

UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**“APLICACIÓN MÓVIL ORIENTADA A LA EDUCACIÓN
NUTRICIONAL DE ALIMENTOS NATURALES PARA
MOTIVAR EL APRENDIZAJE EN ESTUDIANTES DE
PRIMERO DE SECUNDARIA”**

CASO: UNIDAD EDUCATIVA MARISCAL DE ZEPITA MAÑANA

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: Informática y Comunicaciones**

Postulante: Vanessa Quispe Vadillo

Tutor Metodológico: M. Sc. Lic. Ing. Marisol Arguedas Balladares

Tutor Especialista: Lic. Ing. Sergio Ramiro Rojas Saire

Tutor Revisor: M. Sc. Lic. Patricio Jose Ilaluque Vargas

**EL ALTO – BOLIVIA
2024**

DEDICATORIA

A Dios sobre todas las cosas, por poner fortaleza en mi debilidad, por protegerme y guiarme durante todo mi camino.

A mi madre quien es mi mayor motivación para nunca rendirme en mis estudios.

A mis hermanas, por su comprensión, amor y apoyo incondicional.

AGRADECIMIENTOS

Agradezco a Dios por cuidarme en todas mis acciones día a día, por darme las fuerzas para no rendirme y seguir adelante.

A mi madre, por el apoyo incondicional que siempre me brindo. A pesar de todas las circunstancias, siempre estuvo ahí para apoyarme, siendo la razón por la que no me rendí a mitad del camino. En mis momentos difíciles, sus oraciones me dieron fuerzas para seguir adelante. También agradezco a mis hermanas por el ánimo que me brindaron.

A mis tutores, quienes me guiaron e impulsaron para lograr la elaboración de este proyecto. Les agradezco profundamente por su paciencia, por compartir sus conocimientos, por su tiempo y la confianza depositada en mí durante mi formación. Gracias, queridos ingenieros.

Finalmente, gracias a todos mis amigos, quienes siempre me apoyaron en la realización de este proyecto.

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo, Vanessa Quispe Vadillo, estudiante de Taller de Grado II, de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto, identificada con C.I. 8392331 LP y Registro Universitario: 15000236.

Declaro bajo juramento que:

1. Soy autor del Trabajo de Grado titulado:
“ APLICACIÓN MÓVIL ORIENTADA A LA EDUCACIÓN NUTRICIONAL DE ALIMENTOS NATURALES PARA MOTIVAR EL APRENDIZAJE EN ESTUDIANTES DE PRIMERO DE SECUNDARIA” Caso: Unidad Educativa Mariscal de Zepita Mañana.
El mismo que presento bajo la modalidad de **Proyecto de Grado**
2. El texto de mi proyecto de grado respeta y no vulnera los derechos de terceros, incluidos los derechos de propiedad intelectual. En tal sentido, declaro que este Perfil de Proyecto de Grado no ha sido plagiado total ni parcialmente, para la cual he respetado las normas internacionales de citas y referencias de las fuentes consultadas.
3. El texto del Proyecto de Grado que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico.
4. Por tanto, declaro que mi Proyecto de Grado cumple con todas las normas de la carrera de Ingeniera de Sistemas de la Universidad Pública de El Alto.

El incumplimiento de lo declarado da lugar a responsabilidad del declarante, en consecuencia; a través del presente documento asumo frente a terceros, la carrera de Ingeniera de Sistemas de la Universidad Pública de El Alto toda responsabilidad que pueda derivarse por el Proyecto de Grado presentado.

El Alto, junio de 2024

.....
Vanessa Quispe Vadillo
C.I. 8392331
R.U. 15000236

RESUMEN

En la actualidad, existen numerosas aplicaciones móviles diseñadas para diversas funciones, entre las cuales destacan las aplicaciones educativas. Estas aplicaciones ofrecen un entorno de aprendizaje fuera del aula, fomentando el aprendizaje auto dirigido y facilitando el acceso a información valiosa de manera interactiva. El objetivo principal del proyecto es desarrollar una Aplicación Móvil orientada a la Educación Nutricional, que identifique alimentos naturales mediante Visión Artificial y que proporcione la información nutricional a través de técnicas de gamificación para facilitar un aprendizaje eficiente y atractivo para estudiantes de Primero de Secundaria. Para el desarrollo de la aplicación móvil, se utilizaron: técnicas de visión artificial y técnicas de gamificación. La gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados. Por otro lado, la visión artificial es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender imágenes del mundo real.

Palabras clave: *Aplicación móvil, Educación Nutricional, gamificación, Visión Artificial, Realidad Aumentada.*

SUMMARY

Currently, there are numerous mobile applications designed for various functions, among which educational applications stand out. These apps provide a learning environment outside the classroom, promoting self-directed learning and facilitating interactive access to valuable information. The main objective of the project is to develop a Mobile Application focused on Nutritional Education, capable of identifying natural foods using Artificial Vision and providing nutritional information through gamification techniques to facilitate efficient and engaging learning for first-year secondary school students. For the development of the mobile application, artificial vision techniques and gamification techniques were employed. Gamification is a learning technique that applies game mechanics to the educational-professional sphere to achieve better results. On the other hand, artificial vision is a scientific discipline that encompasses methods for acquiring, processing, analyzing, and understanding images from the real world.

Keywords: *Mobile Application, Nutritional Education, gamification, Artificial Vision, Augmented Reality.*

INDICE DE CONTENIDO

1. MARCO PRELIMINAR.....	1
1.1. INTRODUCCIÓN.....	1
1.2. ANTECEDENTES.....	2
1.2.1. Antecedentes Institucionales	2
1.2.1.1. Misión.....	3
1.2.1.2. Visión.	3
1.2.1.3. Objetivo.	3
1.2.1.4. Organigrama.....	4
1.2.2. Antecedentes Internacionales.....	5
1.2.3. Antecedentes Nacionales	5
1.2.4. Antecedentes Locales	6
1.3. PLANTEAMIENTO DEL PROBLEMA.....	7
1.3.1. Problema General	7
1.3.2. Problemas Específicos	8
1.3.3. Formulación del problema	8
1.4. OBJETIVOS	8
1.4.1. Objetivo General.....	8
1.4.2. Objetivos Específicos	8
1.5. JUSTIFICACIÓN	9
1.5.1. Justificación Técnica.....	9
1.5.2. Justificación Económica	9
1.5.3. Justificación Social	10
1.6. METODOLOGÍA.....	10
1.6.1. Metodología de desarrollo	10
1.6.2. Métrica de calidad al software	11
1.6.3. Costo.....	12
1.6.4. Seguridad	12
1.6.5. Pruebas al software	12
1.6.5.1. Caja Blanca.	13
1.6.5.2. Caja Negra.	13
1.6.6. Otras metodologías	13
1.7. HERRAMIENTAS	13
1.8. LIMITES Y ALCANCES	14

1.8.1.	Límites.....	14
1.8.2.	Alcances.....	15
1.9.	APORTES	15
2.	MARCO TEÓRICO	18
2.1.	INTRODUCCIÓN.....	18
2.2.	APLICACIÓN MÓVIL.....	18
2.2.1.	Sistemas Operativos Móviles.....	18
2.2.2.	Tienda de Aplicaciones.....	19
2.2.3.	Diferencia Entre Aplicaciones Móviles y sitios web móviles	20
2.2.4.	Aplicaciones móviles en el ámbito educativo	20
2.2.5.	Aprendizaje móvil (M-Learning)	20
2.3.	EDUCACIÓN NUTRICIONAL	21
2.4.	ALIMENTOS NATURALES.....	22
2.5.	ALIMENTACIÓN SALUDABLE	22
2.6.	IMPORTANCIA DE ALIMENTACIÓN	22
2.7.	ALIMENTO Y NUTRICIÓN	23
2.8.	INFORMACIÓN NUTRICIONAL	24
2.8.1.	Energía.....	24
2.8.2.	Nutrientes	24
2.8.2.1.	Los carbohidratos.....	25
2.8.2.2.	Las proteínas.....	25
2.8.2.3.	Las Grasas o lípidos.....	26
2.8.2.4.	Las vitaminas.....	26
2.8.2.5.	Minerales.....	26
2.9.	CLASIFICACIÓN DE LOS ALIMENTOS.....	27
2.9.1.	Clasificación de los Alimentos Según sus Nutrientes.....	27
2.9.2.	Clasificación de los Alimentos Según su Función	28
2.9.3.	Clasificación de los Alimentos Según su Origen	29
2.10.	VISIÓN ARTIFICIAL.....	29
2.10.1.	Campos de la Visión Artificial.....	29
2.10.2.	Tareas de la Visión Artificial	30
2.10.2.1.	Clasificación de imágenes.....	30
2.10.2.2.	Detección de objetos.....	30
2.10.2.3.	Seguimiento.....	30
2.10.3.	Aprendizaje Automático (Machine Learning).....	31
2.10.3.1.	Aprendizaje Automático supervisado.....	31

2.10.3.2.	Aprendizaje Automático no Supervisado.	32
2.10.3.3.	Aprendizaje Automático semi supervisado.	32
2.10.3.4.	Aprendizaje por refuerzo.	33
2.10.4.	Aprendizaje por transferencia (TL)	33
2.10.5.	Arquitectura de modelo o algoritmo EfficientDet-Lite.....	34
2.10.6.	Biblioteca Model Maker	35
2.11.	REALIDAD AUMENTADA	36
2.11.1.	Elementos de la realidad aumentada	37
2.11.2.	Niveles de la realidad aumentada:	38
2.12.	METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE: MOBILE-D.....	38
2.12.1.	Exploración	38
2.12.2.	Iniciación	39
2.12.3.	Producción	39
2.12.4.	Estabilización	39
2.12.5.	Pruebas	40
2.13.	FASES PARA CREAR UN MODELO DE MACHINE LEARNING (CRISP – DM) 40	
2.13.1.	Comprensión del negocio.....	40
2.13.2.	Comprensión de los datos.....	41
2.13.3.	Preparación de los datos.....	41
2.13.4.	Modelado	41
2.13.5.	Evaluación	42
2.13.6.	Despliegue	42
2.14.	ISO 25010.....	43
2.15.	COCOMO II	43
2.16.	ISO 27034.....	44
2.17.	PRUEBAS DE SOFTWARE	44
2.17.1.	Pruebas de Caja Blanca o Pruebas estructurales	45
2.17.2.	Pruebas de Caja Negra o Pruebas Funcionales.....	46
2.18.	HERRAMIENTAS.....	47
2.18.1.	Google Colaboratory	47
2.18.2.	Android Studio	47
2.18.3.	Librería TensorFlow	48
2.18.4.	Librería Numpy	48
2.18.5.	Python.....	48
2.18.6.	Java	49
2.18.7.	Kotlin.....	49

2.18.8.	Roboflow	50
3.	MARCO APLICATIVO	52
3.1.	INTRODUCCIÓN.....	52
3.2.	DESARROLLO DEL MODELO DE RECONOCIMIENTO CON CRISP-DM	54
3.2.1.	Colectar y entender datos.....	54
3.2.2.	Preparar datos.....	55
3.2.3.	Elegir modelo.....	57
3.2.4.	Entrenar modelo	58
3.2.5.	Evaluar modelo.....	59
3.3.	DESARROLLO DE LA METODOLOGÍA DE SOFTWARE.....	60
3.3.1.	Exploración.....	60
3.3.1.1.	Recopilación de la información.	60
3.3.1.2.	Establecimiento de actores.....	63
3.3.1.3.	Definición del alcance.....	64
3.3.1.4.	Especificación de requisitos.....	64
3.3.2.	Inicialización	66
3.3.2.1.	Establecimiento de recursos.....	66
3.3.2.2.	Software requerido.	66
3.3.3.	Producción	67
3.3.3.1.	Iteración 1.....	67
3.3.3.2.	Iteración 2.....	69
3.3.3.3.	Iteración 3.....	72
3.3.3.4.	Iteración 4.....	76
3.3.3.5.	Iteración 5.....	79
3.3.3.6.	Iteración 6.....	84
3.3.3.7.	Iteración 7.....	87
3.3.3.8.	Iteración 8.....	92
3.3.4.	Estabilización.....	95
3.3.5.	Pruebas	96
3.3.5.1.	Actividades en la fase de pruebas:	96
4.	CALIDAD, COSTO, SEGURIDAD Y PRUEBAS	99
4.1.	MÉTRICA DE CALIDAD	99
4.1.1.	Usabilidad.....	99
4.1.2.	Funcionalidad	101
4.1.3.	Confiabilidad.....	102
4.1.4.	Mantenibilidad	103

4.1.5. Portabilidad	104
4.2. ESTIMACIÓN DEL COSTO	105
4.2.1. Cálculo del esfuerzo y tiempo del proyecto.....	107
4.2.2. Costo total del proyecto	108
4.3. SEGURIDAD	108
4.4. PRUEBAS AL SOFTWARE	112
4.4.1. Pruebas de caja blanca	112
4.4.2. Pruebas de caja negra.....	114
4.4.2.1. Módulo cámara del dispositivo para escanear y reconocer alimentos.....	114
4.4.2.2. Módulo de información sobre alimentos.....	116
4.4.2.3. Calculadora de Índice de Masa Corporal (IMC)	117
4.4.3. Pruebas de rendimiento.....	118
4.4.3.1. Pruebas de Rendimiento del Dispositivo.....	119
4.4.3.2. Pruebas de Rendimiento de la Aplicación.....	119
5. CONCLUSIONES Y RECOMENDACIONES	122
5.1. CONCLUSIONES	122
5.2. RECOMENDACIONES.....	123
REFERENCIAS BIBLIOGRÁFICAS.....	124
ANEXOS.....	129

