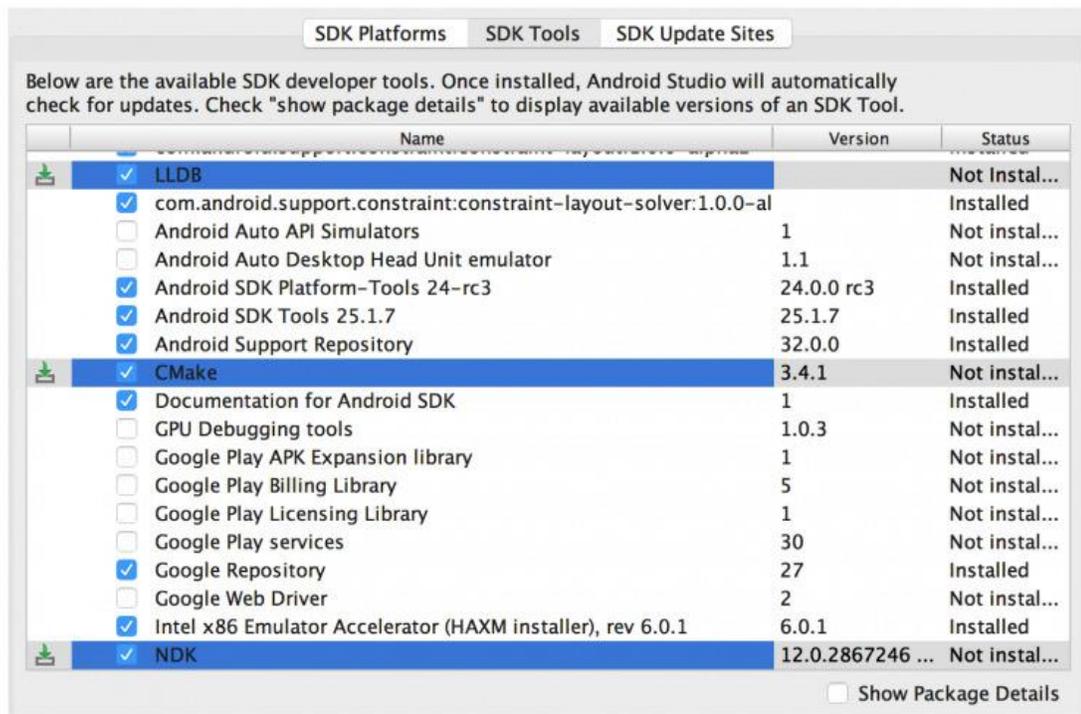
Elaborado por: Cerda Fabricio, 2021.

Además de esto, se deberá tener varios directorios extra relacionados con las API de Google enfocadas a los sistemas operativos Android para accesorios. Se refiere a Wear OS, Google Cast o Android TV. También cuentan con sus propias API y son muy recomendadas.

3. Android Repository y Android Support son dos librerías muy útiles que nos dan una gran cantidad de APIs compatibles con la mayoría de versiones de Android y si se quiere desarrollar aprovechando los Google Play Services se deberá descargar el Google Repository. *Ver imagen 18.*

Imagen 18.

Captura de pantalla de proceso de instalación



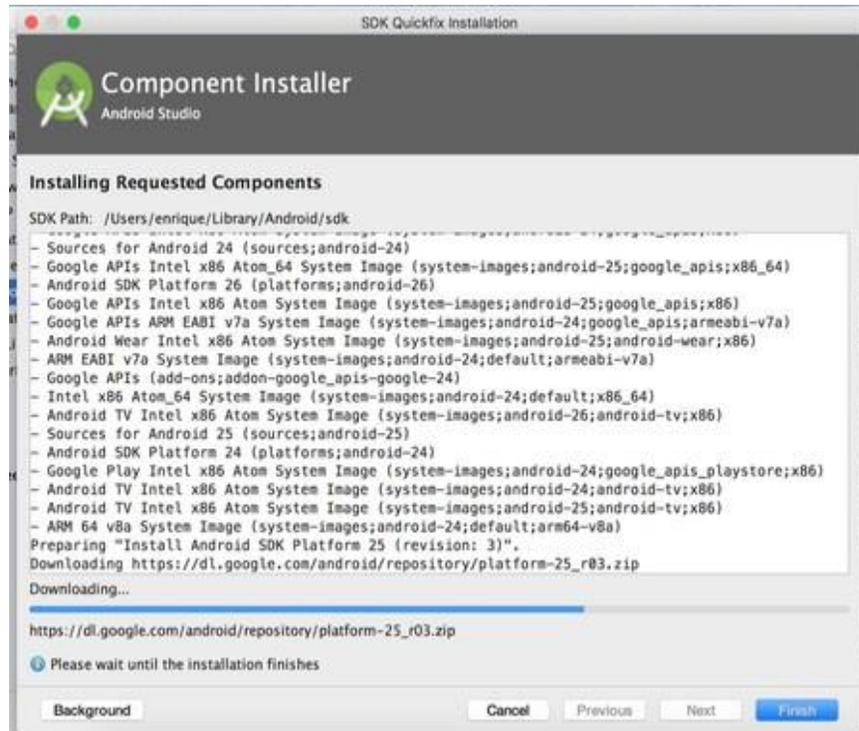
Elaborado por: Cerda Fabricio, 2021.

Estas recomendaciones también son extensibles al SDK Manager si se ha apostado por no instalar Android Studio. Con el paso del tiempo Android cuenta con más paquetes y APIs disponibles, esto facilita las tareas a la hora de crear

aplicaciones, pero para los iniciados les ofrece de repente una enorme cantidad de opciones que puede acabar confundiendo. Es algo positivo y cada vez hay más paquetes separados para que el usuario solo descargue lo que necesite. Ver imagen 19

Imagen 19.

Captura de pantalla de proceso de instalación



Elaborado por: Cerda Fabricio, 2021.

Para la inclusión de mapas en nuestra aplicación Android debemos seguir los siguientes pasos:

- Se descarga y se configura los Google Play Services: Viene con el SDK de Android por lo que se descarga con la ayuda del SDK Manager.
- Obtener una API key: Es una llave que estará asociada exclusivamente a nuestra aplicación por medio de los certificados que usa Android.

- Definir las especificaciones en el Application Manifest: Que se lo hace colocando una línea de código en el archivo AndroidManifest.xml.
 - Añadir un mapa a la aplicación: Se lo realiza añadiendo más líneas de código en los archivos xml como son: el main y el mainActivity.
4. Como paso final, una vez que se haya elegido los paquetes que se quiera añadir al Android SDK solo queda descargar e instalar. Para ello solo se debe de clickar en el cuadro en blanco de cada uno y seleccionarlos, después le dar a OK. Una vez aquí pedirá que se acepte diversas licencias de uso, después se pondrá a descargar todos los paquetes.

Una vez finalice el proceso de carga ya se tiene todo preparado para ponerse a crear aplicaciones Android o a trabajar con las imágenes de fábrica. Y por supuesto, si se quiere en algún momento añadir más paquetes únicamente se debe repetir el proceso.

7.3.4 Crear y Configurar un Dispositivo virtual Android o AVD

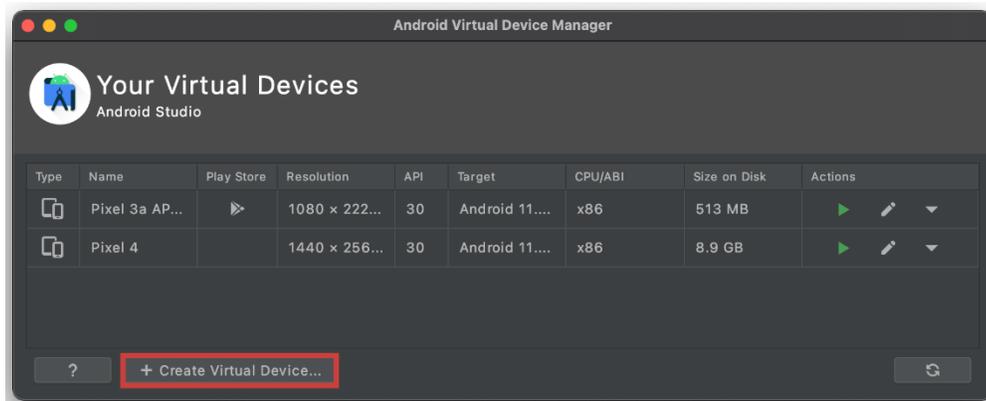
Para poder ejecutar o probar nuestro proyecto en Android debemos crear un dispositivo virtual Android o AVD (Android Virtual Device) lo cual se detalla los pasos para su creación:

Para crear un AVD nuevo:

1. Clic en Tools > AVD Manager para abrir el Administrador de AVD. *Ver imagen 20.*

Imagen 20.

Captura de pantalla de proceso de instalación

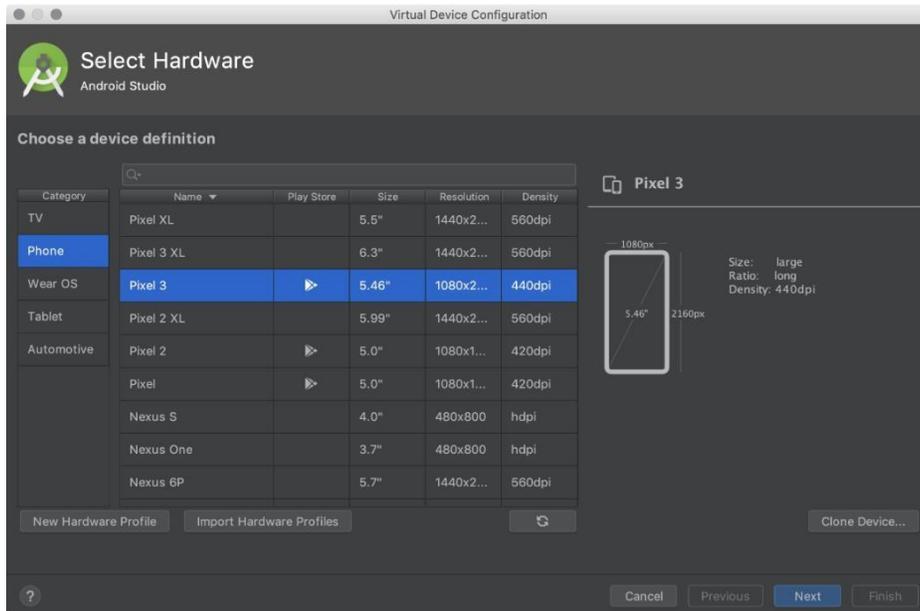


Elaborado por: Cerda Fabricio, 2021.