INDICE DE FIGURAS

Figura 1	<i>Organigrama de la Unidad Educativa</i>	4
Figura 2	<i>Cuota de mercado sobre Sistemas Operativos móviles en el mundo</i>	19
Figura 3	<i>La rueda de los alimentos</i>	28
Figura 4	<i>Rendimiento del modelo EfficientDet</i>	35
Figura 5	<i>Ejemplo de realidad aumentada</i>	37
Figura 6	<i>Fases de la metodología Mobile D</i>	40
Figura 7	<i>Esquema del ciclo CRISP-DM estándar</i>	42
Figura 8	<i>Calidad del producto del software</i>	43
Figura 9	<i>Diagrama de Flujo de Datos</i>	53
Figura 10	<i>Open Image DataSet V6</i>	54
Figura 11	<i>Etiquetado de imágenes</i>	56
Figura 12	<i>Formato a la imagen</i>	57
Figura 13	<i>Arquitectura de EfficientDet</i>	58
Figura 14	<i>Entrenamiento del modelo en Collaboratory</i>	59
Figura 15	<i>Verificación del modelo con imágenes de prueba</i>	60
Figura 16	<i>Clase MainActivity.kt (interfaz principal)</i>	68
Figura 17	<i>Interfaz principal</i>	69
Figura 18	<i>Clase ObjectDetectorHelper.kt (donde se integrará el modelo)</i>	71
Figura 19	<i>Pantalla que usa la cámara del dispositivo</i>	72
Figura 20	<i>Pantalla de navegación por Grupos de Alimentos</i>	74
Figura 21	<i>Pantalla donde se visualizan los alimentos pertenecientes al grupo seleccionado</i>	75
Figura 22	<i>Beneficios del alimento seleccionado</i>	77
Figura 23	<i>Tabla nutricional del alimento seleccionado</i>	78
Figura 24	<i>Pantalla juego 1</i>	81
Figura 25	<i>Funcionamiento del juego</i>	82
Figura 26	<i>Visualización de información de alimento al finalizar cada juego</i>	83
Figura 27	<i>Pantalla juego 2</i>	85
Figura 28	<i>Funcionamiento del juego 2</i>	86
Figura 29	<i>Pruebas de conocimiento</i>	89
Figura 30	<i>Funcionamiento juego 3</i>	90
Figura 31	<i>Algunos conceptos básicos sobre nutrición</i>	91
Figura 32	<i>Interfaz para el cálculo del IMC</i>	93

Figura 33 <i>Recomendaciones de acuerdo a la categoría de peso</i>	94
Figura 34 <i>Cálculo de líneas de código con Statistic</i>	106
Figura 35 <i>Configuración en el archivo build.gradle</i>	110
Figura 36 <i>Método para el cálculo del IMC</i>	112
Figura 37 <i>Pruebas unitarias para el módulo de cálculo del Índice de Masa Corporal utilizando Junit</i>	113

INDICE DE TABLAS

Tabla 1 <i>Personal de la Unidad Educativa</i>	2
Tabla 2 <i>Clases que se usaran para el modelo</i>	54
Tabla 3 <i>Requisitos generales para la imagen</i>	55
Tabla 4 <i>Requisitos de etiquetas</i>	55
Tabla 5 <i>Resultados de los cuestionarios</i>	62
Tabla 6 <i>Usuarios</i>	63
Tabla 7 <i>Establecimiento de roles y responsabilidades</i>	64
Tabla 8 <i>Requisitos funcionales:</i>	65
Tabla 9 <i>Requerimientos no funcionales</i>	66
Tabla 10 <i>Planificación Iteración 1</i>	67
Tabla 11 <i>Planificación Iteración 2</i>	70
Tabla 12 <i>Planificación Iteración 3</i>	73
Tabla 13 <i>Planificación Iteración 4</i>	76
Tabla 14 <i>Planificación Iteración 5</i>	79
Tabla 15 <i>Planificación Iteración 6</i>	84
Tabla 16 <i>Planificación Iteración 7</i>	87
Tabla 17 <i>Planificación Iteración 8</i>	92
Tabla 18 <i>Prueba de compatibilidad</i>	97
Tabla 19 <i>Resultados encuesta Usabilidad</i>	100
Tabla 20 <i>Requisitos funcionales cumplidos que se han implementado</i>	101
Tabla 21 <i>Resultados de las pruebas de portabilidad de la aplicación</i>	105
Tabla 22 <i>Tabla de coeficientes</i>	106
Tabla 23 <i>Resultados del escaneo de frutas y verduras</i>	115
Tabla 24 <i>Resultados sobre el módulo Información de Alimentos</i>	117
Tabla 25 <i>Resultados para el Caso de Prueba 2</i>	118
Tabla 26 <i>Tiempo de arranque</i>	119
Tabla 27 <i>Uso de Memoria</i>	119
Tabla 28 <i>Tiempo de respuesta y estabilidad de la ampliación móvil</i>	119

INDICE DE ECUACIONES

Ecuación 1 <i>Índice de Madurez de Software</i>	103
Ecuación 2 <i>Fórmula que utiliza COCOMO II</i>	105
Ecuación 3 <i>Ecuación para calcular tamaño en KLOC</i>	106
Ecuación 4 <i>Esfuerzo expresado en personas por mes</i>	107
Ecuación 5 <i>Esfuerzo persona por mes</i>	107
Ecuación 6 <i>Tiempo de desarrollo en meses</i>	107
Ecuación 7 <i>Cantidad de personal requerido para el desarrollo</i>	108
Ecuación 8 <i>Costo del proyecto por mes</i>	108
Ecuación 9 <i>Costo total del proyecto</i>	108

CAPÍTULO I

1. MARCO PRELIMINAR

1.1. INTRODUCCIÓN

Con el paso de los años, los dispositivos móviles se han convertido en algo indispensable en nuestra vida cotidiana hasta el punto en que algunas tareas nos resultan complicado sin esta tecnología. Al cobrar importancia los dispositivos móviles llegaron las aplicaciones móviles, los cuales nos ofrecen cualquier servicio sin necesidad de salir de casa. Muchos de los servicios y productos que ofrecen son del ámbito de la comunicación, entretenimiento, transporte, educación, etc.

La Inteligencia Artificial (IA) tiene la capacidad de hacer frente a uno de los mayores desafíos que afronta hoy en día la educación, pues provee elementos de análisis para el diagnóstico cognitivo y personalizado de cada alumno, estilos de aprendizaje, desempeño y motivaciones. La Gamificación es una técnica de aprendizaje que aplica elementos y mecánicas de los juegos en contextos educativos y profesionales para mejorar la adquisición de conocimientos, el desarrollo de habilidades y la motivación de los usuarios.

La combinación de gamificación e inteligencia artificial ofrece experiencias educativas más innovadoras y eficaces, haciendo el aprendizaje más divertido y atractivo, proporcionando retroalimentación inmediata y precisa, favoreciendo la retención y aplicación de conocimientos, equipando a los docentes con mejores herramientas didácticas, entre otros.

La infancia y la adolescencia son dos periodos muy importantes para el aprendizaje, en estas etapas se van adquiriendo conocimientos y habilidades que en un futuro serán la base de nuestro estilo de vida. Un tema muy importante como es la educación nutricional, muchas veces queda fuera de la educación que se imparte en los centros educativos, asimismo, se sabe que existe una relación directa entre nuestra alimentación y nuestro estado de salud, por lo mismo la educación nutricional es un tema muy necesario en esta etapa de aprendizaje.

Por lo anterior se plantea la realización de este proyecto, el cual propone una aplicación móvil orientada a la Educación Nutricional que enseñe a los adolescentes la importancia de comer alimentos saludables y de evitar algunos otros no tan saludables. Esta aplicación utilizará la cámara del dispositivo para que sea capaz de identificar y proporcionar la información sobre el valor nutricional de los alimentos, ayudando así en el proceso del aprendizaje en los adolescentes. En este sentido se utilizará la visión artificial y técnicas de gamificación con el fin de conseguir mejores resultados.

Las metodologías que se utilizarán para el desarrollo de este proyecto son Mobile-D para la creación de la aplicación y CRISP-DM para el desarrollo del modelo de reconocimiento. Entre las herramientas que se emplearán se encuentran: Android Studio, Kotlin, Python, TensorFlow y Google Colaboratory.

1.2. ANTECEDENTES

1.2.1. *Antecedentes Institucionales*

La Unidad educativa, es dependiente a la Dirección Distrital de El Alto - 1. Fue creada el 13 de junio de 1988 con R.M. 873/2013. Actualmente oferta el nivel de Educación Secundaria en dos paralelos.

Tabla 1

Personal de la Unidad Educativa

PERSONAL	CANTIDAD
DIRECTOR	1
SECRETARIA	1
REGENTE	1
CONSERJE	1
DOCENTES	19

Los docentes cuentan con grado de Licenciatura dentro del marco del Modelo Educativo Socio comunitario Productivo. Además, la Institución Educativa cuenta con la organización de padres de familia representada por la Junta Escolar y delegados de Curso y la organización de estudiantes representada por el Centro de Estudiantes.

1.2.1.1. Misión. Lograr una educación integral de señoritas y jóvenes de la Unidad Educativa Técnico Humanístico “Mariscal de Zepita Mañana”, a partir de un proceso educativo dinámico, significativo y participativo rescatando valores de solidaridad y cooperación, teniendo una educación de calidad y con equidad.

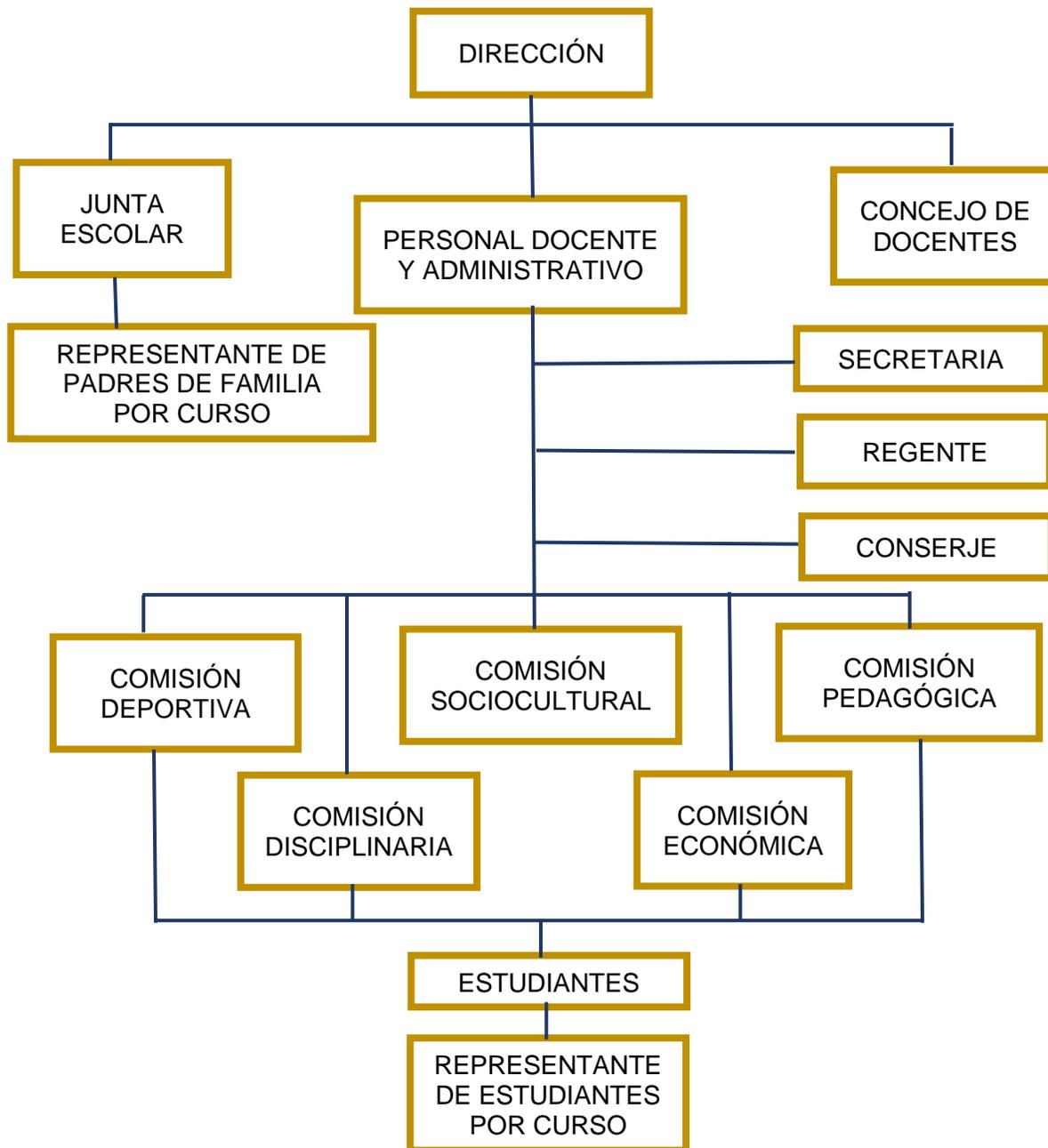
1.2.1.2. Visión. Los estudiantes construyen sus aprendizajes intra e intercultural con una autoestima equilibrada crítica y reflexiva con investigadores y buenos lectores, productores de texto, formuladores de teoremas para la construcción de herramientas. Maestras/os comprometidos con el enfoque pedagógico establecidos en los nuevos fundamentos curriculares.

1.2.1.3. Objetivo. Promover y fortalecer la práctica de los valores socio-comunitarios, creando espacios de interrelación y dialogo reflexivo en el contexto académico, espiritual, social y cultural, fortaleciendo nuestro carácter personal y social en comunidad, asumiendo una nueva forma de ver y sentir la vida con respeto a la Madre Tierra.