2. Clic en Create Virtual Device, en la parte inferior del diálogo de AVD Manager. Nos aparecerá la página Select Hardware. Ver imagen 21.

Imagen 21.

Captura de pantalla de proceso de instalación



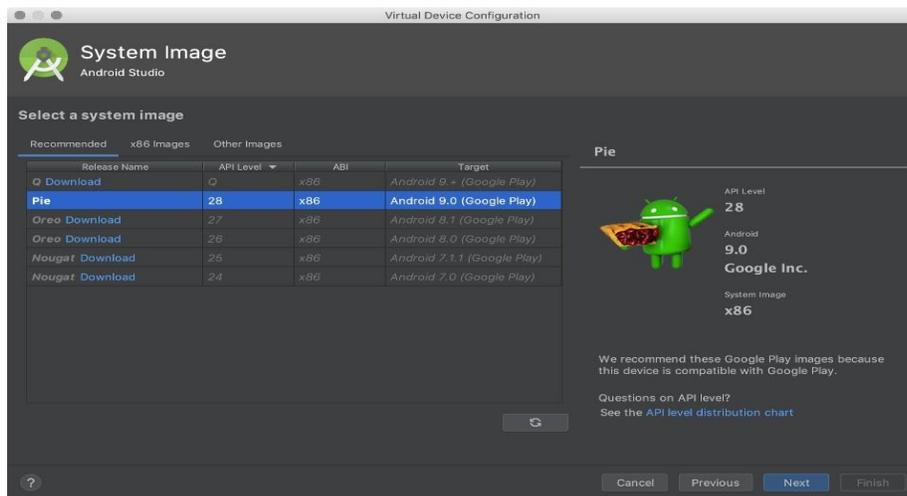
Elaborado por: Cerda Fabricio, 2021.

Hay que tener en cuenta que solo algunos perfiles de hardware incluyen Play Store por indicación. Esto indica que esos perfiles son totalmente compatibles con CTS (Conjunto de pruebas de compatibilidad) y se pueden usar imágenes de sistema que incluyan la app de Play Store.

3. Se selecciona un perfil de hardware y hacemos clic en Next. *Ver imagen 22.*

Imagen 22.

Captura de pantalla de proceso de instalación

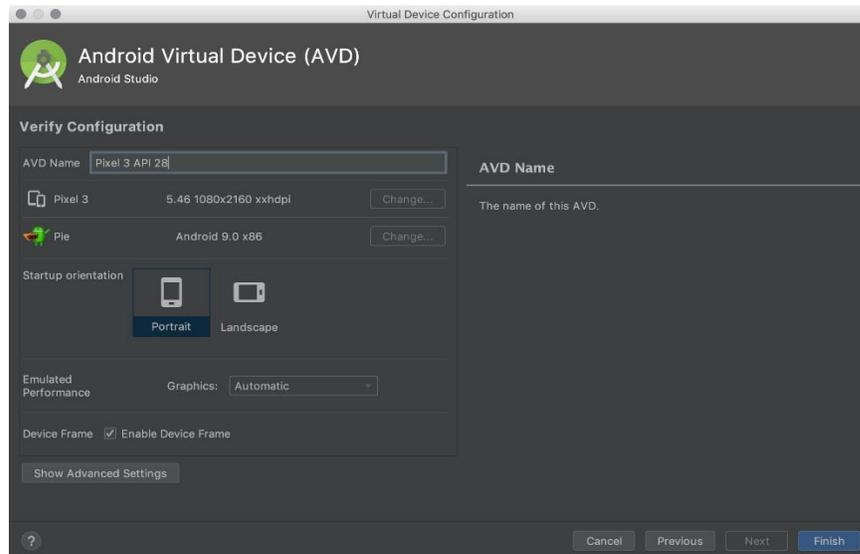


Elaborado por: Cerda Fabricio, 2021.

4. Seleccionar la imagen del sistema para un nivel de API determinado y hacemos clic en Next, aparecerá la página Verify Configuration. *Ver imagen 23.*

Imagen 23.

Captura de pantalla de proceso de instalación

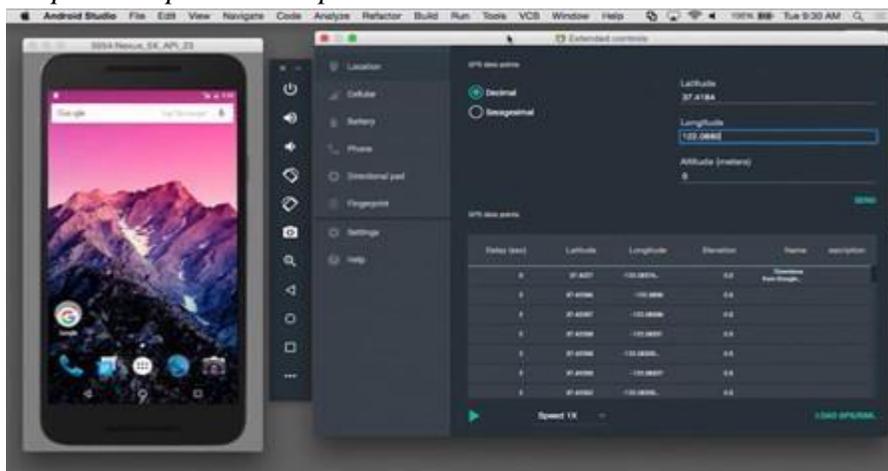


Elaborado por: Cerda Fabricio, 2021.

5. Modificar las propiedades de AVD según sea necesario y, luego dar clic en Finish, el nuevo AVD nos aparecerá en la página Your Virtual Devices o en el diálogo Select Deployment Target. *Ver imagen 24.*

Imagen 24.

Captura de pantalla de proceso de instalación



Elaborado por: Cerda Fabricio, 2021.

De esta manera se ha realizado la instalación y configuración de todas las herramientas necesarias para el desarrollo de la aplicación móvil Android.

H. RESULTADOS

A continuación, en este literal se tratará cada una de las Fases para el desarrollo de una aplicación móvil según la metodología Extreme Programming (XP) como son: *La Fase de Planificación* en donde se analiza involucrados y requerimientos funcionales y no funcionales del software. *La fase de Diseño* donde está toda la parte de diagramación, *La Fase de Codificación* que es concretamente la fase de programación y por último la *Fase de Pruebas* donde se realiza las pruebas respectivas con su debida documentación.

8.1 Fase de Planeación

Descripción de involucrados (Stakeholders)

A continuación, se muestra a los stakeholders o involucrados en el proyecto de desarrollo de la aplicación móvil Android utilizando palabras en el idioma Kichwa que permita el aprendizaje de operaciones matemáticas básicas para niños de 5 a 10 años de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo. *Ver Tabla 3*

Tabla 3.

Involucrados del proyecto

NOMBRE	CARGO	LUGAR	DESCRIPCIÓN
Mgs. Rubén Maximiliano Cerda Mamallacta	Rector	Escuela "Paulino Grefa"	Persona encargada de la parte Administrativa de la Escuela.
Lic. Morales Tapuy Andrea Estefanía	Docente	Escuela "Paulino Grefa"	Docente de la materia de Matemáticas de la Escuela.

Elaborado propio por: Cerda Fabricio, 2021.

Todos cumplen un papel fundamental dentro de la aplicación a desarrollar.

8.1.1 Historias de Usuario

Tabla 4.

Funciones del aplicativo

HISTORIA DE USUARIO	
Numero: 1	Usuario: Docente
Nombre de la historia: Funciones del aplicativo	
Prioridad en el negocio: Alta (Alta/Media/Baja)	
Puntos asignados: 4	Interacción asignada: 1
Responsable: Fabricio Cerda	
Descripción: Como docente deseo tener una aplicación móvil el cual mediante un juego se pueda enseñar las operaciones básicas en el idioma Kichwa a los alumnos con una edad promedio de 5 a 10 años.	
Criterio de aceptación:	
<ul style="list-style-type: none">• Desarrollo de la aplicación móvil para dispositivos Android.	

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 5.

Ingreso al sistema

HISTORIA DE USUARIO	
Numero: 2	Usuario: Docente
Nombre de la historia: Ingreso al sistema	
Prioridad en el negocio: Alta (Alta/Media/Baja)	
Puntos asignados: 3	Interacción asignada: 1
Responsable: Fabricio Cerda	
Descripción: Como docente deseo abrir la aplicación y entrar de manera inmediata	
Criterio de aceptación:	

- Debe abrir la aplicación sin necesidad de registrarse.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 6.

Diseño del aplicativo

HISTORIA DE USUARIO	
Numero: 3	Usuario: Docente
Nombre de la historia: Diseño del aplicativo	
Prioridad en el negocio: Alta (Alta/Media/Baja)	
Puntos asignados: 3	Interacción asignada: 1
Responsable: Fabricio Cerda	
Descripción: Como docente deseo que el diseño del aplicativo sea didáctico y de fácil de comprender	
Criterio de aceptación:	
<ul style="list-style-type: none"> • Pantalla de bienvenida. • Botón para comenzar a jugar 	

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 7.

Contenido del aplicativo

HISTORIA DE USUARIO	
Numero: 4	Usuario: Docente
Nombre de la historia: Contenido del aplicativo	
Prioridad en el negocio: Alta (Alta/Media/Baja)	
Puntos asignados: 5	Interacción asignada: 1
Responsable: Fabricio Cerda	

Descripción: Como docente deseo que el aplicativo contenga contenido multimedia enfocada para niños y que pueda captar la atención y fácil de comprender.

Criterio de aceptación:

- Imágenes con frutas y números

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 8.

Dificultad del sistema

HISTORIA DE USUARIO

Numero: 5

Usuario: Docente

Nombre de la historia: Dificultad del sistema

Prioridad en el negocio: Alta
(Alta/Media/Baja)

Puntos asignados: 5

Interacción asignada: 1

Responsable: Fabricio Cerda

Descripción: Como docente deseo que las operaciones matemáticas inicien en la suma, mientras haya un número de aciertos avance a la resta y de la misma manera terminar en las multiplicaciones.

Criterio de aceptación:

- Niveles para avanzar en cada operación matemática.
- Al terminar los niveles mensaje de “FELICITACIONES”.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 9.

Disponibilidad del aplicativo

HISTORIA DE USUARIO

Numero: 6

Usuario: Docente

Nombre de la historia: Disponibilidad del aplicativo

Prioridad en el negocio: Alta
(Alta/Media/Baja)

Puntos asignados: 5

Interacción asignada: 2

Responsable: Fabricio Cerda

Descripción: Como docente deseo el archivo .apk para la distribución e instalación en los dispositivos móviles.

Criterio de aceptación:

- Entregar archivo .apk en una memoria usb.

Elaborado propio por: Cerda Fabricio, 2021.

8.1.2 El Plan de Entregas (Release Plan)

Basándonos en las historias de usuario definidas para el desarrollo de la aplicación móvil, se ha elaborado el siguiente plan de entrega, el cual muestra las historias de usuario que se llevarán a cabo en cada iteración. Para este plan de entrega se ha tomado en cuenta la prioridad y el esfuerzo de cada historia de usuario.

Tabla 10.

Plan de entregas

Historias	Interacción	Prioridad	Esfuerzo	Fecha de Inicio	Fecha final
Historias 1	1	Alta	4	10/12/2020	1/12/2020
Historias 2	1	Alta	3	10/12/2020	1/12/2020
Historias 3	1	Alta	3	5/1/2021	6/1/2021
Historias 4	1	Alta	5	10/1/2021	11/1/2021
Historias 5	1	Alta	5	23/1/2021	24/1/2021
Historias 6	2	Alta	5	3/2/2021	4/2/2021

Elaborado propio por: Cerda Fabricio, 2021.

8.1.3 Plan de Iteraciones (Iteration Plan)

Después de haber hecho cálculos, como resultado se obtiene que el desarrollo del proyecto tiene 2 iteraciones para el aplicativo móvil.

Iteración 01:

- Funciones del aplicativo
- Ingreso al sistema
- Diseño del aplicativo
- Contenido del aplicativo
- Dificultad del aplicativo

Iteración 02:

- Disponibilidad del aplicativo

8.1.4 Reuniones Diarias de Seguimiento (Stand – Up Meeting)

La metodología extreme programming (XP) nos sugiere la reunión diaria de seguimiento, pero en este proyecto no se consideró como una característica fundamental a tomarse en cuenta, pues el proyecto se lo lleva a cabo por una sola persona. Por esta razón la etapa no es necesaria ya que para el desarrollo de la aplicación utilizamos información recolectada al inicio del proyecto.

8.1.5 Requerimientos funcionales de la aplicación

Disponibilidad del aplicativo

- Entregar archivo. apk para la distribución e instalación en los dispositivos móviles.

Ingreso al sistema

- Abrir la aplicación en el dispositivo móvil

Diseño

- Pantalla de bienvenida

Dificultad

- Botón de comenzar
- Imágenes acordes a la edad de los estudiantes
- Operaciones matemáticas básicas (Suma, Resta, Multiplicación)

8.1.6 Requerimientos no funcionales de la aplicación

a. Rendimiento de la aplicación

- La aplicación ofrecerá la respuesta de datos en tiempo real
- El tiempo de respuesta de datos al usuario no debe tardar más de 5 segundos.

b. Seguridad y privacidad

- No requiere de permisos para que la aplicación pueda iniciarse en el dispositivo.

c. Disponibilidad

- La aplicación se entregará a la institución Educativa en un dispositivo de almacenamiento portátil para que pueda ser distribuida con sus alumnos.
- La aplicación no dependerá de una conexión a internet o plan de datos, la aplicación cuenta con una base de datos interna y se podrá visualizar todo el contenido de la aplicación.

d. Mantenibilidad

- La aplicación estará en constante mantenimiento ya que se podría agregar más niveles para grados más altos con nuevas operaciones matemáticas.

e. Portabilidad

- En el sistema desarrollado ofrece compatibilidad con otras plataformas Android desde la versión 4.0. ya sea en una Tablet o un Smartphone.

f. Operabilidad

- La aplicación podrá ser operada por cualquier usuario.

8.1.7 Restricciones de la aplicación

En dispositivo móvil en donde se ejecutará la aplicación y para su correcto funcionamiento se deberá tener los siguientes requisitos como mínimos, además necesitaremos almacenamiento dentro el dispositivo donde se almacenará la información descargada.

- Procesador: 1 GHz Dual Core
- Memoria RAM: 1GB.
- Almacenamiento: 16 MB disponibles.
- Pantalla de 5” pulgadas en adelante.
- Sistema operativo. Android 4.0 o superior

Cabe mencionar y recalcar que todas las restricciones descritas y analizadas anteriormente es solo una sugerencia para el óptimo funcionamiento de la aplicación, es decir que sea rápida y tenga una buena visualización, y no quita o restringe que la aplicación pueda funcionar en dispositivos de menor gama o incluso en Smartphone que tienen la pantalla más pequeña que una Tablet.

8.2 Fase de Diseño

Se elaborarán diseños breves que sirven de referencia para la implementación. Otra práctica fundamental de la metodología de Programación Extrema (XP) es utilizar diseños tan simples como sea posible. El principio es utilizar el diseño más sencillo que consiga que todo funcione evitando diseñar características extra y que tomaran demasiado tiempo.

8.2.1 Simplicidad en el diseño

El software debe ser lo más simple posible, pero sin dejar de considerar todas las necesidades de los usuarios. Se muestra un diseño de la estructura de la función de la aplicación.

8.2.1.1 Diagrama Arquitectónico

Imagen 25.

Diagrama arquitectónico de la aplicación



Elaborado propio por: Fabricio Cerda, 2021.

8.2.2 Diagrama de modelado UML.

A continuación, se empleará el modelado UML (Lenguaje Unificado de Modelado), que se puede usar para modelar distintos tipos de sistemas. En este proyecto se ha optado en la realización de los Diagramas más relevantes como son:

- Diagramas de Casos de Uso.
- Diagramas de Clases.

- Diagramas de Secuencia.

8.2.2.1 Diagramas de caso de uso

A continuación, identificaremos los actores. *Ver Tabla 11*

Tabla 11.

Actores de la aplicación

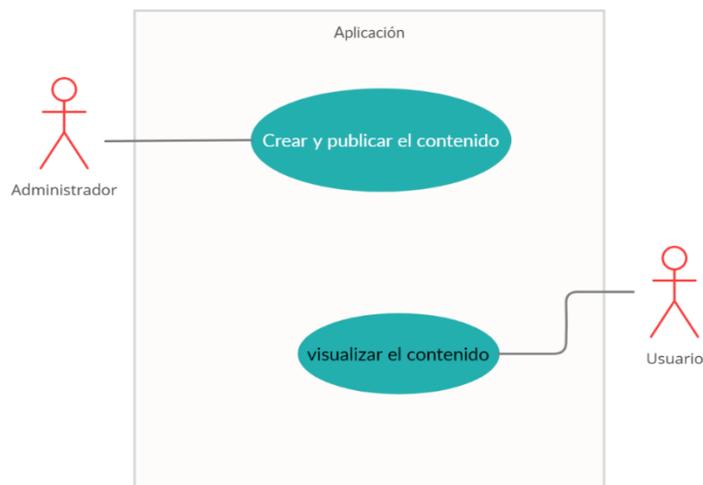
Actores de la aplicación	
Administrador	,
Usuario	Usar la aplicación y visualizar el contenido

Elaborado propio por: Cerda Fabricio, 2021.

Empecemos con las primeras acciones descritas en el caso de uso textual: el usuario buscando el contenido de la aplicación. *Ver imagen 26.*

Imagen 26.

Caso de uso general de la aplicación



Elaborado propio por: Cerda Fabricio, 2021.

a. Especificación de caso de uso

Ingreso a la aplicación. Ver Tabla 12, imagen 27.

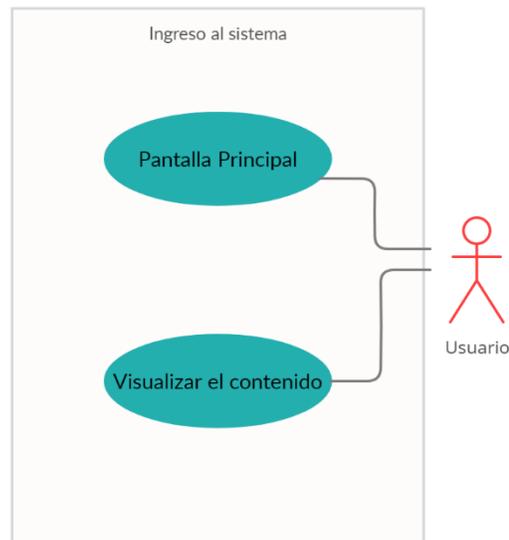
Tabla 12.

Ingreso a la aplicación.

CASO DE USO 1	
Nombre	Ingresar a la aplicación.
Descripción	Visualizar el contenido.
Actores	Agente usuario.
Precondición	Tener instalado el apk en el dispositivo móvil Android.
Flujo del sistema	La aplicación carga la información de su base de datos. La aplicación muestra la pantalla principal.

Elaborado propio por: Cerda Fabricio, 2021.

Imagen 27.
Ingreso a la aplicación.



Elaborado por: Cerda Fabricio, 2021.

Navegación dentro de la aplicación. Ver tabla 13, imagen 28.

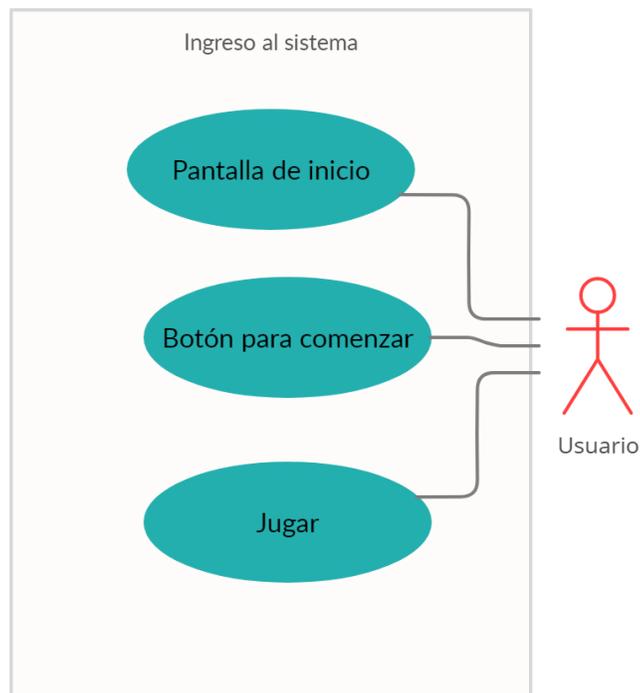
Tabla 13.