1.2.1.4. Organigrama.

Figura 1

Organigrama de la Unidad Educativa



Nota. Tomado de U. E. Mariscal de Zepita (2024).

1.2.2. Antecedentes Internacionales.

- (CIRAD et al., 2024), “*PLANTNET*” [Aplicación móvil]; esta aplicación permite identificar todo tipo de plantas tomando una fotografía de la especie silvestre u ornamental y luego subirla a la aplicación, si la planta ya ha sido catalogada y en la base de datos hay imágenes de esa especie la identificación será en segundos. Actualmente esta aplicación alberga información de más de 1147 especies pertenecientes al Océano Índico, 6416 en Europa Occidental y 938 especies en América del Sur. La aplicación está disponible en Android e IOS y no tiene ningún costo.
- (Mefood, 2020), “*Oorenji & Ciencia y Nutrición*” [Aplicación móvil]; esta es una aplicación sobre nutrición que tiene como objetivo facilitar el seguimiento de una dieta saludable, ajustándose a los requerimientos y necesidades de cada usuario en todo momento. Oorenji personaliza el plan de alimentación acorde a sus preferencias y objetivos en solo un minuto gracias a un algoritmo detalladamente definido por expertos de diferentes ámbitos.
- (Elvira y Peñaloza, 2020), “*Impacto de una App Móvil en la Promoción de Hábitos Alimenticios Saludables en estudiantes de Grado Séptimo de dos Instituciones Educativas Públicas*”, trabajo de grado de la Universidad Santander; tiene como objetivo el fomentar a través de un Aplicación móvil el hábito del consumo de alimentos saludables para un adecuado funcionamiento del cuerpo humano en estudiantes del grado séptimo, haciendo de esta manera una clase que despierte el interés de los estudiantes. Para lograr el objetivo como primera medida se plantea un diagnóstico con una serie de preguntas donde se investiga los principales alimentos que consumen a diario y los conocimientos que tienen acerca de las aplicaciones móviles.

1.2.3. Antecedentes Nacionales

- (N. Mamani, 2020), con la tesis de grado titulado “*Videojuego Educativo para Concientizar sobre la Importancia del Reciclaje*” de la Universidad Mayor de San Andrés, Bolivia. El

objetivo de esta investigación es el desarrollar un videojuego educativo en 3D para ayudar a concientizar y motivar la práctica del reciclaje de manera entretenida y divertida.

- (Casas, 2022), con la tesis de grado titulado “*Tutor Inteligente con Realidad Virtual para la Enseñanza de Conceptos y Teorías de la Biología Celular a Jóvenes Estudiantes de 15 a 18 años en las Unidades Educativas de La-Paz Bolivia*” de la Universidad Mayor de San Andrés, propone desarrollar una aplicación basada en la tecnología de la Realidad Virtual (VR) y tutores inteligentes, para dispositivos móviles con sistema Android, con el propósito de poder mejorar el aprendizaje en los estudiantes de nivel secundario de 15 a 18 años, para esta rama de la Biología denominada ‘Biología Celular’ en las unidades educativas de La – Paz Bolivia.
- (Siles, 2020), con su proyecto de grado titulado “*Desarrollo de Aplicación Móvil para Identificar los Rasgos del Rostro y brindar Asesoramiento en Maquillaje, utilizando Inteligencia Artificial*”, de la Universidad Mayor de San Simón, Bolivia. Este proyecto tiene como objetivo principal el desarrollo de una aplicación móvil como una herramienta de asesoramiento personalizado para el uso correcto de productos de belleza, utilizando Inteligencia Artificial.

1.2.4. Antecedentes Locales

- (A. Mamani, 2020), con la tesis de grado titulado “*Tutor Inteligente para el Aprendizaje de Matemáticas en Sexto de Primaria*”, de la Universidad Pública de El Alto, Bolivia. Este trabajo es una investigación pretende fortalecer el conocimiento académico del estudiante de sexto de secundaria a través de un tutor web inteligente para el aprendizaje de matemáticas bajo malla curricular de grado mencionado.
- (Silva, 2020), “*Aplicación de Realidad Aumentada como Herramienta Orientada al Proceso de Enseñanza y Aprendizaje*”, de la Universidad Pública de El Alto, Bolivia. Esta investigación propone el desarrollo de una aplicación con tecnologías como la realidad

aumentada, haciendo que al interactuar con los objetos 3D se pueda motivar el aprendizaje y reforzar los conocimientos de los estudiantes de la carrera de odontología, de esta forma hacerle frente a la enseñanza tradicional y teórica.

- (Mendoza, 2020), con el proyecto de grado titulado “*Tutor Inteligente Móvil para la Enseñanza de la Estática y Dinámica en Estudiantes de Quinto de Secundaria*” de la Universidad Pública de El Alto, Bolivia. Debido a que los estudiantes de quinto de Secundaria presentan ciertas dificultades en la asimilación de contenidos, así como en la resolución de problemas en los temas de Estática y Dinámica, este proyecto plantea desarrollar de un Tutor Inteligente Móvil capaz de colaborar en el aprendizaje que permitirá optimizar la asimilación de conocimientos en un tiempo mínimo y de forma dinámica.

1.3. PLANTEAMIENTO DEL PROBLEMA

1.3.1. Problema General

Según un estudio sobre el estado nutricional de estudiantes de 5 a 18 años en Bolivia por parte del Ministerio de Salud y Deportes (2022), muestra que existen malos hábitos alimenticios, siendo algunos resultados la prevalencia de malnutrición por exceso en escolares de 5 a 18 años que alcanza al 35,6%, conformada por un 21,9% de sobrepeso, 11,1% de obesidad y 2,2% de obesidad severa.

La información de los valores nutricionales de los alimentos no es tan mencionada en la Educación Secundaria, por lo que una gran mayoría de los estudiantes no están informados sobre el valor nutricional de cada alimento como debería ser, siendo que ellos reciben información teórica y con métodos tradicionales, que no les llama la atención, por lo tanto y no están capacitados para seleccionar y consumir alimentos de alto valor nutritivo. Esto es una de las principales causas de trastornos alimenticios que conllevan pérdida de peso y retraso en el desarrollo y muchos que derivan en riesgos en corto o a largo plazo. La educación nutricional debe aplicarse en todas las etapas de la vida, principalmente en la etapa de la

Educación Primaria – Secundaria, porque es en esta etapa donde se forman los hábitos saludables.

1.3.2. Problemas Específicos

- Información teórica y tradicional, en estudiantes acerca de los alimentos naturales y su valor nutricional, lo que ocasiona desinterés sobre el tema por parte de los estudiantes.
- Adquisición de malos hábitos alimenticios por parte de los estudiantes, por el desconocimiento sobre la elección de alimentos naturales de alto valor nutritivo.
- Desequilibrio nutricional debido a los malos hábitos alimenticios, lo que puede provocar problemas de salud, como trastornos de peso y retraso en el desarrollo físico y cognitivo.

1.3.3. Formulación del problema

¿Cómo motivar a los estudiantes de Primero de Secundaria en el aprendizaje sobre la información nutricional de los alimentos naturales de una forma divertida e interactiva, para mejorar los hábitos alimenticios y así disminuir los riesgos de contraer algunas enfermedades relacionadas a la nutrición?

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar una Aplicación Móvil orientada a la Educación Nutricional, que identifique alimentos naturales mediante Visión Artificial y que proporcione la información nutricional a través de Técnicas de Gamificación para facilitar un aprendizaje eficiente y atractivo para estudiantes de Primero de Secundaria.

1.4.2. Objetivos Específicos

- Indagar los requerimientos relacionados el valor nutricional, tales como las vitaminas, fibras, minerales que un organismo humano necesita para mantener un estado nutricional adecuado y prevenir enfermedades.

- Elaborar una interfaz de usuario intuitiva y amigable que permita a los estudiantes de Primero de Secundaria interactuar fácilmente con la aplicación.
- Integrar un sistema de Visión Artificial que permita a los usuarios identificar alimentos naturales utilizando la cámara de sus dispositivos móviles.
- Incorporar características de gamificación para hacer el aprendizaje sobre nutrición más divertida y motivadora para los estudiantes.
- Diseñar funcionalidades de Realidad Aumentada que superpongan información nutricional relevante sobre los alimentos identificados en tiempo real.

1.5. JUSTIFICACIÓN

1.5.1. *Justificación Técnica*

Según el Boletín Estadístico en Telecomunicaciones (BET, 2022), Bolivia alcanzó los 12.10 millones de líneas móviles al cerrar el primer semestre 2022. La mayoría de la población boliviana cuenta con un celular de gama media con Android 8 o superior, lo que garantiza una amplia audiencia para esta aplicación educativa. Su impacto educativo se maximizará debido a la accesibilidad y la popularidad de estas tecnologías.

Para la ejecución y el uso de esta aplicación, se requiere un dispositivo móvil Android con una cámara de video y un Sistema Operativo igual o superior a Android 8.0. Esto es técnicamente factible, ya que la mayoría de la población boliviana ya cuentan con dispositivos que cumplen con estos requisitos, lo que facilita la implementación del proyecto.

1.5.2. *Justificación Económica*

El software que se utilizará en el desarrollo de este proyecto será de la línea del software libre y otras tecnologías Open Source, muchas de estas tecnologías ofrecen licencia gratuita; es decir, no tendrá un costo dado que su uso no tiene un fin comercial. El proyecto estará disponible de forma gratuita para los usuarios finales.

1.5.3. Justificación Social

La justificación social de esta aplicación móvil se puede fundamentar en varios aspectos, tales como:

- Permitir a los estudiantes del nivel secundario tener una herramienta de apoyo dinámico, didáctico e interactivo en el aprendizaje sobre valores nutricionales. La Realidad Aumentada reforzara el aprendizaje e incrementara la motivación por aprender.
- Proporcionar a los profesores un recurso educativo interactivo y atractivo para la enseñanza de los valores nutricionales de alimentos naturales.
- Salud Pública: Promover hábitos alimenticios saludables desde una edad temprana puede contribuir a la prevención de enfermedades relacionadas a la nutrición.
- Educación Integral: Proporcionar una educación nutricional efectiva a través de herramientas innovadoras y tecnológicas ayuda a complementar el currículo educativo tradicional, preparando a los estudiantes para tomar decisiones informadas y saludables sobre su alimentación y nutrición a lo largo de sus vidas.
- Innovación Tecnológica: El desarrollo de aplicaciones móviles con características avanzadas no solo beneficia a los estudiantes en términos de aprendizaje, sino que también fomenta la innovación y el avance tecnológico en el campo de la educación y la salud, posicionando a la institución educativa como líder en la adopción de nuevas herramientas para el beneficio de sus alumnos.

1.6. METODOLOGÍA

1.6.1. Metodología de desarrollo

El presente proyecto se desarrollará bajo el marco de la metodología Mobile-D, ya que está enfocada al desarrollo de software móvil. Mobile-D es una mezcla de otras metodologías conocidas como son: Extreme Programming, Crystal Methodologies y Rational Unified

Process. Según Amaya (2013, p. 8), se compone de las siguientes fases: exploración, inicialización, producción, estabilización y pruebas.

- **Exploración.** Se dedica al establecimiento de un plan de proyecto y los conceptos básicos, por lo que es posible separarlo del ciclo de desarrollo principal, aunque no es recomendable.
- **Inicialización.** Se prepara e identifica todos los recursos necesarios, los planes para las siguientes fases y se establece el entorno técnico.
- **Producción.** Se repite iterativamente las subbases, el propósito de esta fase es implementar la funcionalidad requerida en el producto mediante la aplicación del ciclo de desarrollo iterativo e incremental.
- **Estabilización.** Se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funcione correctamente.
- **Pruebas.** Tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema.

1.6.2. Métrica de calidad al software

Para las métricas de calidad se utilizara la norma ISO/IEC 25010, también conocida como SQuaRE (Software product Quality Requirements and Evaluation), es un estándar internacional que define un modelo de calidad del producto de software y proporciona un marco estándar para definir, evaluar y especificar los requerimientos de calidad del software, lo que permite a las organizaciones asegurar que los productos de software cumplan con los estándares de calidad esperados y satisfagan las necesidades y expectativas de los usuarios.

Esta norma define ocho características de calidad de software, los cuales son: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, seguridad y compatibilidad (NormasISO.org, 2023).

1.6.3. Costo

Para la estimación de costo del presente proyecto se utilizará COCOMO II (Constructive Cost Model), el cual permite realizar estimaciones en función del tamaño del software y de un conjunto de factores de coste y escala. COCOMO pretende establecer una relación matemática con la que posible estima el esfuerzo (hombre – mes) y el tiempo requerido para desarrollar un proyecto. El dato clave para hacer las estimaciones es el número de líneas de código fuente (Gómez et al., s. f.).

1.6.4. Seguridad

La norma ISO/IEC 27034 es una norma internacional que aborda la seguridad en el desarrollo de aplicaciones de software. Esta norma proporciona un marco integral para integrar la seguridad en todas las etapas del ciclo de vida del desarrollo de aplicaciones, desde la concepción hasta la operación y el mantenimiento.

Algunos de los aspectos clave de ISO/IEC 27034 incluyen: enfoque en el ciclo de vida del desarrollo de aplicaciones, proceso de gestión de riesgos, aplicación de controles de seguridad, consideraciones de seguridad en el diseño y la arquitectura, pruebas de seguridad (NormasISO.org, 2023).

En resumen, ISO/IEC 27034 proporciona un marco completo para integrar la seguridad en el proceso de desarrollo de aplicaciones, lo que ayuda a garantizar que las aplicaciones sean más seguras y resistentes a las amenazas de seguridad.