Navegación en la aplicación.

CASO DE USO 2	
Nombre	Navegación dentro de la aplicación.
Descripción	Visualizar el contenido.
Actores	Agente usuario.
Precondición	Tener instalado el apk en el dispositivo móvil Android.
Flujo del sistema	La aplicación muestra el contenido que se dispone en ella.

Elaborado propio por: Cerda Fabricio, 2021.

Imagen 28.
Visualización del contenido.



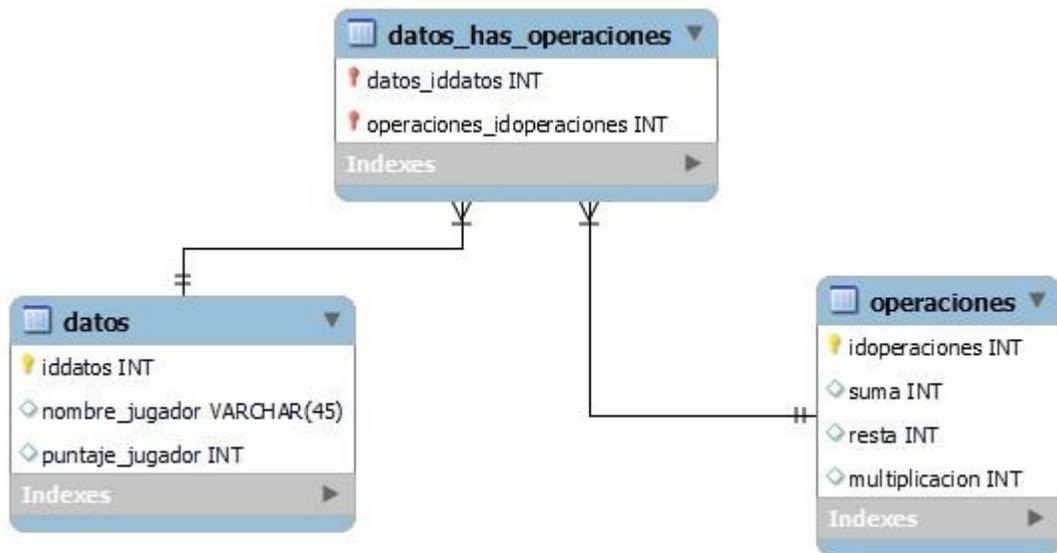
Elaborado por: Cerda Fabricio, 2021.

8.2.2.2 Diagrama de Clases

El diagrama que se mostrara a continuación está directamente relacionado con las clases que se utilizaron en la aplicación móvil, ya que para poder construir o desarrollar la misma fue necesario crear nuestras propias tablas en la Base de Datos de nuestro proyecto Android (SQLite). Es importante recalcar que las tablas y clases son las mismas que existen en la base de datos principal de donde extraemos los datos.

El en siguiente Diagrama vamos a ver todas las clases con sus respectivos atributos y como están relacionados unas con otras. *Ver imagen 29.*

*Imagen 29.
Diagrama de relación entre las tablas.*



Elaborado propio por: Cerda Fabricio, 2021.

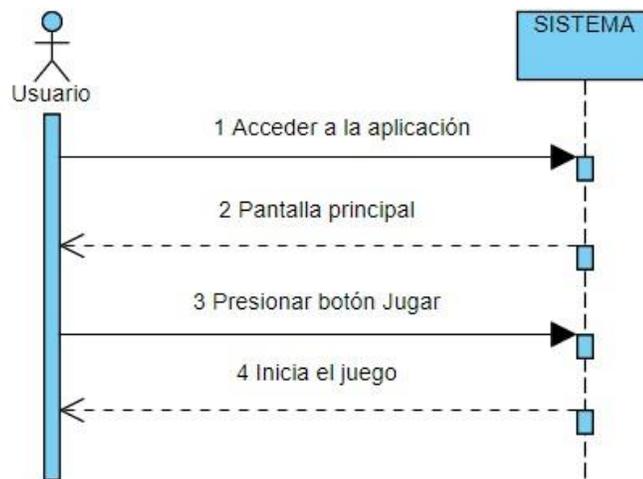
Como puede ver, hay 2 tablas SQL (`datos`, `operaciones`) y 1 índice (`iddatos INT`) que también es clave foránea. Este esquema de base de datos corresponde a modelos de base de datos en el código.

8.2.2.3 Diagramas de secuencia

Muestra una interacción que representa la secuencia de mensajes entre instancias de clases, componentes, subsistemas o actores. El tiempo fluye por el diagrama y muestra el flujo de control de un participante a otro. Se utilizan diagramas de secuencia para visualizar instancias y eventos en lugar de clases y métodos. En el diagrama, puede aparecer más de una instancia del mismo tipo. También puede haber más de una ocurrencia del mismo mensaje.

1. Diagrama de secuencia, ingresar al sistema. *Ver imagen 30.*

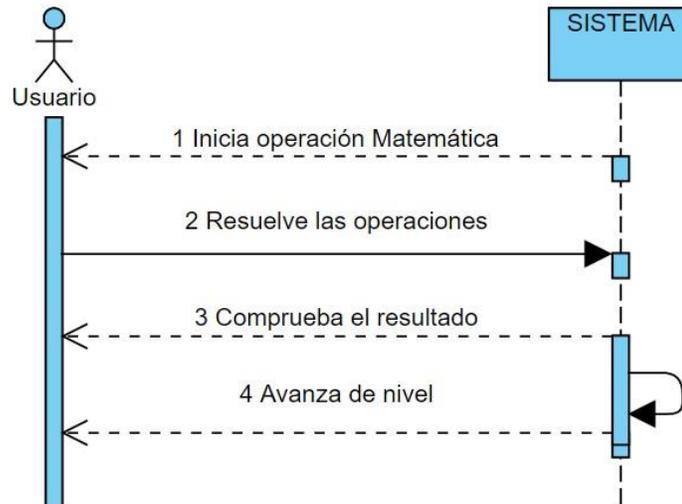
*Imagen 30.
Diagrama de secuencia, ingresar al sistema.*



Elaborado propio por: Cerda Fabricio, 2021.

2. Diagrama de secuencia, procesos. Ver imagen 31.

Imagen 31.
Diagrama de secuencia, procesos.



Elaborado propio por: Cerda Fabricio, 2021.

8.2.3 Metáfora

Explicaremos de manera sencilla el propósito del proyecto y guiar la estructura, arquitectura del mismo. La Metáfora del sistema nos permite descubrir de una manera clara lo que va hacer el sistema a partir de las historias de usuario, para ello se usa un vocabulario técnico y entendible para el usuario.

APLICACIÓN PARA EL APRENDIZAJE DE OPERACIONES MATEMÁTICAS EN EL IDIOMA KICHWA PARA NIÑOS DE 5 A 10 AÑOS DE EDAD

La aplicación permitirá el aprendizaje de operaciones matemáticas básicas mediante un juego, permitirá al jugador visualizar los números en el idioma Kichwa y con ejemplos para que su comprensión sea más rápida, consta de 3 niveles los cuales son las sumas, restas y multiplicación a medida que su acierto sea correcto ira avanzando de

dificultad es decir que avanzara a la siguiente operación y de esta manera pondrá en práctica sus habilidades matemáticas de una forma más divertida y entretenida.

8.2.4 Tarjetas CRC

Tarjetas CRC de la Aplicación

En esta técnica utilizamos las llamadas tarjetas CRC, las cuáles registran el nombre de las clases, sus responsabilidades en dicha clase que son sus objetivos, y los colaboradores que son otras clases que ayudan a conseguir cumplir a esta con sus responsabilidades. Se realiza una tarjeta por cada historia de usuario.

Tabla 14.

Tarjeta CRC Ingreso al sistema

Clase: Ingreso Sistema	
Responsabilidades:	Colaboradores:
Mostrar pantalla principal	MainActivity
Direccionar a las diferentes interfaces	

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 15.

Tarjeta CRC Diseño del sistema

Clase: Diseño Sistema	
Responsabilidades:	Colaboradores:
Direccionar a las diferentes interfaces	AdminSQLiteOpenHelper

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 16.

Tarjeta CRC Operaciones del sistema

Clase: OperacionesSistema	
Responsabilidades:	Colaboradores:
Direccionar a las diferentes niveles	onCreate
	AdminSQLiteOpenHelper
	MainActivity

Elaborado propio por: Cerda Fabricio, 2021.

8.3 Fase de Codificación

En esta fase se mostrará una parte del código y la interfaz gráfica que se utilizó para llevar a cabo el proyecto.

8.3.1 Propiedad Colectiva del Código

La construcción de una aplicación en Android se debe adaptar a la estructura estándar que nos muestra el IDE de desarrollo Android Studio. En esta etapa se detalla parte del código básicamente de algunas líneas donde sea importante acotar de las clases más relevantes de nuestro proyecto y la estructura generada para la aplicación.