1.6.5. Pruebas al software

La prueba de software es el proceso de verificación y evaluación sobre lo que un software o producto debería hacer y cumplir. En el campo del desarrollo de software existen diferentes técnicas de pruebas de software cada una con sus propias particularidades. Las pruebas de software que se utilizarán en esta aplicación serán las pruebas de caja blanca y caja negra.

1.6.5.1. Caja Blanca. Las pruebas de caja blanca, también conocidas como pruebas estructurales o pruebas basadas en la lógica interna de un programa tienen como propósito comprobar la calidad e integridad del código del software (Tripleten, 2024).

1.6.5.2. Caja Negra. El objetivo de las pruebas de cajas negras es verificar que el software funciona como espera el usuario final, sin importar lo referente a la estructura del código interno (Gomez, 2023).

1.6.6. Otras metodologías

Según Ruedas (2019), para realizar el algoritmo de Inteligencia Artificial se debe seguir los siguientes pasos genéricos basados en el estándar CRISP-DM:

- **Colectar Datos.** Se debe investigar y obtener los datos.
- **Preparar los datos.** También se le conoce como la limpieza de datos, el formateo.
- **Elegir el modelo.** Se debe elegir el modelo de acuerdo al objetivo que se tenga: algoritmos de clasificación, predicción, deep learning, etc.
- **Entrenar el modelo.** Esta etapa está dedicada al entrenamiento del modelo a partir de los datos de entrenamiento. Los parámetros se ajustan automáticamente por el algoritmo seleccionado a medida que se entrena el modelo.
- **Evaluación del modelo.** Se debe verificar el modelo mediante la introducción de los datos de test, que son datos que la maquina aun no conoce.

1.7. HERRAMIENTAS

- **Google Colaboratory.** Es un producto de Google Research, permite escribir y ejecutar código python desde el navegador. Google Colaboratory hace que la ciencia de datos, aprendizaje automático y el aprendizaje profundo sean accesible para investigadores, estudiantes que no puedan pagar una infraestructura computacional costosa (Google, 2024).

Roboflow. Roboflow es un marco de desarrollo de Visión por Computadora para

recopilación de datos, preprocesamiento y técnicas de entrenamiento de modelos (Bhattacharyya, 2020). Esta herramienta tiene conjunto de datos públicos disponibles para los usuarios, también pueden crear sus propios conjuntos de datos personalizados. Roboflow acepta varios tipos de anotación, para el entrenamiento de modelos dispone de una amplia variedad de bibliotecas de modelos, tales como: EfficientNet, MobileNet, Pytorch, etc.

- **Python.** Es un lenguaje de programación de código abierto de alto nivel interpretado, multiparadigma soportando la orientación a objetos, programación imperativa y programación funcional, además es multiplataforma. Python es ampliamente usado en el desarrollo de software, ciencia de datos y machine learning (Caminiti, 2021).
- **Kotlin.** Es un lenguaje de programación moderno y orientado a objetos, diseñado por JetBrains, una empresa de software con sede en Rusia. Fue lanzado en 2011 y se utiliza principalmente para desarrollar aplicaciones móviles y de escritorio, así como para crear microservicios y aplicaciones web (Wikipedia, 2024).
- **Tensorflow.** Es una plataforma y librería de código abierto para el aprendizaje automático, aprendizaje profundo y computación numérica, es ampliamente usado a nivel mundial y cuenta con una gran cantidad de colaboradores. TensorFlow es una gran plataforma para construir y entrenar redes neuronales, que permiten detectar y descifrar patrones y correlaciones, análogos al aprendizaje (Gazca, 2018).
- **Android Studio.** Es el entorno de desarrollo integrado (IDE) oficial de la plataforma de Android. Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la licencia Apache 2.0 (Android Studio, 2024).

1.8. LIMITES Y ALCANCES

1.8.1. Límites

- Esta aplicación se limitará a dispositivos móviles Android, con versión 8.0 en adelante.

- Esta aplicación va directamente dirigido a estudiantes de Primero de Secundaria (12 y 15 años).
- Esta aplicación móvil, al utilizar la cámara del dispositivo, solo reconocerá frutas y verduras.

La aplicación móvil no podrá:

- Reconocer alimentos compuestos (conformado por diversas sustancias alimenticias), como son: galletas, panes, cereales, tortas, yogurt, etc.
- Reconocer un conjunto de sustancias alimenticias procesadas o preparadas, como son: ensaladas, jugos, arroz con pollo, hamburguesa, etc.
- Estará disponible solo en idioma español.

1.8.2. Alcances

La Aplicación móvil deberá:

- Reconocer alimentos naturales, tales como frutas, verduras y hortalizas, en tiempo real.
- Desplegar en texto la información nutricional del alimento reconocido: proteínas, minerales, vitaminas, grasas.
- Informar sobre las ventajas y tabla nutricional de los alimentos naturales.
- incorporar un formulario o secuencia de preguntas que el usuario deberá completar una vez al día. Este formulario evaluará el progreso y el aprendizaje del usuario respecto a los conocimientos nutricionales.
- La aplicación ofrecerá recomendaciones personalizadas basadas en el Índice de Masa Corporal (IMC) del usuario.

1.9. APORTES

- Una herramienta útil para contribuir a la Educación Nutricional de forma didáctica, dinámica e interactiva en el aprendizaje sobre los nutrientes de los alimentos para estudiantes de Primero de Secundaria.

- Esta investigación se basa en la aplicación de tecnologías como la Visión Artificial y Realidad Aumentada, generando conocimientos para mejorar el aprendizaje sobre Educación Nutricional en los niños y mejorar la calidad de vida.

CAPÍTULO II

2. MARCO TEÓRICO

2.1. INTRODUCCIÓN

En el presente capítulo se describirán los fundamentos teóricos, técnicas, herramientas y metodologías utilizados en el desarrollo de este proyecto, para facilitar la comprensión del trabajo.

2.2. APLICACIÓN MÓVIL

Una aplicación móvil “es un tipo de aplicación diseñada para ejecutarse en un dispositivo móvil, que puede ser un teléfono inteligente o una tableta. Incluso si las aplicaciones suelen ser pequeñas unidades de software con funciones limitadas, se las arreglan para proporcionar a los usuarios servicios y experiencias de calidad” (Herazo, 2022).

Las aplicaciones móviles son programas informáticos desarrollados en diversos lenguajes de programación específicamente diseñados para su uso en dispositivos como Smartphone y tabletas. También se les conoce como apps, abreviatura de la palabra "aplicación" en inglés. Su atributo distintivo radica en su utilidad, dinamismo y facilidad de instalación (Quiroz, 2022).

2.2.1. *Sistemas Operativos Móviles*

Android e iOS son los sistemas operativos más utilizados en el mundo de la telefonía móvil y representan casi la totalidad de los dispositivos móviles lanzados en los últimos años. Hasta el marzo del 2023, Android es el sistema operativo líder a nivel mundial, con un 71% en el mercado, mientras que iOS representa el 28% (Mena, 2023).

Figura 2

Cuota de mercado sobre Sistemas Operativos móviles en el mundo



Nota. Adaptado de *El mapa mundial de Android e iOS*, por Mena, 2023.

2.2.2. Tienda de Aplicaciones

Algunas de las tiendas de aplicaciones móviles más destacadas son:

- Google Play desarrollada por Google Inc.
- App Store de Apple
- Windows Phone Store de Microsoft
- BlackBerry World
- Amazon AppStore

2.2.3. Diferencia Entre Aplicaciones Móviles y sitios web móviles

Una aplicación móvil se refiere a programas que se instalan en el software del dispositivo haciendo que se encuentren allí de forma permanente y pueda usarse de forma continua. Pueden o no requerir de conexión a datos para su uso dependiendo de la naturaleza de las mismas, pero no desaparecerán del móvil aún y cuando no se tenga conexión a internet. Un sitio web móvil, se refiere a una página que ha sido creada bajo una modalidad responsiva y puede visualizarse y ajustarse a las pantallas de los mismos. En otras palabras, simplemente se coloca la url de la página web que se desea buscar desde un teléfono inteligente y en cuanto aparezca esta se adaptara al espacio de visión, sin tener algún programa instalado (Softcorp, 2019).

2.2.4. Aplicaciones móviles en el ámbito educativo

La incorporación de aplicaciones móviles en la educación, han posibilitado un nuevo enfoque del aprendizaje por internet (e-learning), conocido como m-learning o aprendizaje móvil (móvil learning).

El m-learning o aprendizaje móvil se habla de algo espontaneo, informal, personal, aprendizaje situado, situaciones reales, etc., el m-learning se basa en instrucciones, utilizando audio vos, imágenes en movimiento y aprendizaje de campo (Herrera Bernal, 2008, p. 46).

Una aplicación móvil educativa es un programa multimedia diseñado para ejecutarse en un teléfono inteligente, computadoras u otro dispositivo electrónico y usado como una herramienta m-learning.

2.2.5. Aprendizaje móvil (M-Learning)

“El aprendizaje móvil es la capacidad de obtener o proporcionar contenidos educativos en dispositivos de bolsillo personales tales como PDAs y teléfonos inteligentes” (ATLANTE, 2016).

Las características del aprendizaje móvil son:

- Micro lecciones.
- Trasciende el espacio de aprendizaje formal y brinda una estructura de aprendizaje informal.
- Es una forma de apoyo rendimiento.

Las aplicaciones móviles educativas hacen más atractiva el estudio, ya que permiten a los estudiantes aprender de forma innovadora y mucho más divertida

Algunas de las ventajas de las aplicaciones móviles son:

- Permiten el aprendizaje en cualquier contexto, fuera o dentro del aula.
- Las aplicaciones móviles suelen contar con un componente lúdico, integran la dinámica típica del juego y recompensa para conseguir los objetivos de aprendizaje.
- Mejora la atención por parte de los alumnos, ya que las aplicaciones móviles contienen contenido multimedia como videos imágenes, audio, etc., esto hace que sea más atractivo para los alumnos.

2.3. EDUCACIÓN NUTRICIONAL

“La educación nutricional se refiere a la difusión de información útil acerca de los alimentos y sus nutrientes, pero su función no se reduce a ello, sino también a ofrecer las herramientas necesarias para mejorar los hábitos de alimentación y en consecuencia los procesos de nutrición” (Universidad Internacional de Valencia, 2021).

Según el Comité Municipal de Seguridad Alimentaria de Sucre (2021), la Educación Nutricional representa un proceso de incorporación constante y opcional de comportamientos con el propósito de mejorar los patrones de alimentación, la salud y el bienestar, promoviendo el consumo de alimentos saludables, nutritivos y de manera sostenible.

“La educación nutricional es mucho más que cubrir nuestras necesidades nutricionales; consiste en una variedad de estrategias educativas, implementadas en distintos niveles, que

tienen como objetivo ayudar a las personas a lograr mejoras sostenibles en sus prácticas alimentarias. La educación nutricional se refiere también a la difusión de información útil acerca de los alimentos y sus nutrientes, y ofrecer las herramientas necesarias para mejorar la salud de las personas” (Recoletas Salud, 2023).

2.4. ALIMENTOS NATURALES

Se define como alimentos naturales a todos “aquellos a los cuáles no se les ha añadido ningún componente, es decir, no se les ha adicionado industrialmente sal, azúcar, grasas u otros componentes y tampoco han recibido un tratamiento industrial” (Guillermo, 2024).

Existen también alimentos conocidos como alimentos mínimamente procesados, los cuales no han sido enriquecidos con ningún aditivo, pero han pasado por un procesamiento mínimo que no modifica su composición natural.

2.5. ALIMENTACIÓN SALUDABLE

Una alimentación saludable es aquella que proporciona los nutrientes necesarios que el cuerpo necesita para mantener el buen funcionamiento y desarrollo del organismo, conservar o reestablecer la salud y minimizar el riesgo de enfermedades (Ministerio de Salud, 2018).

Una buena salud requiere de una alimentación variada y equilibrada, la variedad tiene que ver con la incorporación de todos los nutrientes que el cuerpo necesita, ya que por sí mismo es capaz de sintetizar muy pocos, y el equilibrio se refiere a la manera de alimentarnos que nos permite obtener de los alimentos los nutrientes necesarios para mantener nuestra salud y que se adapte a nuestras necesidades individuales: edad, sexo, actividad física, embarazo, etc.

2.6. IMPORTANCIA DE ALIMENTACIÓN

La alimentación en los niños es uno de los aspectos más importantes en su desarrollo, si los niños tienen una mala alimentación, esto puede afectar en su desarrollo, rendimiento

escolar, mayor propensión a padecer gripes y resfriados, obesidad, y otros problemas relacionados a la nutrición, estos podrían afectar corto a largo plazo, predisponiéndolos a padecer más patologías crónicas en su edad adulta (Caribe, 2018).

Como se acaba de mencionar, una alimentación por encima de sus necesidades o con alimentos pocos nutritivos, es negativos para la salud de los niños. Algunos de los problemas derivados de la mala alimentación más comunes son:

- El peso y la talla fueras de los valores normales, ya sea por exceso o defecto, como resultado de ello, podrían aparecer hipertensión arterial, osteoporosis, enfermedades renales y del corazón.
- Falta de hierro, esto puede desembocar en la anemia, esto puede afectar el desempeño en áreas emocional, cognitivo y social. “Además acorta la capacidad de analizar y entender y podría conllevar dificultades en el aprendizaje” (Sanchez Arias, 2023).
- Hipotonía o debilidad muscular, otra consecuencia de la mala alimentación, puede llegar a afectar el rendimiento físico, cognitivo y el cese del crecimiento del niño.
- Debilidad del sistema inmunitario, los niños con mala alimentación son más propensos a desarrollar infecciones de todo tipo.
- La obesidad es la acumulación excesiva de grasa en el cuerpo, puede ocasionar desórdenes hormonales, diabetes, aumento del colesterol, alteraciones óseas, etc.

Una mala alimentación en los niños puede ocasionar danos graves llegando a ser permanentes, por esto es necesario promover la buena alimentación en los niños, para garantizar un aporte adecuado de nutrientes.

2.7. ALIMENTO Y NUTRICIÓN

La definición de alimento según la Organización Mundial de la Salud [OMS], Organización de las Naciones Unidas para la Agricultura y Alimentación [FAO] (2007) “Es toda

sustancia elaborada, semielaborada o natural, que se destina al consumo humano, incluyendo las bebidas, el chicle y cualesquiera otras sustancias que se utilizan en la fabricación, preparación o tratamiento de alimentos”.

“La nutrición es el proceso fisiológico mediante el cual nuestro organismo recibe, transforma y utiliza las sustancias químicas contenidas en los alimentos” (Hurtado, 2013).

2.8. INFORMACIÓN NUTRICIONAL

La información o valor nutricional de un alimento, nos brinda información sobre los nutrientes que nos aporta para tener una alimentación sana y equilibrada. El valor nutritivo de un alimento determina el valor energético y la carga de nutrientes del mismo: grasas, vitaminas, minerales, etc.

2.8.1. Energía

La energía es el combustible que el cuerpo humano necesita para vivir y ser productivo, la caloría es una medida de energía. La energía no es ningún nutriente, este se obtiene tras la obtención de los macronutrientes a través del proceso de respiración celular. Nuestro organismo hace uso de la energía en diferentes propósitos, tales como:

- aproximadamente el 20% de la energía es usada para el metabolismo cerebral.
- Mantener la temperatura y funciones vitales, conocido también como metabolismo basal.
- Producir movimiento, por esta razón, en función del grado de la actividad física, nuestro cuerpo necesitara más o menos aporte de energía.

2.8.2. Nutrientes

Los nutrientes son sustancias químicas contenidas en los alimentos y bebidas, son indispensables la actividad del organismo, cumpliendo al menos uno de los siguientes fines básicos (ELIKA, 2018).

- Proporcionar energía.

- Aportar materiales para el crecimiento, la reparación y la reposición de los tejidos.
- Modular las reacciones bioquímicas que se producen en el organismo.

Estos nutrientes incluyen:

- Macronutrientes: hidratos de carbono, lípidos o grasos y proteínas.
- Micronutrientes: Vitaminas y minerales.

2.8.2.1. Los carbohidratos. También llamados glúcidos o hidratos de carbono, compuestos principalmente de carbono, hidrogeno y oxígeno. Constituyen la principal fuente de energía de la alimentación humana y proceden mayoritariamente de alimentos de origen vegetal. Existe tres tipos principales de carbohidratos (Sociedad Mexicana de Nutrición y Endocrinología, 2021, p. 1):

- Azúcares: Llamados también carbohidratos simples porque se encuentran en su forma más básica, se encuentran los azúcares que se encuentran naturalmente en frutas, verduras y leche.
- Almidones: Son carbohidratos complejos, ya que están hechos de muchos azúcares simples unidos, Los almidones se encuentran en pan, cereal y pasta.
- Fibra: También es un carbohidrato complejo,

2.8.2.2. Las proteínas. “Son moléculas grandes y complejas que desempeñan muchas funciones críticas en el cuerpo. Realizan la mayor parte del trabajo en las células y son necesarias para la estructura, función y regulación de los tejidos y órganos del cuerpo. Las proteínas están formadas por cientos y miles de unidades más pequeñas llamadas aminoácidos, que se unen entre sí en largas cadenas” (MedlinePlus, 2021).

Las proteínas están formadas por hidrogeno, oxígeno y nitrógeno. Algunos otros pueden contener a azufre y fosforo. Gracias a su contenido de nitrógeno benefician al crecimiento corporal, también ayudan al mantenimiento de los tejidos y permiten transportar compuestos esenciales a través de la sangre.

2.8.2.3. Las Grasas o lípidos. Son un tipo de nutriente presentes en los alimentos que el cuerpo utiliza para producir tejido nervioso y hormonas. Los lípidos contienen una gran fuente de energía, pero tienen el doble de calorías que los carbohidratos o proteínas (Gavin, 2022). Se puede distinguir tres tipos fundamentales de grasa:

- Grasas saturadas: La mayoría de este tipo de grasas son de origen animal, este tipo de grasas pueden subir el nivel del colesterol
- Grasas insaturadas: Son las más beneficiosas para el cuerpo humano por sus efectos sobre los lípidos plasmáticos y algunos contienen ácidos grasos que son nutrientes esenciales, ya que el organismo no puede fabricarlo y el único modo de conseguirlo es mediante la ingestión directa. Se encuentran en las plantas y los pescados.
- Grasas trans: Se obtiene mediante la hidrogenación de los aceites vegetales. Se encuentran en margarinas, productos horneados, snacks.

2.8.2.4. Las vitaminas. Pertenecen al grupo de los micronutrientes, las vitaminas son sustancias presentes en los alimentos, son muy necesarios en cantidades relativamente pequeñas, pero su déficit puede causar numerosas enfermedades carenciales. Se clasifican en:

- Vitaminas hidrosolubles: Vitamina B1, B2, B3, B5, B6, B7, B9, B12, C.
- Vitaminas liposolubles: Vitamina A, D, E, K.

2.8.2.5. Minerales. Pertenecen al grupo de los micronutrientes, son sustancias químicas imprescindibles para el funcionamiento del metabolismo. Los minerales pueden realizar una función reguladora o estructural (NESTLE, s. f., parr. 1).

La deficiencia o el exceso de minerales pueden perjudicar la salud. Los minerales se clasifican en:

- Macrominerales (calcio, fósforo, sodio, potasio, cloro, magnesio y azufre), son aquellos que el organismo necesita en cantidades grandes.

- Microminerales (hierro, cobre, zinc, manganeso, molibdeno, yodo, flúor, cobalto y selenio) que se necesitan en muy pequeña cantidad.

2.9. CLASIFICACIÓN DE LOS ALIMENTOS

La clasificación de los alimentos es necesario para conocer un alimento y sobre lo que aporta en mayor y menor cantidad de nutrientes para poder tener una dieta balanceada.

2.9.1. Clasificación de los Alimentos Según sus Nutrientes

Los alimentos se pueden clasificar según diferentes aspectos, tales como su función, origen, composición y nutrientes.

Si se clasifica los alimentos según sus nutrientes, se encuentra la llamada “rueda de los alimentos”, la cual junta a los alimentos en siete grupos de la siguiente manera:

Grupo I. Leche y derivados: Proteínas, Función plástica.

Grupo II. Carne, huevos y pescado: Proteínas, Función plástica.

Grupo III. Tubérculos, legumbres y frutos secos: Proteínas, carbohidratos, vitaminas y minerales, Función plástica y Energética.

Grupo IV. Verduras y frutas: Vitaminas y minerales, Función reguladora.

Grupo V. Cereales y derivados: Carbohidratos, Función energética.

Grupo VI. Grasas, aceite y mantequilla: Grasas, Función energética.

Figura 3

La rueda de los alimentos



Nota. Antiguamente esta representación se utilizaba en forma de pirámide, hoy en día se utiliza un formato circular. Tomada de *Rueda de alimentos*, por Figueroa, 2020 (<https://www.carmenfigueroa.es/la-rueda-de-los-alimentos/>).

2.9.2. Clasificación de los Alimentos Según su Función

Alimentos energéticos. Estos proveen al cuerpo energía necesaria para nuestra actividad cotidiana. Algunos ejemplos: Productos de panadería, pasta, cereales, etc.

Alimentos constructores o plásticos. Se califican así a los alimentos que facilitan la reparación celular, cicatrización de heridas y formación de tejidos. Algunos ejemplos: leche, carne, legumbres, etc.

Alimentos protectores o reguladores. Son los alimentos que contienen vitaminas y minerales. Ejemplo: Frutas, verduras, hortalizas, etc.

2.9.3. Clasificación de los Alimentos Según su Origen

Alimentos de origen vegetal. Estos surgen directamente de la tierra y aportan grandes cantidades de minerales y vitaminas al organismo.

Alimentos de origen animal. Son productos comestibles que provienen precisamente de los animales, por ejemplo: carnes rojas y blancas, huevos, lácteos, etc.

2.10. VISIÓN ARTIFICIAL

La visión artificial es la “Transformación de una imagen del mundo real a un formato digital inteligible por un ordenador y su posterior procesamiento por parte de este” (Fernandez Garcia, s. f., p. 26).

“La visión artificial es una disciplina cuyo principal objetivo es procesar y analizar imágenes del mundo real, con la finalidad de que estas puedan ser entendidas y tratadas por un ordenador” (Borrella Petisco, 2022).

La visión artificial o visión por computadora “permite que las computadoras y los sistemas obtengan información significativa de imágenes digitales, videos y otras entradas visuales, y tomen acciones o hagan recomendaciones basadas en esa información” (International Business Machines (IBM), 2023).

2.10.1. Campos de la Visión Artificial

Los campos más utilizados son:

- Medicina, la detección de tumores y la participación asistida de robots de maquinaria de alta precisión en operaciones mínimamente invasivas son un gran ejemplo de la utilidad de esta tecnología.
- Industrial en el control de calidad, haciendo uso de sensores especiales un robot puede

optimizar todo el proceso de selección y de control de calidad tanto en piezas del sector automotriz, como frutas y alimentos defectuosos en los sectores alimenticios de todo tipo.

- En los trabajos de precisión, los sistemas de visión artificial son capaces de detectar y medir piezas, diseños y patrones, llevando a cabo acciones precisas.
- Lectura y procesamiento de datos, los protocolos para lectura de caracteres, símbolos y datos basados en cámaras que implementen códigos de reconocimiento de patrones son cada vez más efectivos.
- Vigilancia y seguridad, la visión artificial puede detectar, identificar y seguir objetos para describir y comprender su comportamiento.

2.10.2. Tareas de la Visión Artificial

Entre las principales tareas de la visión artificial cabe destacar: clasificación de imágenes, detección de objetos, seguimiento, recuperación de imágenes basada en contenido, reconocimiento, análisis de videos, reconstrucción de escena, restauración de imágenes, etc.

2.10.2.1. Clasificación de imágenes. La clasificación de imágenes es una rama de la visión artificial con numerosas aplicaciones, consiste en ver una imagen y poder clasificarla, más precisamente poder determinar a qué tipo pertenece una imagen, para esto se utilizan redes neuronales capaces de detectar patrones en las imágenes; sin embargo, cada día surgen nuevas arquitecturas y algoritmos para entrenar modelos.

2.10.2.2. Detección de objetos. La detección o reconocimiento de objetos por inteligencia artificial (IA) permite a los sistemas informáticos analizar y comprender el contenido visual de una imagen. Utilizando algoritmos y modelos de IA, pueden identificar objetos, reconocer rostros, extraer texto y mucho más.

2.10.2.3. Seguimiento. El seguimiento de objetos sigue o rastrea un objeto una

vez que se detecta. Esta a menudo se ejecuta con imágenes capturadas en secuencia o con videos en tiempo real. Los vehículos autónomos, por ejemplo, no solo deben clasificar e identificar objetos como peatones, otros automóviles e infraestructura vial, sino que también deben detectarlos en movimiento para evitar colisiones y obedecer las leyes de tránsito

2.10.3. Aprendizaje Automático (Machine Learning)

El aprendizaje automático (AA) implica la creación de algoritmos y modelos estadísticos que aprovechan la capacidad de las computadoras para realizar tareas sin recibir instrucciones directas. En lugar de depender de instrucciones explícitas, se basa en la detección de patrones y la realización de inferencias. Mediante el uso de algoritmos de aprendizaje automático, las computadoras pueden analizar extensos conjuntos de datos históricos y reconocer tendencias en ellos. Esto les permite ofrecer resultados más precisos a partir de un conjunto inicial de datos (Amazon Web Services (AWS), 2023).

Los algoritmos de Machine Learning se pueden clasificar en cuatro estilos de aprendizaje distintos en función de la salida esperada y del tipo de entrada, estos son:

2.10.3.1. Aprendizaje Automático supervisado. Son algoritmos con conjuntos de datos etiquetados que “clasifiquen datos o predigan resultados de forma precisa. A medida que los datos se introducen en el modelo, este ajusta sus ponderaciones hasta que dicho modelo se haya ajustado adecuadamente” (AWS, 2023).

El aprendizaje supervisado utiliza un conjunto de datos de entrenamiento etiquetados para enseñar a los modelos a generar la salida deseada. Estos datos incluyen datos de entrada y resultados correctos, que permitirá que el modelo aprenda con el tiempo. Algunos algoritmos de aprendizaje supervisado son:

- Redes Neuronales
- Naive Bayes
- Regresión Lineal

- Regresión Logística
- Máquinas de Vectores de Soporte (SVM)
- K Vecino más Cercano (KNN)x
- Bosque aleatorio

El etiquetado de datos o anotación es parte del pre tratamiento en el desarrollo de un modelo machine learning, que refiere al proceso para categorizar los datos sin procesar con los valores de salida definidos, de manera que un modelo de aprendizaje automático pueda aprender de ellos.