- **Fragmento de La Clase Java MainActivity**

```
• package com.example.frutiapp;

import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.inputmethod.InputMethodManager;
```

```

import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

private EditText et_Nombre;
private ImageView iv_personaje;
private TextView tv_bestScore;
private MediaPlayer mp;

    int num_aleatorio = (int) (Math.random() * 10);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContent(R.layout.activity_main);

        et_Nombre = (EditText)findViewById(R.id.txt_nombre);
iv_personaje = (ImageView)findViewById(R.id.imageView_Personaje);
tv_bestScore = (TextView)findViewById(R.id.textView_bestScore);
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        int id;
        if(num_aleatorio == 0 || num_aleatorio == 10){
            id = getResources().getIdentifier("mango", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 1 || num_aleatorio == 9){
id = getResources().getIdentifier("fresa", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 2 || num_aleatorio == 8){
id = getResources().getIdentifier("manzana", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 3 || num_aleatorio == 7){
id = getResources().getIdentifier("sandia", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 4 || num_aleatorio == 5 ||
num_aleatorio == 6){
            id = getResources().getIdentifier("uva", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        }
        AdminSQLiteOpenHelper admin = new
AdminSQLiteOpenHelper(this, "BD", null, 1);

```

```

        SQLiteDatabase BD = admin.getWritableDatabase();
        Cursor consulta = BD.rawQuery(
            "select * from puntaje where score = (select max(score) from
puntaje)", null);
        if(consulta.moveToFirst()){
            String temp_nombre = consulta.getString(0);
            String temp_score = consulta.getString(1);
            tv_bestScore.setText("YALLISKANKI: " + temp_score + "
Kan: " + temp_nombre);//Mejor puntaje
            BD.close();
        } else {
            BD.close();
        }

        mp = MediaPlayer.create(this, R.raw.alphabet_song);
        mp.start();
        mp.setLooping(true);

    }
    public void Jugar(View view){
        String nombre = et_Nombre.getText().toString();

        if(!nombre.equals("")){
            mp.stop();            mp.release();

            Intent intent = new Intent(this,
Main2Activity_Nivel1.class);

            intent.putExtra("jugador", nombre);
            startActivity(intent);            finish();

        } else {
            Toast.makeText(this, "KANPA SHUTITA KILLKANA KANKI",
Toast.LENGTH_LONG).show();//debes escribir tu nombre
            et_Nombre.requestFocus();
            InputMethodManager imm =
(InputMethodManager) getSystemService(this.INPUT_METHOD_SERVICE);
            imm.showSoftInput(et_Nombre,
InputMethodManager.SHOW_IMPLICIT);
        }
    }
}

```

```
@Override  
public void onBackPressed(){  
}  
}
```

Clases Java donde definimos la funcionalidad de cada actividad, la clase Java extiende una Activity la cual contiene un método principal llamada onCreate en el cual se llamará a los métodos que conforman la actividad.

- **Layout**

```

• <?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/globos"
    tools:context=".MainActivity" >

    <ImageView
      android:id="@+id/imageView_Personaje"
      android:layout_width="239dp"
      android:layout_height="173dp"
      android:layout_alignParentStart="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentTop="true"
      android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
      android:layout_marginStart="81dp"
      android:layout_marginLeft="81dp"
      android:layout_marginTop="23dp"
      android:layout_marginEnd="90dp"
      android:layout_marginRight="90dp"
      android:contentDescription="@string/app_name"
      app:srcCompat="@drawable/manzana" />

    <EditText
      android:id="@+id/txt_nombre"
      android:layout_width="244dp"
      android:layout_height="41dp"
      android:layout_below="@+id/imageView_Personaje"
      android:layout_alignParentStart="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentEnd="true"
      android:layout_alignParentRight="true"
      android:layout_marginStart="86dp"
      android:layout_marginLeft="86dp"
      android:layout_marginTop="33dp"
      android:layout_marginEnd="81dp"
      android:layout_marginRight="81dp"
      android:autoFillHints=""
      android:background="@drawable/fondoet"
      android:ems="10"
      android:hint="@string/et_nombre"
      android:inputType="textPersonName"

```

```

android:maxLength="15"
android:paddingLeft="8dp"
android:paddingRight="50dp"
android:textColor="@color/colorPrimaryDark"
android:textColorHint="@color/colorPrimaryDark"
android:textSize="25sp" />

    <Button
        android:id="@+id/button"
android:layout_width="244dp"
android:layout_height="43dp"
android:layout_above="@+id/textView_bestScore"
android:layout_below="@+id/txt_nombre"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_marginStart="86dp"
android:layout_marginLeft="86dp"
android:layout_marginTop="21dp"
android:layout_marginEnd="81dp"
android:layout_marginRight="81dp"
android:layout_marginBottom="150dp"
android:background="@drawable/fondoboton"
android:onClick="Jugar"
android:text="@string/Boton"
android:textColor="@android:color/background_light"
android:textSize="24sp" />

    <TextView
android:id="@+id/textView_bestScore"
android:layout_width="match_parent"
android:layout_height="48dp"
android:layout_alignParentStart="true"
android:layout_alignParentLeft="true"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_alignParentBottom="true"
android:layout_marginStart="0dp"           android:layout_marginLeft="0dp"
android:layout_marginEnd="0dp"
android:layout_marginRight="0dp"
android:layout_marginBottom="77dp"
android:background="@drawable/fondoet"
android:contentDescription="@string/app_name"
android:gravity="center"
android:textColor="@color/colorPrimaryDark"
android:textSize="23sp" />
</RelativeLayout>

```

Básicamente los layout contienen archivos .xml y contiene el diseño de la actividad.

- **Values contiene los styles y string.**

```
<resources>
  <string name="app_name">AppKichwa</string> //nombre de la
aplicacion
  <string name="et_nombre">¿Ima shuti Kan?</string> //como te
llamas
  <string name="Boton">PUKLLAY / JUGAR </string> //jugar
  <string name="et_vidas">Kawsay: </string> //Vidas
  <string name="jugador_nombre">Pukllak Shutik</string> //jugador
  <string name="total_score">Apiskanki: 0</string> Puntos
  <string name="Boton_sumar">TARIPANA</string> //comprobar
  <string name="et_respuesta">Kutipay</string> //respuesta
</resources>
```

Con este código conseguimos: Asignar nombres a las variables (Títulos) y reproduce la interfaz para que el usuario seleccione una de las opciones.

- **AndroidMa**

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.frutiapp">

    <application
android:allowBackup="true"          android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/AppTheme">
        <activity
android:name=".Main2Activity_Nivel6"

android:configChanges="orientation|screenSize"></activity>
        <activity
            android:name=".Main2Activity_Nivel5"
            android:configChanges="orientation|screenSize"/>
        <activity
            android:name=".Main2Activity_Nivel4"
            android:configChanges="orientation|screenSize"/>
        <activity
            android:name=".Main2Activity_Nivel3"
            android:configChanges="orientation|screenSize"/>
        <activity
            android:name=".Main2Activity_Nivel2"
            android:configChanges="orientation|screenSize"/>
        <activity
            android:name=".Main2Activity_Nivel1"
            android:configChanges="orientation|screenSize"/>
        <activity android:name=".MainActivity"

```

- **nifest archivo xml.**

```

android:configChanges="orientation|screenSize">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

```

```
</manifest>
```

Que contiene las propiedades, permisos y las actividades a las que se podrán acceder dentro de la aplicación Android.

- **Fragmento del código Java de la programación de las operaciones básicas**

```
package com.example.frutiapp;

import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;

public class Main2Activity_Nivel1 extends AppCompatActivity {

    private TextView tv_nombre, tv_score;
    private ImageView iv_Auno, iv_Ados, iv_vidas;
    private EditText et_respuesta;
    private MediaPlayer mp, mp_great, mp_bad;

    int score, numAleatorio_uno, numAleatorio_dos, resultado, vidas = 3;
    String nombre_jugador, string_score, string_vidas;
    String numero [] = {"cero", "uno", "dos", "tres", "cuatro", "cinco",
    "seis", "siete", "ocho", "nueve"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2_nivel1);
        Toast.makeText(this, "SHUK PAKTA - ICHILLA YAPAYKUNA",
        Toast.LENGTH_LONG).show();//"Nivel 1 - Sumas basicas
```

```

        tv_nombre = (TextView)findViewById(R.id.textView_nombre);
tv_score = (TextView)findViewById(R.id.textView_score);          iv_Auno
= (ImageView)findViewById(R.id.imageView_NumUno);          iv_Ados =
(ImageView)findViewById(R.id.imageView_NumDos);          iv_vidas =
(ImageView)findViewById(R.id.imageView_vidas);          et_respuesta =
(EditText)findViewById(R.id.editText_resultado);
        nombre_jugador =
getIntent().getStringExtra("jugador");
tv_nombre.setText("Pukllak Shutti: " + nombre_jugador);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        mp = MediaPlayer.create(this, R.raw.goats);
mp.start();
        mp.setLooping(true);

        mp_great = MediaPlayer.create(this, R.raw.wonderful);
mp_bad = MediaPlayer.create(this, R.raw.bad);

        NumAleatorio();
    }
    public void Comparar(View view){
        String respuesta = et_respuesta.getText().toString();

        if(!respuesta.equals("")){

            int respuesta_jugador = Integer.parseInt(respuesta);
if(resultado == respuesta_jugador){
mp_great.start();          score++;
            tv_score.setText("Apiskanki: " + score);
            et_respuesta.setText("");
            BaseDeDatos();

            } else {

                mp_bad.start();
vidas--;          BaseDeDatos();
                switch
(vidas){
case 3:
iv_vidas.setImageResource(R.drawable.tresvidas);
break;

```

```
                case
2:
                Toast.makeText(this, "ISHKAY KAWSAYLLAMI SAKIRIN!",
Toast.LENGTH_SHORT).show(); //Te quedan 2 vidas!
                iv_vidas.setImageResource(R.drawable.dosvidas);
break;
                case
1:
                Toast.makeText(this, "SHUK KAWSAYLLAMI SAKIRIN!",
```

```

        Toast.LENGTH_SHORT).show();// te quedan 1 vidas
        iv_vidas.setImageResource(R.drawable.unavida);
break;
        case
0:
        Toast.makeText(this, "TUKUY KANPAK KAWSAYTACHU
CHINKARICHISHKANKI!", Toast.LENGTH_SHORT).show(); //Has perdido todas tus
vidas!
        Intent intent = new Intent(this,
MainActivity.class);
        startActivity(intent);
finish();
mp.release();
        mp.stop();
        break;
    }

    et_respuesta.setText("");
}

    NumAleatorio();

} else {
    Toast.makeText(this, "KUTIPAY KANPA YUYAITA",
Toast.LENGTH_SHORT).show();//escribe tu respuesta
}

}
    public void NumAleatorio(){
if(score <= 9){
    numAleatorio_uno = (int) (Math.random() * 10);
numAleatorio_dos = (int) (Math.random() * 10);

    resultado = numAleatorio_uno + numAleatorio_dos;

    if(resultado <= 10){
        for(int i = 0; i < numero.length; i++){
            int id = getResources().getIdentifier(numero[i],
"drawable", getPackageName());

```

```
                if (numAleatorio_uno == i){
iv_Auno.setImageResource(id);                }
if(numAleatorio_dos == i){
iv_Ados.setImageResource(id);
    }
    }
    } else {
        NumAleatorio();
    }

} else {

    Intent intent = new Intent(this, Main2Activity_Nivel2.class);

    string_score = String.valueOf(score);
```

```

string_vidas = String.valueOf(vidas);
intent.putExtra("jugador", nombre_jugador);
intent.putExtra("score", string_score);
intent.putExtra("vidas", string_vidas);

        startActivity(intent);
finish();                mp.stop();
mp.release();
    }
}
    public void BaseDeDatos(){
        AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this, "BD",
null, 1);
        SQLiteDatabase BD = admin.getWritableDatabase();
        Cursor consulta = BD.rawQuery("select * from puntaje where score =
(select max(score) from puntaje)", null);
if(consulta.moveToFirst){
        String temp_nombre = consulta.getString(0);
String temp_score = consulta.getString(1);

        int bestScore = Integer.parseInt(temp_score);

        if(score > bestScore){
            ContentValues modificacion = new ContentValues();
modificacion.put("nombre", nombre_jugador);
modificacion.put("score", score);

            BD.update("puntaje", modificacion, "score=" + bestScore,
null);
        }

        BD.close();

    } else {

        ContentValues insertar = new ContentValues();
insertar.put("nombre",
nombre_jugador);                insertar.put("score",
score);

        BD.insert("puntaje", null, insertar);
        BD.close();
    }
}
@Override
public void onBackPressed(){

}
}
}

```

Aquí determinamos toda la programación de las operaciones básicas que contiene la aplicación, conexión a diferentes niveles y reglas básicas del juego.

8.3.2 Integraciones Permanentes

La metodología XP recomienda una integración diaria, esto no fue un inconveniente en este proyecto debido que al no poder cumplir con la programación en parejas y ser desarrollado por una sola persona el código siempre estuvo integrado y en la última versión funcional hasta la fecha.

8.4 Fase de Pruebas

Las Pruebas o Test Case son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio. Una vez finalizada la fase de codificación y para la comprobación del correcto funcionamiento de la aplicación en cada uno de sus niveles, se realizarán las pruebas respectivas. Esta etapa es crucial en el proceso porque permite detectar errores y corregirlos.

8.4.1 Pruebas de aceptación

Es una prueba formal que se lleva a cabo para determinar si un sistema se ajusta a sus criterios de aceptación y que permite al usuario determinar si acepta o no el sistema, cada prueba de aceptación trata de probar la funcionalidad de cada una de las historias de usuarios mencionadas en la fase de planeación.

Tabla 17.

Prueba de aceptación, funciones del aplicativo.

PRUEBA DE ACEPTACIÓN	
Casos de Prueba: Funciones	Opción de Prueba: Funciones
Numero de Caso de Prueba: 1	Numero de HU: 1
Nombre de Caso de Prueba: Funciones del aplicativo	
Precondiciones:	
• Dispositivo móvil Android desde versión 4.0	

Datos:

- Abrir la aplicación en el dispositivo móvil

Resultado Esperado:

Si el usuario abre la aplicación notara todas las funciones requeridas al inicio.

Evaluación: Las acciones a realizar son satisfactorias en esta actividad.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 18.

Prueba de aceptación, ingreso al sistema.

PRUEBA DE ACEPTACIÓN

Casos de Prueba: Operabilidad

Opción de Prueba: Operabilidad

Numero de Caso de Prueba: 2

Numero de HU: 2

Nombre de Caso de Prueba: Ingreso al sistema **Precondiciones:**

- Dispositivo móvil Android desde versión 4.0

Datos:

- Abrir la aplicación en el dispositivo móvil

Resultado Esperado:

Si el usuario abre la aplicación notara todas las funciones requeridas al inicio y su pantalla principal a la espera de cualquier acción.

Evaluación: Las acciones a realizar son satisfactorias en esta actividad.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 19.

Prueba de aceptación, diseño del aplicativo.

PRUEBA DE ACEPTACIÓN

Casos de Prueba: Diseño

Opción de Prueba: Diseño

Numero de Caso de Prueba: 3

Numero de HU: 3

Nombre de Caso de Prueba: diseño del aplicativo **Precondiciones:**

- Dispositivo móvil Android desde versión 4.0

Datos:

- Abrir la aplicación en el dispositivo móvil

Resultado Esperado:

Si el usuario abre la aplicación notara que todo el diseño está enfocado para niños de una manera más didáctica y comprensiva.

Evaluación: Las acciones a realizar son satisfactorias en esta actividad.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 20.

Prueba de aceptación, contenido del aplicativo.

PRUEBA DE ACEPTACIÓN

Casos de Prueba: Contenido

Opción de Prueba: Contenido

Numero de Caso de Prueba: 4

Numero de HU: 4

Nombre de Caso de Prueba: diseño del aplicativo **Precondiciones:**

- Dispositivo móvil Android desde versión 4.0

Datos:

- Abrir la aplicación en el dispositivo móvil

Resultado Esperado:

Si el usuario abre la aplicación notara que todo el contenido está adaptado para niños entre edades de 5 a 10 años haciendo que el aprendizaje sea más entretenido y atractivo para ellos.

Evaluación: Las acciones a realizar son satisfactorias en esta actividad.

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 21.

Prueba de aceptación, dificultad del aplicativo.

PRUEBA DE ACEPTACIÓN	
Casos de Prueba: Dificultad	Opción de Prueba: Dificultad
Numero de Caso de Prueba: 5	Numero de HU: 5
Nombre de Caso de Prueba: Dificultad del aplicativo Precondiciones:	
<ul style="list-style-type: none">• Dispositivo móvil Android desde versión 4.0	
Datos:	
<ul style="list-style-type: none">• Abrir la aplicación en el dispositivo móvil	
Resultado Esperado:	
Si el usuario abre la aplicación e inicia el juego las operaciones básicas inician desde la suma y según vaya haciendo aciertos avanzara al siguiente nivel y si no acierta en las respuestas se disminuye un intento.	
Evaluación: Las acciones a realizar son satisfactorias en esta actividad.	

Elaborado propio por: Cerda Fabricio, 2021.

Tabla 22.

Prueba de aceptación, disponibilidad del aplicativo.

PRUEBA DE ACEPTACIÓN	
Casos de Prueba: Disponibilidad	Opción de Prueba: Disponibilidad
Numero de Caso de Prueba: 6	Numero de HU: 6
Nombre de Caso de Prueba: Disponibilidad del aplicativo Precondiciones:	
<ul style="list-style-type: none">• Dispositivo móvil Android desde versión 4.0	
Datos:	
<ul style="list-style-type: none">• Archivo de instalación .apk	
Resultado Esperado:	
Si el usuario instala el archivo apk deberá abrirse sin ningún problema en el dispositivo móvil Android.	
Evaluación: Las acciones a realizar son satisfactorias en esta actividad.	

Elaborado propio por: Cerda Fabricio, 2021.

8.4.2 Pruebas unitarias

Este testing nos permite encontrar errores y bugs durante la fase de desarrollo y antes de llevar nuestro aplicativo a la fase final o de publicación. Se sabe a ciencia cierta, que encontrar y resolver algún bug que se encuentra en la etapa de desarrollo se lo realiza dentro del mismo proceso, se corrige ese método o líneas de código, y se vuelve a pasar la prueba unitaria hasta una vez que corrijamos el bug detectado.

Por lo tanto, al ya tener el aplicativo exportado y haber obtenido resultados satisfactorios con las pruebas de aceptación entre el usuario y el aplicativo damos por finalizada las pruebas unitarias.

I. CONCLUSIONES

Gracias a la entrega de la aplicación móvil Android como herramienta de enseñanza y aprendizaje de operaciones matemáticas para los alumnos de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna cantón Tena de la provincia de Napo, se concluye lo siguiente.

El uso de la Metodología Extreme Programming o XP, fue fundamental porque gracias a ella se pudo elaborar software de calidad, además, se fue cumpliendo con cada una de sus fases permitiendo que podamos mostrar algunas versiones previas al usuario lo que fue de gran ayuda porque se van realizando los correctivos correspondientes para dejar la aplicación totalmente funcional antes de la versión final.

Durante el desarrollo del proyecto de titulación, se pudo evidenciar que los estudiantes de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa", tenían falencias en la realización de ejercicios de suma, resta y multiplicación. Para dar solución a esta problemática, se diseñó una propuesta alternativa sobre la importancia del uso de juegos didácticos para mejorar el proceso de enseñanza – aprendizaje de las operaciones básicas de suma, resta y multiplicación.

Se implementó en dispositivos móviles de los padres de los estudiantes y en su totalidad y de manera efectiva se validó el impacto pedagógico mediante el análisis estadístico y descriptivo de los resultados, para mejorar el proceso enseñanza – aprendizaje de la suma, resta multiplicación de los estudiantes Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa".

J. RECOMENDACIONES

Se recomienda el uso de alguna Metodología Ágil para el desarrollo de una aplicación móvil, así que gracias a la metodología podemos ser más disciplinados a la hora de elaborar o construir algún proyecto de software. Las metodologías ágiles permiten construir un buen equipo de trabajo y nos permite configurar nuestro propio entorno de desarrollo en base a las necesidades, además, la planificación es flexible permitiendo responder a cambios repentinos en el proyecto.

Para la emulación se recomienda usar un dispositivo real ya sea una tablet o un smartphone porque el ADV o Emulador de Android carece de algunas herramientas como GPS, brújula, etc. y sobre todo evitamos consumir recursos de procesamiento de nuestro computador y poder probar la verdadera funcionalidad de la Aplicación en un entorno real.

Se recomienda seguir modificando o mejorando esta aplicación para que sea más completa que esta, ya que al ser una versión inicial carece de algunas funcionalidades como por ejemplo videos, etc.

Es necesario que el docente de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" asigne actividades en esta herramienta e incentive hacer el uso de esta aplicación a los sus alumnos.

Se recomienda para futuros trabajos de investigación, el desarrollo de aplicaciones móviles utilizando realidad aumentada en sectores como la educación, ya que se ha demostrado que tiene influencia positiva ya sea en la afluencia como en otras y es mayor que usando medios tradicionales.

K. BIBLIOGRAFÍA

- Acuña, J. M. (4 de Junio de 2019). *Espacio de Tecnología*. Obtenido de <https://jmacuna.tecnoblog.guru/2017/03/sistemas-operativos-moviles.html>
- Android. (2014). *Developer Android*. Obtenido de <https://developer.android.com/studio>
- Android Studio. (10 de 03 de 2020). *Android Studio*. Obtenido de <https://developer.android.com/studio>
- Android Studio. (10 de 03 de 2021). *Android Studio*. Obtenido de <https://developer.android.com/studio>
- Aparice, R. (2010). *Conectados en el Ciberespacio*. Obtenido de https://books.google.com.ec/books?id=JCB0jleuU_oC&printsec=frontcover&hl=es#v=onepage&q&f=false
- Baña, A. A. (8 de Diciembre de 2016). *Android Studio FAQs*. Obtenido de <https://androidstudiofaqs.com/conceptos/ventajas-desventajas-android-studio>
- Bastar, S. G. (2012). *METODOLOGÍA DE LA INVESTIGACIÓN*. México: RED TERCER MILENIO.
- Beck, K., Cynthia, A., & Gamma, E. (9 de Noviembre de 2000). *Extreme Programming Explained*. Obtenido de <http://ptgmedia.pearsoncmg.com/images/9780321278654/samplepages/9780321278654.pdf>
- Bine, Lep;. (2016). *Herramientas Digitales*. Obtenido de <https://diana769.webnode.mx/fotogaleria/herramientas-digitales/>
- Briones, G. (1997). *Metodología de la investigación cuantitativa en las ciencias sociales*. Bogota, Colombia: Corcas. Obtenido de

<https://metodoinvestigacion.files.wordpress.com/2008/02/metodologia-de-lainvestigacion-guillermo-briones.pdf>

Brito Abundis, C. J. (Diciembre de 2013). *Metodologías para desarrollar software*.

Obtenido de

<http://recibe.cucei.udg.mx/revista/es/vol2no3/pdf/computacion05.pdf>

Bustamante, J. J. (23 de Septiembre de 2016). *LOS SISTEMAS OPERATIVOS PARA*

DISPOSITIVOS MÓVILES Y PARA PC. Obtenido de

<https://lossoparads.wordpress.com/2016/09/23/los-sistemas-operativos-paradispositivos-moviles/>

Campos, A. (3 de 3 de 2014). *Mundo Contac*. Obtenido de

<https://mundocontact.com/las-apps-moviles-potencian-al-sector-turistico/>

Castro, L. (4 de 1 de 2014). *¿Qué es una librería?* Obtenido de

<https://aprendeweb.net/librerias/>

Coditramuntana. (2019). *Arquitectura app*. Obtenido de

<https://coditramuntana.com/es/subservicio/desarrollo-de-aplicaciones-moviles>

Comercio, E. (18 de 06 de 2018). *El Comercio*. Obtenido de

<https://www.elcomercio.com/viajar/apps-ayudan-movilidad-turistadestinos.html>

EcuRed. (2003). *Metodologías de desarrollo de Software*.

Obtenido de

https://www.ecured.cu/Metodologias_de_desarrollo_de_Software

El Telegrafo. (17 de Septiembre de 2019). *Aplicaciones Educativas en el Ecuador*.

Obtenido

de

<https://www.eltelegrafo.com.ec/noticias/tecnologia/1/aplicaciones-educacion>

Escribano, D. P. (2010). *Análisis del desarrollo de aplicaciones móviles multiplataforma*.

- Fernando, T. J. (2016). *IMPLEMENTACIÓN DE APPS EDUCATIVAS PARA DISPOSITIVOS MÓVILES*. Quito.
- González, A. (15 de 4 de 2013). *Glosario ¿Qué es Dalvik?* Obtenido de <http://www.tuprogramacion.com/glosario/que-es-dalvik/>
- González, M. (5 de Junio de 2014). *Colombia Digital*. Obtenido de <https://colombiadigital.net/actualidad/experiencias/item/7167-el-turismo-encolombia-a-traves-de-su-celular.html>
- Hernández Sampieri, R. (2014). *Metodologías de la investigación*. México: McGRAW-HILL / INTERAMERICANA EDITORES, S.A. DE C.V.
- Jiménez, E. (23 de Julio de 2013). *Android Ayuda*. Obtenido de <https://androidayuda.com/2013/07/26/desarrollando-para-android-i-androidstudio/>
- Joskowicz, J. (10 de Febrero de 2008). *Reglas y Prácticas en*. Obtenido de <http://iie.fing.edu.uy/~josej/docs/XP%20-%20Jose%20Joskowicz.pdf>
- Muradas M, Y. (23 de Marzo de 2018). *SQLite para Android: La herramienta definitiva*. Obtenido de <https://openwebinars.net/blog/sqlite-para-android-laherramienta-definitiva/>
- Murrillo, W. (18 de 4 de 2008). *La investigación científica*. Obtenido de Monografias.com: <https://www.monografias.com/trabajos15/investcientifica/invest-cientifica.shtml>
- Navarra. (17 de Septiembre de 2018). *Uso de dispositivos móviles*. Obtenido de Acércate a las TIC: <https://www.navarra.es/NR/rdonlyres/48F9746B-080C-4DEA-BD95-A5B6E01797E1/315641/7Usodedispositivosmoviles.pdf>
- NextPit. (14 de Noviembre de 2012). *Android Pit*. Obtenido de <https://www.androidpit.es/sdk-android>

- Oracle. (16 de Marzo de 2020). *Oracle Technology Network*. Obtenido de <https://www.oracle.com/technetwork/es/java/javase/downloads/index.html>
- Orozco, D. (1 de 6 de 2011). *Definición de android*. Obtenido de <https://conceptodefinicion.de/android/>
- R. R. Vique. (2012). *Métodos para el desarrollo de aplicaciones móviles*. España: Calatuña.
- Sandoval, J., McAnally, L., & Lavigne, G. (2013). *El Teléfono inteligente (smartphone) como herramienta pedagógica*. Obtenido de UDG Virtual: <http://www.udgvirtual.udg.mx/apertura/index.php/apertura/article/view/372/3>
- Segovia, D. (2014). *Análisis diseño e implementación de una aplicación Android que permite el control de iluminación a través de un dispositivo móvil dentro del área administrativa y laboratorios del Liceo 16 de Junio de la ciudad de Machala utilizando la metodología Mobi*. Machala, El Oro, Ecuador.: Tesis de Grado.
- Software de Comunicaciones. (2016). *Programación en dispositivos móviles portables*. Obtenido de <https://sites.google.com/site/swcuc3m/home/android/generalidades/2-Arquitectura-de-android>
- Subirats, J. (16 de 4 de 2014). *Qué es y para qué sirve una .APK*. Obtenido de <http://www.fandroides.com/que-es-y-para-que-sirve-una-apk/>
- Super intendencia de Telecomunicaciones. (2012). Evolución de la Telefonía Móvil en el Ecuador. *Revista Institucional SUPERTEL*, 4, 31, 32.
- Tecnología Fácil. (6 de Mayo de 2018). *Tecnología Fácil*. Obtenido de <https://tecnologia-facil.com/que-es/que-es-android/>

- Thomson Course Technology. (2004). *Diccionario de Informatica E Internet: Computer and Internet Technology Definitions in Spanish*. Canadá: Thomson Learning.
- Universidad Carlos III de Madrid. (5 de Julio de 2012). *oftware de Comunicaciones, Programación en dispositivos móviles portables*. Obtenido de <https://sites.google.com/site/swcuc3m/home/android/generalidades/dalvikvm1>
- Universidad Politécnica de Madrid. (2012). *Doctorado en Ingeniería de Sistemas Telemáticos 20091 Metodología de desarrollo ágil para sistemas móviles Introducción al desarrollo con Android y el iPhone*. Obtenido de http://www.adamwesterski.com/wpcontent/files/docsCursos/Agile_doc_TemasAnv.pdf
- University-Complutense, Diccionarios Oxford. (2002). *Diccionario de Internet*. España: Complutense.
- Vega, S. G. (2015). *Creación de una aplicación para dispositivos móviles que brinde información sobre la historia de los lugares turísticos de la ciudad de Guayaquil*. Guayaquil, Ecuador: Proyecto de Titulación.
- Vico, A. J. (17 de Febrero de 2011). *Arquitectura de Android*. Obtenido de <https://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>
- Wells, D. (2001). *Ciclo de vida de un proyecto XP*. Obtenido de <http://oness.sourceforge.net/proyecto/html/ch05s02.html>
- Wells, D. (Octubre de 8 de 2013). *Extreme Programming*. Obtenido de <http://www.extremeprogramming.org/>
- Zenva Pty Ltd. (2020). *De Idea a App*. Obtenido de <https://deideaaapp.org/arquitecturas-de-aplicaciones-moviles/>
- Zona, M. (15 de Octubre de 2019). *Mercado dominado por iOS y Android mientras Windows Phone retrocede*. Obtenido de Available:

<http://www.movilzona.es/2014/08/15/el-964-del-mercado-dominado-por-iosy-android-mientras-windows-phone/>

K. ANEXOS

Anexo N° 1. Código de clase MainActivity.java