2.10.3.2. Aprendizaje Automático no Supervisado. El aprendizaje automático no supervisado usa “algoritmos de machine learning para analizar y agrupar en clústeres conjuntos de datos sin etiquetar. Estos algoritmos descubren agrupaciones de datos o patrones ocultos sin necesidad de ninguna intervención humana” (IBM, 2023). El aprendizaje automático no supervisado ocurre cuando se proporciona datos de entrada al algoritmo sin tener ningún dato de salida previamente etiquetado. Posteriormente, el algoritmo por sí mismo descubre patrones y conexiones dentro de los datos. A continuación, se describen algunos tipos de técnicas utilizadas en este tipo de aprendizaje:

- Agrupación en clústeres
- Aprendizaje de reglas de asociación
- Densidad de probabilidad
- Reducción de dimensionalidad

2.10.3.3. Aprendizaje Automático semi supervisado. Es una técnica de aprendizaje automático híbrido que utiliza una combinación de datos etiquetados y no etiquetados. Esta técnica se fundamenta en el uso de una reducida porción de datos etiquetados junto con una abundante cantidad de datos sin etiquetar para entrenar los sistemas. Inicialmente, los datos etiquetados se emplean para un entrenamiento parcial del

algoritmo de machine learning (Gamco, 2021). Posteriormente, el propio algoritmo, entrenado de manera parcial, asigna etiquetas a los datos no etiquetados, un procedimiento conocido como pseudo etiquetado. Seguidamente, el modelo se reentrena utilizando la combinación resultante de datos, sin necesidad de programación explícita.

2.10.3.4. Aprendizaje por refuerzo. El aprendizaje por refuerzo (RL) es una técnica dentro del campo del machine learning (ML) que instruye al software para que tome decisiones con el fin de obtener los mejores resultados. Se asemeja al método de aprendizaje por prueba y error utilizado por los seres humanos para alcanzar sus metas. Las acciones del software que contribuyen al logro de su objetivo se refuerzan, mientras que aquellas que se desvían del mismo se desestiman.

Los algoritmos de RL emplean un enfoque de recompensa y castigo al procesar datos.

Aprenden de la retroalimentación de cada acción y, de manera autónoma, determinan las mejores estrategias para alcanzar los resultados deseados. Además, estos algoritmos pueden operar considerando la gratificación diferida.

2.10.4. Aprendizaje por transferencia (TL)

El aprendizaje por transferencia es una técnica del machine learning “en la que un modelo previamente entrenado para una tarea se ajusta con precisión para una nueva tarea relacionada” (AWS, 2023). Entrenar un nuevo modelo de machine learning es un procedimiento extenso y exigente, que demanda abundante información, capacidad computacional y múltiples repeticiones antes de estar apto para su implementación; sin embargo, utilizando como punto de partida modelos pre-entrenados, el aprendizaje por transferencia permite desarrollar rápidamente modelos eficaces y resolver problemas complejos de visión por computadora o procesamiento de lenguaje natural (NPL).

Esta metodología se utiliza con frecuencia en aplicaciones de detección de objetos, reconocimiento de imágenes, reconocimiento de voz, entre otros. El aprendizaje por

transferencia es una técnica de uso habitual ya que:

- Permite entrenar modelos con menos datos etiquetados reutilizando modelos de uso habitual que ya se han entrenado con conjunto de datos de gran tamaño.
- Ayuda a reducir el tiempo de entrenamiento y los recursos informáticos.
- Permite aprovechar arquitecturas de modelos desarrollados por la comunidad de investigadores.

¿Cuáles son los pasos del aprendizaje por transferencia?

Existen tres pasos principales para ajustar un modelo de machine learning para una nueva tarea:

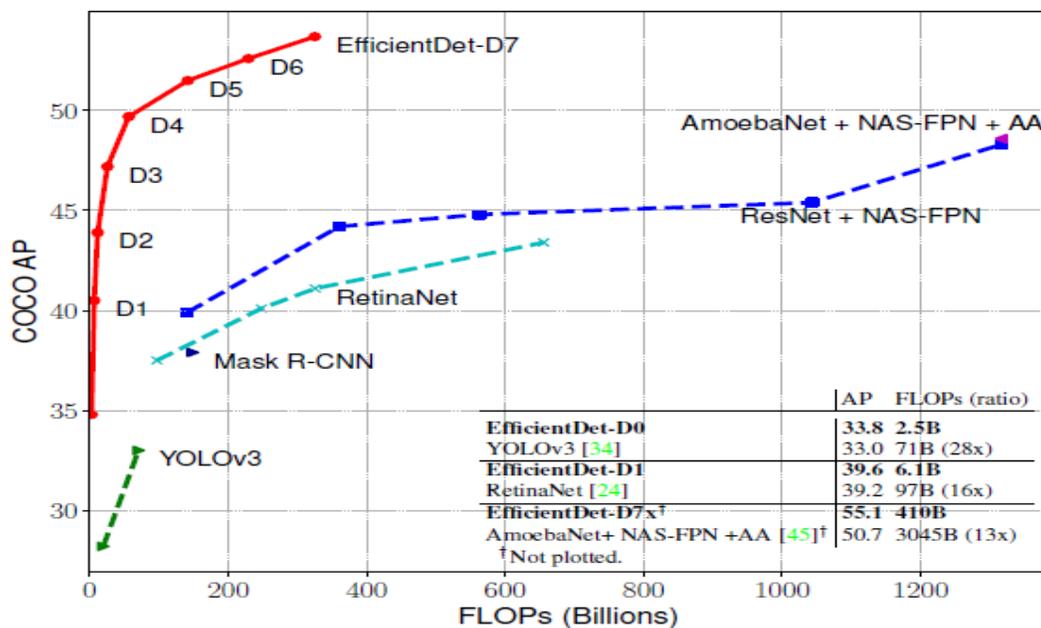
- Seleccionar un modelo previamente entrenado. Un aspecto importante al momento de elegir un modelo es comprender la tarea original del modelo para que haya una transición más eficaz hacia una nueva tarea.
- Configurar el modelo previamente entrenado. Hay dos métodos para esto: congelar las capas previamente entrenadas, eliminar la última capa.
- Entrenar el modelo para el dominio del destino. Es probable que el modelo que ha sido entrenado genere resultados diferentes a los deseados. Tras observar y evaluar cómo se desempeña el modelo durante el proceso de entrenamiento, es posible ajustar los hiperparámetros o la arquitectura de la red neuronal para mejorar la calidad de los resultados obtenidos.

2.10.5. Arquitectura de modelo o algoritmo EfficientDet-Lite

EfficientDet es una familia de modelos de detección de objetos, estos modelos funcionan de 2 a 4 veces más rápido en la GPU y de 5 a 11 veces más rápido en la CPU que otros detectores.

Figura 4

Rendimiento del modelo EfficientDet



Nota. EfficientDet logra una precisión promedio media (mAP) de 52.2, superando el modelo anterior de última generación en 1.5 puntos. Por Mingxing Tan, et al. (2022).

Los efficientDet se desarrollan sobre la base de la red troncal avanzada, un nuevo BiFPN y una nueva técnica de escalado:

- Backbone: se emplea EfficientNet como red troncal.
- BiFPN: una red de característica bidireccional mejorada con normalización rápida.
- Escalado: se utiliza un solo factor de escala compuesto para controlar la profundidad, el ancho y resolución de todas las redes troncales, de funciones y de predicción.

2.10.6. Biblioteca Model Maker

Model Maker es una biblioteca de TensorFlow Lite que “simplifica el proceso de entrenamiento de un modelo de TensorFlow Lite con un conjunto de datos personalizados. Utiliza el aprendizaje por transferencia para reducir la cantidad de datos de entrenamiento

necesarios y acortar el tiempo de entrenamiento” (TensorFlow, 2022). Actualmente esta biblioteca admite las siguientes tareas de aprendizaje automático: clasificación de imágenes, detección de objetos, clasificación de texto, respuesta a la pregunta BERT, clasificación de audio y recomendación. Model Maker permite entrenar modelos de TensorFlow Lite usando un conjunto de datos personalizados en solo unas pocas líneas de código. Los módulos de esta biblioteca son:

- módulo `audio_classifier`: API para entrenar un modelo de clasificación de audio.
- módulo `config`: APIs para la configuración de TFLite Model Maker.
- módulo `image_classifier`: API para entrenar un modelo de clasificación de imágenes.
- módulo `model_spec`: API para la especificación del modelo de TFLite Model Maker.
- módulo `object_detector`: API para entrenar un modelo de detección de objetos.
- módulo `question_answer`: API para entrenar un modelo que puede responder preguntas basadas en un texto predefinido.
- módulo `recommendation`: API para entrenar un modelo de recomendación en el dispositivo.
- módulo `text_classifier`: API para entrenar un modelo de clasificación de texto.

2.11. REALIDAD AUMENTADA

“Realidad aumentada es un recurso tecnológico que ofrece experiencias interactivas al usuario a partir de la combinación entre la dimensión virtual y la física, con la utilización de dispositivos digitales” (Grapsas, 2019).

La realidad aumentada es un entorno que incluye datos e información digital en aspectos del mundo real (Azuma, 1997).

“La realidad aumentada es una versión mejorada e interactiva de un entorno del mundo real que se logra a través de elementos visuales digitales, sonidos y otros estímulos sensoriales mediante tecnología holográfica” (Corporation Microsoft, s. f.)

Figura 5

Ejemplo de realidad aumentada



Nota: Adaptado de *Realidad Aumentada*, por Bouzas, 2021.

2.11.1. Elementos de la realidad aumentada

Según Edgar Mozas Fenoll (2016), en general, la realidad aumentada debe estar formada por los siguientes elementos:

- Cámara. Dispositivo que capta la imagen del mundo real.
- Procesador. Elemento del hardware que combina la imagen con la información que debe superponer.
- Software. Programa informático que gestiona el proceso.
- Pantalla. Dispositivo que muestra los elementos del mundo real superpuestos con los elementos de la realidad aumentada.
- Activador. Elemento del mundo real que utiliza el software para reconocer el entorno físico y seleccionar la información virtual asociada que se debe añadir.
- Marcador. Encargado de reproducir las imágenes creadas por el procesador y donde se verá el modelo en 3D.

2.11.2. Niveles de la realidad aumentada:

Lens-Fitzgerald, (De la Horra Villace, 2017) , propone una clasificación de cuatro niveles para aplicaciones de realidad aumentada:

- Nivel 0. Hiperenlazado al mundo físico (physical world hyper linking). Este nivel se refiere al uso de código de barras, códigos QR, etc.
- Nivel 1. Realidad aumentada basada en marcadores (marker based AR), son imágenes en blanco y negro, con dibujos sencillos y asimétricos.
- Nivel 2. Realidad aumentada sin marcadores (markerless AR), se refiere al uso de objetos, imágenes o localizaciones, haciendo uso de GPS y la brújula de los dispositivos para localizar situaciones o puntos de interés.
- Nivel 3. Visión aumentada (Augmented Visión), este nivel usa dispositivos creados específicamente para la realidad aumentada, un ejemplo de este nivel son las gafas de Google Glass y las HoloLens de Microsoft.

2.12. METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE: MOBILE-D

Según Amaya (2013, p. 8), esta metodología se basa en prácticas ágiles como Extreme Programming (XP), Crystal Methodologies y Rational Unified Process (RUP). El objetivo de esta methodologies es conseguir ciclos de desarrollo rápidos en equipos muy pequeños.

La metodología Mobile-D consta de cinco fases: exploración, iniciación, producción, estabilización, y prueba del sistema. Cada una de estas fases, tienen también etapas, tareas y prácticas asociadas.

2.12.1. Exploración

Esta fase se centra en la planificación y el establecimiento de las características del proyecto. Las tareas dentro de esta fase son:

- Recopilación de la información
- Establecimiento de actores

- Requisitos iniciales
- Planificación inicial
- Definición del alcance.

2.12.2. Iniciación

En esta fase se debe preparar, identificar y establecer el entorno técnico, como los recursos físicos, tecnológicos y de comunicaciones. Esta fase se divide en cuatro etapas:

- Puesta en marcha del proyecto
- Arquitectura del software
- Planificación inicial
- Día de prueba
- Día de salida.

2.12.3. Producción

Se repiten iterativamente la programación (planificación, trabajo, liberación) hasta implementar todas las funcionalidades usando el desarrollo dirigido por pruebas para llevar a cabo toda la implementación. Esta fase consta de tres etapas:

- Día de planificación. Tareas y análisis de requisitos.
- Día de trabajo. Se implementa la funcionalidad del sistema previsto durante el día de planificación.
- Día de liberación. Se debe llevar a cabo la integración del sistema, seguido de las pruebas de aceptación.

2.12.4. Estabilización

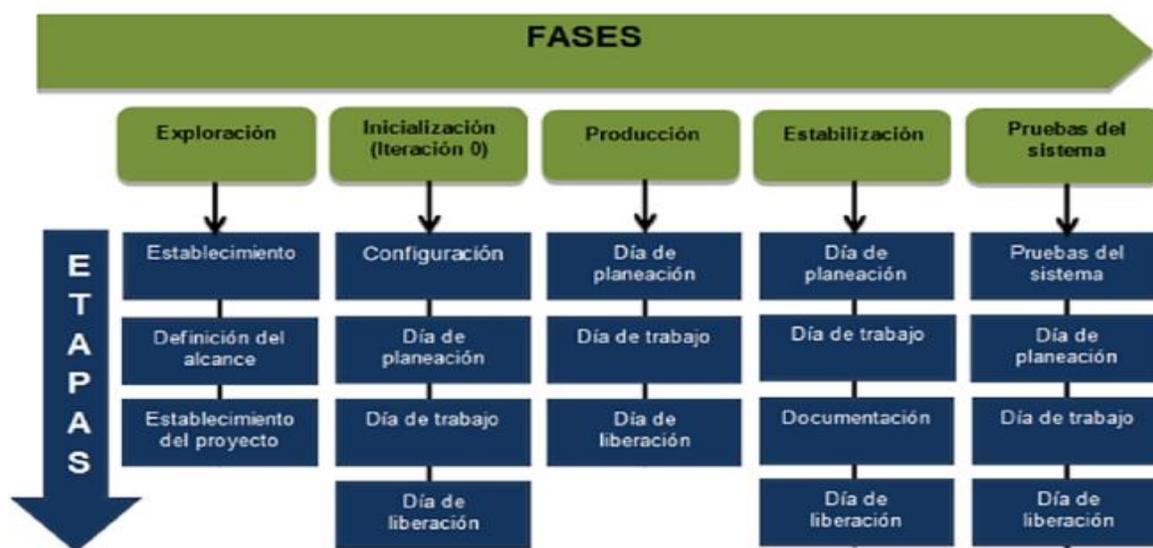
Se realizan las últimas acciones de integración de los módulos separados y asegurar que el sistema completo funcione correctamente. En esta fase se debe considerar la elaboración de la documentación.