```
package com.example.frutiapp;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase; import
android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import
android.view.View;
import android.view.inputmethod.InputMethodManager;
import android.widget.EditText; import
android.widget.ImageView; import
android.widget.TextView; import
android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    private EditText et_Nombre;
private ImageView iv_personaje;
private TextView tv_bestScore;
private MediaPlayer mp;

    int num_aleatorio = (int) (Math.random() * 10);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentViews(R.layout.activity_main);

        et_Nombre = (EditText)findViewById(R.id.txt_nombre);
        iv_personaje = (ImageView)findViewById(R.id.imageView_Personaje);
tv_bestScore = (TextView)findViewById(R.id.textView_bestScore);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);

        int id;
        if(num_aleatorio == 0 || num_aleatorio == 10){
            id = getResources().getIdentifier("mango", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 1 || num_aleatorio == 9){
id = getResources().getIdentifier("fresa", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 2 || num_aleatorio == 8){
id = getResources().getIdentifier("manzana", "drawable",
getPackageName());
            iv_personaje.setImageResource(id);
        }
    }
}
```

```

        } else if (num_aleatorio == 3 || num_aleatorio == 7){
id = getResources().getIdentifier("sandia", "drawable",
getPackageName());
        iv_personaje.setImageResource(id);
        } else if (num_aleatorio == 4 || num_aleatorio == 5 || num_aleatorio
== 6){
        id = getResources().getIdentifier("uva", "drawable",
getPackageName());
        iv_personaje.setImageResource(id);
        }
AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this, "BD",
null, 1);
        SQLiteDatabase BD = admin.getWritableDatabase();
        Cursor consulta = BD.rawQuery(
        "select * from puntaje where score = (select max(score) from
puntaje)", null);
        if(consulta.moveToFirst()){
                String temp_nombre = consulta.getString(0);
String temp_score = consulta.getString(1);
                tv_bestScore.setText("YALLISKANKI: " + temp_score + " Kan: " +
temp_nombre); //Mejor puntaje
                BD.close();
        } else {
                BD.close();
        }

        mp = MediaPlayer.create(this, R.raw.alphabet_song);
mp.start();
        mp.setLooping(true);