2.12.5. Pruebas

Se pasa a una fase de testeo hasta llegar a una versión estable según lo establecido anteriormente con el cliente, esto para lograr la disponibilidad de una versión estable y plenamente funcional del sistema.

Figura 6

Fases de la metodología Mobile D



Nota: Adaptada de *Metodología Mobile-D*, por Leyva, et al. (2017)

2.13. FASES PARA CREAR UN MODELO DE MACHINE LEARNING (CRISP – DM)

CRISP-DM (Cross Industry Estándar Process for Data Mining) es una metodología ampliamente utilizada en proyectos de Ciencia de Datos, Minería de Datos y Big – Data. Esta metodología está compuesta por seis fases, pudiendo tomar forma secuencial o cíclica, dependiendo de los resultados de los resultados del modelo (Rueda, 2019). Estas fases son:

2.13.1. Comprensión del negocio

En esta fase se debe comprender a profundidad la problemática que se quiere solucionar, estableciendo los objetivos y requisitos del proyecto

2.13.2. Comprensión de los datos

Dado el problema que se desea resolver, se debe investigar y obtener datos que se utilizara para entrenar a la máquina. La cantidad y calidad de información que se consiga, impactara directamente en el resultado del modelo., las tareas dentro de esta fase son:

- Recolección de datos.
- Describir los datos.
- Explorar los datos.
- Verificar la calidad de los datos.

2.13.3. Preparación de los datos

El objetivo de esta fase es la construcción del conjunto de datos (dataset) final, los cuales serán utilizados para entrenar el modelo. Estos datos deben ser ordenados y limpiados, así también se debe tener balanceada la cantidad de datos que se tiene por cada clase y debe separarse los datos generalmente en dos grupos: uno para entrenamiento y el otro para la evaluación del modelo. Las tareas dentro de esta fase son:

- Limpieza de datos.
- Construcción de datos.
- Integración de datos.
- Dar formato a los datos.

2.13.4. Modelado

Se debe elegir un modelo que se adecue al problema que queremos resolver. Existen algoritmos de clasificación, detección, regresión, etc. Algunos modelos y sus aplicaciones que generalmente se utilizan son: Logistic Regresion (predicción de precios de inmuebles), Fully connected networks (clasificación), Recurrent Neural Networks (reconocimiento de voz), etc. En este paso se entrenará el modelo con los datos de entrenamiento preparados en el paso anterior, se debe verificar los resultados obtenidos y corregirlos y volver a iterar.

2.13.5. Evaluación

Si el modelo cumple con las expectativas del proyecto se procede a la fase de despliegue, y si no, se debe iterar nuevamente hasta llegar a los resultados deseados.

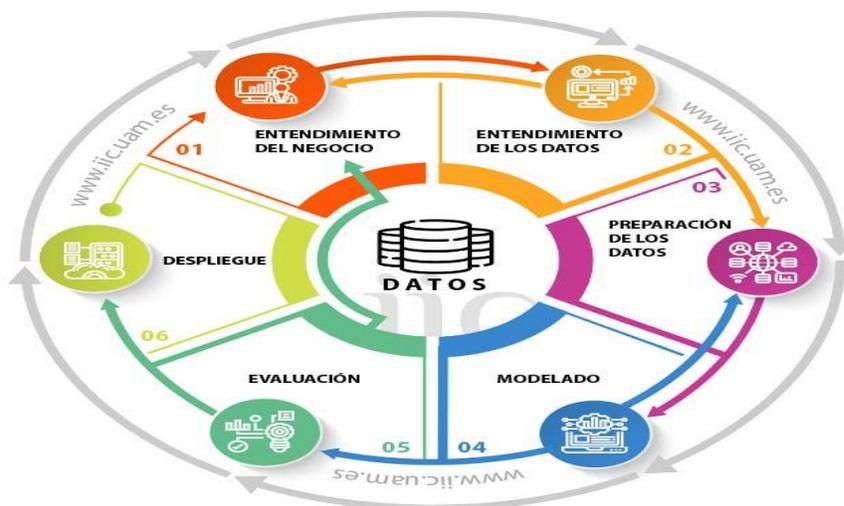
En un modelo de reconocimiento se debe evaluar el modelo creado con los datos de evaluación que contiene entradas que el modelo desconoce y verificar la precisión del modelo ya entrenado. Si durante la evaluación no se obtuvo buenos resultados y la predicción es mínima, es posible que se tenga problemas de overfitting y se debe retornar al paso de entrenamiento, haciendo antes una configuración de parámetros del modelo. Se puede incrementar la cantidad de veces que se itere los datos de entrenamiento (EPOCHS).

2.13.6. Despliegue

En esta fase se debe planificar una estrategia para ejecutar la puesta en marcha del modelo en un entorno de producción, se ve la relación entre la simulación y el mundo real, por ende, se puede integrar el modelo a un sistema real con el que pueda comunicarse.

Figura 7

Esquema del ciclo CRISP-DM estándar



Nota: Reproducida de CRISP-DM estándar, por Pablo Haya, 2022.

2.14. ISO 25010

La ISO 25010 es un estándar global que ofrece pautas para valorar la calidad de los sistemas y software. Se enfoca en los aspectos que deben cumplir los productos de software para asegurar la satisfacción del usuario y su eficacia en la operatividad (NormasISO.org, 2023b). Las características que definen la calidad se muestran en la siguiente figura:

Figura 8

Calidad del producto del software



Nota: Adaptada de Modelo de calidad del producto, por Rodríguez et al. (2015),

2.15. COCOMO II

COCOMO II, o Constructive Cost Model II, es una técnica de estimación de costos y planificación de proyectos de software. Es una versión mejorada del modelo COCOMO original desarrollado por Barry Boehm en la Universidad del Sur de California en la década de 1980. COCOMO II proporciona una serie de modelos y herramientas para estimar la cantidad de

esfuerzo, tiempo y recursos necesarios para completar un proyecto de desarrollo de software (Gómez et al., s. f.). COCOMO II está compuesto por tres modelos:

- **Composición de la aplicación**, se emplea durante la etapa de prototipación del desarrollo del software.
- **Modelo diseño temprano**, se utiliza en las primeras etapas del desarrollo del software, donde se evalúan las alternativas de hardware y software del proyecto.
- **Post arquitectura**, se aplica en la etapa de desarrollo, este modelo utiliza: puntos función y/o líneas de código fuente, un conjunto de 17 atributos denominados factores de costo, cinco factores que determinan un exponente.

2.16. ISO 27034

La ISO 27034, conocida también como "Seguridad de la Información en Aplicaciones", es un conjunto de directrices que ayuda a garantizar la seguridad de las aplicaciones y en proteger la información confidencial. Este estándar se fundamenta en la gestión de riesgos y ofrece pautas para aplicar medidas de seguridad a lo largo del ciclo de vida de las aplicaciones (NormasISO.org, 2023).

La ISO 27034 proporciona algunos beneficios que incluyen:

- Mejora de la seguridad de las aplicaciones.
- Reducción de riesgos y vulnerabilidades.
- Cumplimiento normativo.
- Eficiencia y efectividad en el desarrollo de las aplicaciones.
- Protección de la información confidencial.

2.17. PRUEBAS DE SOFTWARE

Las pruebas de caja blanca y negra son dos enfoques distintos para realizar pruebas de software que se utilizan para evaluar la funcionalidad y calidad de un programa de software desde diferentes perspectivas.

2.17.1. Pruebas de Caja Blanca o Pruebas estructurales

La prueba de caja blanca se puede realizar para varios propósitos diferentes. Los tres tipos de pruebas de caja blanca son (Tripleten, 2024):

- Pruebas unitarias: Estas pruebas se enfocan en asegurar que cada componente o función dentro de una aplicación funcione correctamente. Esto garantiza que la aplicación cumpla con los requisitos de diseño durante todo el proceso de desarrollo.
- Pruebas de integración: Estas pruebas se centran en las interfaces entre los componentes dentro de una aplicación. Realizadas después de las pruebas unitarias, aseguran que no solo cada componente funcione bien por separado, sino también que puedan trabajar juntos de manera efectiva.
- Pruebas de regresión: Los cambios pueden dañar cosas dentro de una aplicación. Las pruebas de regresión garantizan que el código siga superando los casos de prueba existentes después de realizar actualizaciones de funcionalidad o seguridad en una aplicación.

Algunas de las características de las pruebas de caja blanca son:

- También conocidas como pruebas estructurales, estas pruebas se centran en examinar la estructura interna del código fuente del software.
- Los probadores de caja blanca tienen acceso al código fuente del programa y conocen su lógica interna y estructura.
- Las pruebas de caja blanca están diseñadas para probar caminos de ejecución específicos en el código, asegurándose de que se ejecuten todas las declaraciones y se alcancen todas las ramas de control.
- Los métodos comunes de pruebas de caja blanca incluyen pruebas de cobertura de código, pruebas de camino, pruebas de flujo de datos, entre otros.

2.17.2. Pruebas de Caja Negra o Pruebas Funcionales

Las técnicas de prueba son herramientas que los testers utilizan para localizar los casos de prueba más efectivos para detectar errores y asegurar una cobertura óptima en las pruebas de un sistema. Al utilizar técnicas de caja negra, los testers se enfocan exclusivamente en las entradas y salidas del sistema, sin considerar su estructura interna (Gomez, 2023).

La técnica de **partición de equivalencia**: implica dividir los datos de entrada en categorías o clases de equivalencia, agrupando aquellos que producen el mismo resultado. Esta técnica se basa en la idea de que todos los elementos dentro de una clase deben ser tratados de manera similar por el sistema. Algunas consideraciones importantes para aplicar esta técnica son:

- Cada dato de entrada debe pertenecer a una sola clase de equivalencia.
- Los valores válidos que el sistema debe aceptar se denominan "partición de equivalencia válida".
- Los valores no válidos que el sistema rechaza se denominan "partición de equivalencia no válida".
- Cada partición de equivalencia no válida debe ser tratada como un caso de prueba independiente y no combinarse con otros casos solo porque son inválidos.

Análisis de valor de límite (Boundary Value Analysis): Esta técnica es una extensión de la partición de equivalencia, solo se utiliza para valores numéricos y cuando los valores se encuentran ordenados. Una partición contiene un valor inicial y un valor final (un número mínimo y uno máximo) y son denominados como valores límite.

A diferencia de la partición de equivalencia, que toma como mínimo un número dentro del rango de la partición como dato de entrada, en esta técnica obligatoriamente deben tomarse como mínimo los dos valores límite y opcionalmente, un valor entre los límites.

Tabla de decisión (Decision table): Es utilizada para probar el comportamiento de un sistema utilizando diferentes combinaciones de entrada. Este es un enfoque donde las diferentes combinaciones de entrada y los resultados se muestran en forma tabular.

- En las filas se colocan las condiciones y los resultados, donde las condiciones se ponen en la parte superior y los resultados en la parte inferior.
- Cada columna corresponde a una regla la cual es definida por las combinaciones de entrada.
- Los valores a colocarse son generalmente booleanos como V (verdadero) y F (Falso). Si hay un valor que no importa para obtener cierto resultado se puede marcar con un guion "-".

Técnica de Transición de Estado (Statu Transition Testing): En esta técnica las condiciones de entrada provocan cambios de estado. Es ideal para cuando el sistema bajo prueba depende de eventos o valores pasados y para cuando se debe probar un sistema que consta de una secuencia de eventos.

2.18. HERRAMIENTAS

2.18.1. Google Colaboratory

“Google Colaboratory, o “Colab” para abreviar, es un producto de Google Research. Permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador. Es especialmente adecuado para tareas de aprendizaje automático, análisis de datos y educación” (Google, 2024).

Colaboratory permite ejecutar celdas de código como si se tratara de un cuaderno de Jupyter Notebook, es perfecto para implementar algoritmos de aprendizaje automático, ya que no se limita en los recursos locales de la computadora, sino que se ejecuta en las máquinas virtuales de Google, estas tienen un ciclo de vida.

2.18.2. Android Studio

“Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de apps para Android y está basado en IntelliJ IDEA” (Android Studio, 2024). Algunas de sus características son:

- Compilación flexible basado en Gradle.
- Emulador rápido y cargado de funciones.
- Entorno unificado lo cual permite trabajar con cualquier tipo de dispositivo Android.
- Proporciona compatibilidad con servicios en la nube como Google Cloud Platform.

Android Studio admite lenguajes de programación como: Kotlin, Java, y C++.

2.18.3. Librería TensorFlow

Tensorflow es una biblioteca de código abierto para el aprendizaje automático (AA), fue desarrollada por Google para construir y entrenar redes neuronales, los cálculos se pueden realizar en CPU, GPU y TPU (unidad de procesamiento específicamente para el aprendizaje automático) (Gazca, 2018). TensorFlow nos permite crear y entrenar modelos de aprendizaje automático mediante redes neuronales y APIs intuitivas.