    }
    public void Jugar(View view){
        String nombre = et_Nombre.getText().toString();

        if(!nombre.equals("")){
mp.stop();                mp.release();

                Intent intent = new Intent(this, Main2Activity_Nivel1.class);

                intent.putExtra("jugador", nombre);
startActivity(intent);                finish();

        } else {
                Toast.makeText(this, "KANPA SHUTITA KILLKANA KANKI",
Toast.LENGTH_LONG).show(); //debes escribir tu nombre

                et_Nombre.requestFocus();
                InputMethodManager imm =
(InputMethodManager) getSystemService(this.INPUT_METHOD_SERVICE);
                imm.showSoftInput(et_Nombre, InputMethodManager.SHOW_IMPLICIT);

```

```
    }  
  }  
  @Override  
  public void onBackPressed(){  
  }  
}
```

Anexo N° 2. Código de clase MainActivity_Nivel1.java

```
package com.example.frutiapp;

import android.content.ContentValues;
import android.content.Intent; import
android.database.Cursor;
import android.database.sqlite.SQLiteDatabase; import
android.media.MediaPlayer;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import
android.view.View; import
android.widget.EditText; import
android.widget.ImageView; import
android.widget.TextView; import
android.widget.Toast;

public class Main2Activity_Nivel1 extends AppCompatActivity {

    private TextView tv_nombre, tv_score;
private ImageView iv_Auno, iv_Ados, iv_vidas;
private EditText et_respuesta;
    private MediaPlayer mp, mp_great, mp_bad;

    int score, numAleatorio_uno, numAleatorio_dos, resultado, vidas = 3;
String nombre_jugador, string_score, string_vidas;
    String numero [] = {"cero", "uno", "dos", "tres", "cuatro", "cinco",
"seis", "siete", "ocho", "nueve"};

    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2_nivel1);
        Toast.makeText(this, "SHUK PAKTA - ICHILLA YAPAYKUNA",
Toast.LENGTH_LONG).show();//"Nivel 1 - Sumas basicas

        tv_nombre = (TextView)findViewById(R.id.textView_nombre);
tv_score = (TextView)findViewById(R.id.textView_score);          iv_Auno
= (ImageView)findViewById(R.id.imageView_NumUno);          iv_Ados =
(ImageView)findViewById(R.id.imageView_NumDos);          iv_vidas =
(ImageView)findViewById(R.id.imageView_vidas);          et_respuesta =
(EditText)findViewById(R.id.editText_resultado);
        nombre_jugador = getIntent().getStringExtra("jugador");
tv_nombre.setText("Pukllak Shutu: " + nombre_jugador);

getSupportActionBar().setDisplayHomeAsUpEnabled(true);
getSupportActionBar().setIcon(R.mipmap.ic_launcher);

mp = MediaPlayer.create(this, R.raw.goats);
mp.start();
mp.setLooping(true);
```

```

mp_great = MediaPlayer.create(this, R.raw.wonderful);
mp_bad = MediaPlayer.create(this, R.raw.bad);

    NumAleatorio();
}
public void Comparar(View view){
    String respuesta = et_respuesta.getText().toString();

    if(!respuesta.equals("")){

        int respuesta_jugador = Integer.parseInt(respuesta);
if(resultado == respuesta_jugador){
mp_great.start();                score++;
    tv_score.setText("Apiskanki: " + score);
et_respuesta.setText("");
    BaseDeDatos();

        } else {

            mp_bad.start();
vidas--;                BaseDeDatos();
            switch (vidas){
case 3:
iv_vidas.setImageResource(R.drawable.tresvidas);
break;

                case

2:
                    Toast.makeText(this, "ISHKAY KAWSAYLLAMI SAKIRIN!",
Toast.LENGTH_SHORT).show(); //Te quedan 2 vidas!
iv_vidas.setImageResource(R.drawable.dosvidas);
break;

                        case

1:
                            Toast.makeText(this, "SHUK KAWSAYLLAMI SAKIRIN!",
Toast.LENGTH_SHORT).show();// te quedan 1 vidas
iv_vidas.setImageResource(R.drawable.unavida);
break;

                                case

0:
                                    Toast.makeText(this, "TUKUY KANPAK KAWSAYTACHU
CHINKARICHISHKANKI!", Toast.LENGTH_SHORT).show(); //Has perdido todas tus
vidas!

                                        Intent intent = new Intent(this,
MainActivity.class);
                                        startActivity(intent);
finish();                mp.stop();
mp.release();            break;
                                }
}
}

```

```

        et_respuesta.setText("");
    }

    NumAleatorio();

    } else {
        Toast.makeText(this, "KUTIPAY KANPA YUYAITA",
Toast.LENGTH_SHORT).show();//escribe tu respuesta
    }

}
    public void NumAleatorio(){
if(score <= 9){
        numAleatorio_uno = (int) (Math.random() * 10);
numAleatorio_dos = (int) (Math.random() * 10);

        resultado = numAleatorio_uno + numAleatorio_dos;

        if(resultado <= 10){
            for(int i = 0; i < numero.length; i++){
                int id = getResources().getIdentifier(numero[i],
"drawable", getPackageName());
                if (numAleatorio_uno == i){
iv_Auno.setImageResource(id);
                }
if(numAleatorio_dos == i){
iv_Ados.setImageResource(id);
                }
            }
        } else {
            NumAleatorio();
        }

    } else {
        Intent intent = new Intent(this, Main2Activity_Nivel2.class);
        string_score = String.valueOf(score);
string_vidas = String.valueOf(vidas);
intent.putExtra("jugador", nombre_jugador);
intent.putExtra("score", string_score);
intent.putExtra("vidas", string_vidas);

        startActivity(intent);
finish();
        mp.stop();
mp.release();
    }
}
    public void BaseDeDatos(){
        AdminSQLiteOpenHelper admin = new AdminSQLiteOpenHelper(this, "BD",
null, 1);
        SQLiteDatabase BD = admin.getWritableDatabase();

```

```

Cursor consulta = BD.rawQuery("select * from puntaje where score =
(select max(score) from puntaje)", null);
if(consulta.moveToFirst()){
    String temp_nombre = consulta.getString(0);
String temp_score = consulta.getString(1);

    int bestScore = Integer.parseInt(temp_score);

    if(score > bestScore){
        ContentValues modificacion = new ContentValues();
modificacion.put("nombre", nombre_jugador);
modificacion.put("score", score);

        BD.update("puntaje", modificacion, "score=" + bestScore,
null);
    }

    BD.close();

} else {

    ContentValues insertar = new ContentValues();

    insertar.put("nombre", nombre_jugador);
insertar.put("score", score);

    BD.insert("puntaje", null, insertar);
    BD.close();
}
}
@Override
public void onBackPressed(){

}
}
}

```

Anexo N° 3. Formato del documento de autorización para el uso de nombre de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa".

Tena, 16 de enero del 2021

**AUTORIZACIÓN PARA EL USO DE NOMBRE DE LA ESCUELA
“PAULINO GREFA”**

Yo, **RUBÉN MAXIMILIANO CERDA MAMALLACTA** con cédula de ciudadanía N° 1500520604 y en calidad de legítimo Director de la Escuela Centro Educativo Cultural Bilingüe "Paulino Grefa" de la parroquia Muyuna del cantón Tena de la provincia de Napo:

Autorizo al estudiante **FABRICIO JOSÉ CERDA GREFA** a hacer uso del material investigativo. La información podrá ser usada para:

- Difusión de actividades y eventos a través de la aplicación móvil en desarrollo.
- Uso interno en proyectos de enseñanza y aprendizaje.

El estudiante se compromete a que la utilización de la información ningún caso supondrá un daño o desprestigio hacia la institución.

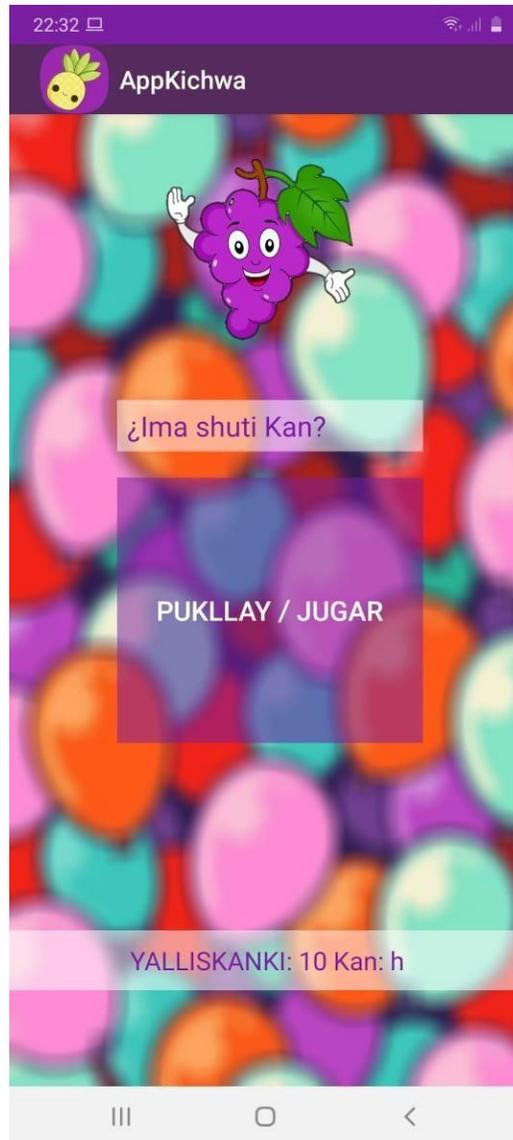
Atentamente.

Mgs. Rubén Maximiliano Cerda Mamallacta
DIRECTOR DE LA ESCUELA
C.I: 1500520604

Anexo N° 4. Capturas de Pantalla en diferentes dispositivos instalados

Imagen 32.

Celular Samsung A31, Android 10.



Elaborado por: Cerda Fabricio, 2021.

Anexo N° 5. Capturas de Pantalla en diferentes dispositivos instalados

*Imagen 32.
Celular Huawei Y6, Android 8.*



Elaborado por: Cerda Fabricio, 2021.