2.18.4. Librería Numpy

Es una librería de python que se especializa en el cálculo numérico y el análisis de datos. Esta librería es ideal para el procesamiento de vectores y matrices de grandes dimensiones multidimensionales

2.18.5. Python

Python es un lenguaje de programación de código abierto, multiparadigma y multinivel, soporta parcialmente la programación orientada a objetos, imperativa y funcional. Es un lenguaje interpretado con una sintaxis clara y fácil de utilizar, tipado dinámicamente, siendo de los lenguajes de programación más utilizados en el ámbito educativo y científico: aplicaciones web, ciencia de datos, y el aprendizaje automático (Caminiti, 2021). Se puede integrar con los lenguajes de programación: C, C++, COM, ActiveX, CORBA y JAVA.

Entre las bibliotecas de Python más populares se encuentran: Pandas, NumPy, Requests y Keras. Entre los frameworks más conocidos de Python se encuentran: Django, Flask y Pytorch.

2.18.6. Java

Java es un lenguaje de programación multiplataforma, orientada a objetos, tipado estáticamente, dinámico e interpretado.” Es un lenguaje de programación rápido, seguro y fiable para codificar todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del lado del servidor” (AWS, s. f.).

Java es ampliamente usado debido a:

- Funciones y bibliotecas incorporadas
- Recursos de aprendizaje de alta calidad.
- Apoyo comunitario activo.
- Herramientas de desarrollo de alta calidad.
- Plataforma independiente.
- Seguridad.

2.18.7. Kotlin

“Es un lenguaje de programación multiplataforma, estáticamente tipado, de alto nivel y propósito general con inferencia de tipos. Kotlin está diseñado para ser totalmente interoperable con Java” (Wikipedia, 2024).

Algunas características y ventajas de kotlin son:

- Interoperabilidad con Java, kotlin es compatible con java, lo que significa que se puede utilizar bibliotecas y frameworks de java en proyectos de kotlin.
- Curva de aprendizaje sencilla, es un lenguaje intuitivo, fluido y fácil de aprender, además como es de código abierto, tiene un gran apoyo por parte de la comunidad de kotlin.
- Soporte de plataforma múltiple, se puede utilizar para cualquier tipo de desarrollo, desde la web del lado del servidor y del lado del cliente, hasta Android e iOS, como el lenguaje se ejecuta en JVM, permite compartir código entre diferentes plataformas.
- Flexibilidad, es un lenguaje flexible que tiene construcciones funcionales y orientadas a

objetos.

2.18.8. Roboflow

Roboflow es un marco de desarrollo de Visión por Computadora para recopilación de datos, pre procesamiento y técnicas de entrenamiento de modelos (Bhattacharyya, 2020). Esta herramienta tiene conjunto de datos públicos disponibles para los usuarios, también pueden crear sus propios conjuntos de datos personalizados. Roboflow acepta varios tipos de anotación, para el entrenamiento de modelos dispone de una amplia variedad de bibliotecas de modelos, tales como: EfficientNet, MobileNet, Pytorch, etc.

CAPÍTULO III

3. MARCO APLICATIVO

3.1. INTRODUCCIÓN

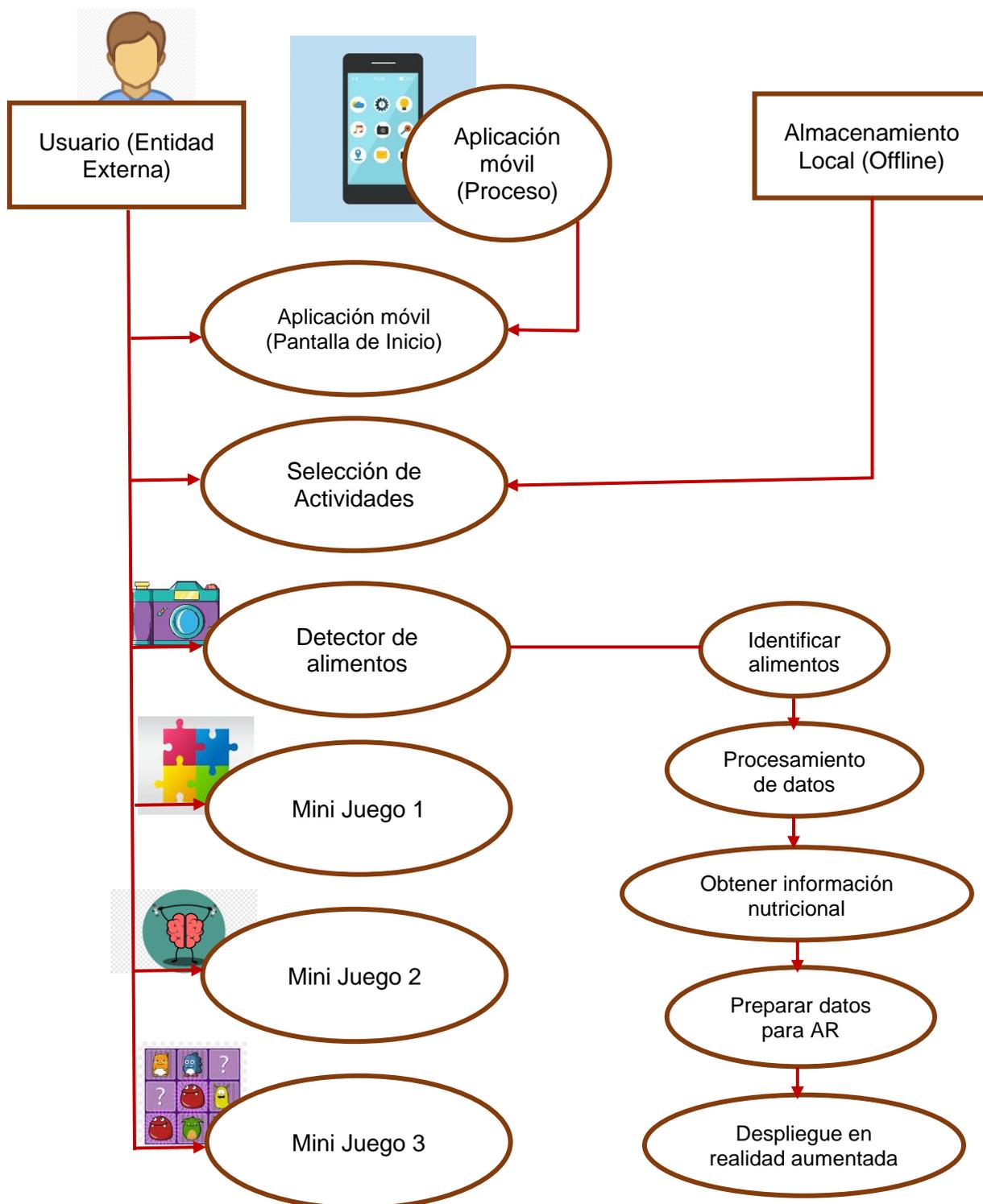
En este capítulo se detallará el proceso del desarrollo de la aplicación, para ello se utilizarán dos metodologías:

Para el desarrollo de la aplicación móvil de este proyecto se utilizará la metodología Mobile-D, el cual consta de cinco fases: exploración, iniciación, producción, estabilización y pruebas. Para realizar el Modelo de Reconocimiento de Alimentos se utilizara la metodología CRISP-DM, para esto se seguirán los siguientes pasos: coleccionar datos, preparar datos, elegir el algoritmo, entrenar el modelo, evaluar el modelo.

DIAGRAMA DE FLUJO DE DATOS DE LA APLICACIÓN MÓVIL

Figura 9

Diagrama de Flujo de Datos



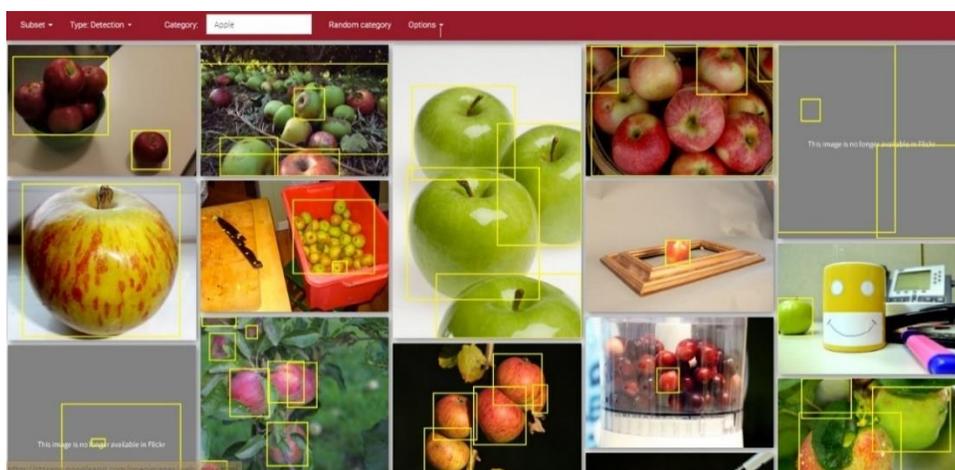
3.2. DESARROLLO DEL MODELO DE RECONOCIMIENTO CON CRISP-DM

3.2.1. Colectar y entender datos

En este paso se recolectó las imágenes para construir el conjunto de datos que servirá para la construcción del modelo. Como principal fuente de imágenes para la construcción del conjunto de datos se utilizó Open Image Dataset v6.

Figura 10

Open Image DataSet V6



El conjunto de datos se construyó con 15 clases, cada clase compuesta por 350 imágenes aproximadamente, estas clases son:

Tabla 2

Clases que se usaran para el modelo

CLASES		
Manzana	Zanahoria	Durazno
Uva	Tomate	Sandia
Naranja	Calabaza	Rábano
Plátano	Brócoli	Piña
Frutilla	Palta	Mango

Se recolecto las imágenes para el entrenamiento con varios ángulos, resoluciones y fondos. Los requisitos generales que deben cumplir las imágenes para formar parte del set de datos son:

Tabla 3

Requisitos generales para la imagen

Tipos de archivos permitidos	JPEG PNG ICO BMP
Tamaño del archivo de la imagen	Tamaño máximo: 1.5 Mb
Tamaño de la imagen en pixeles	Tamaño sugerido: 1024 x 1024 pixeles

3.2.2. Preparar datos

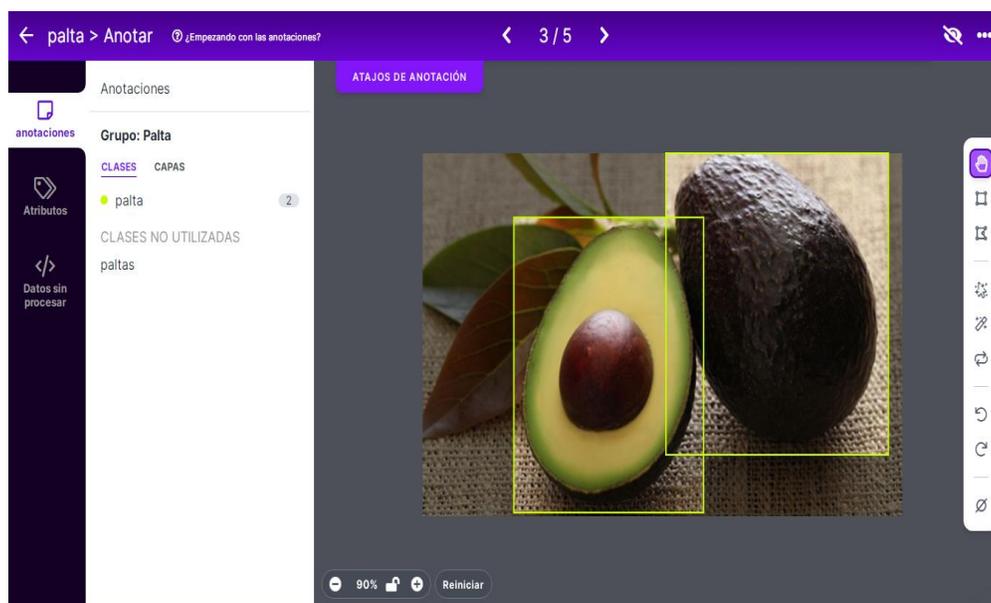
En este paso se etiquetó las imágenes con la herramienta Roboflow, los requisitos para el etiquetado son:

Tabla 4

Requisitos de etiquetas

Requisitos de anotación	Por cada clase, se recomienda el uso de 1000 anotaciones por etiqueta. Los cuadros de límites deben estar etiquetados de forma exhaustiva: si en una imagen hay dos manzanas, deben etiquetarse ambas en cuadros diferentes.
Tamaño del cuadro de limite	8 x 8 pixeles, como mínimo.
Cantidad de imágenes por etiqueta	Se recomienda al menos 100 muestras, en clases menores a 50 clases.

Figura 11

Etiquetado de imágenes

Las imágenes etiquetadas se dividieron de la siguiente forma:

- Conjunto de datos de entrenamiento (80% de las imágenes)
Este conjunto de datos se utilizará para entrenar el modelo
- Conjunto de datos de validación (10% de las imágenes)
Este conjunto de datos se utilizará para la optimización de parámetros o con el fin de decidir cuándo detener el entrenamiento.
- Conjunto de datos de prueba (10% de las imágenes)
Este conjunto se utilizará para evaluar el modelo: determinar la tasa de errores, la calidad y la exactitud del modelo.

Posteriormente se debe crear un archivo CSV con los cuadros de límite y etiquetas, es necesario que el archivo CSV se encuentre en el siguiente formato:

- Conjunto a asignar el contenido de esa fila (TRAIN, VALIDATION, TEST).
- La dirección de la imagen donde se encuentre.

- La etiqueta (clase) que identifica como se categoriza el objeto.
- Cuadro límite del objeto ($x_relative_min$, $y_relative_min$, $x_relative_max$, $y_relative_max$).

Figura 12

Formato a la imagen

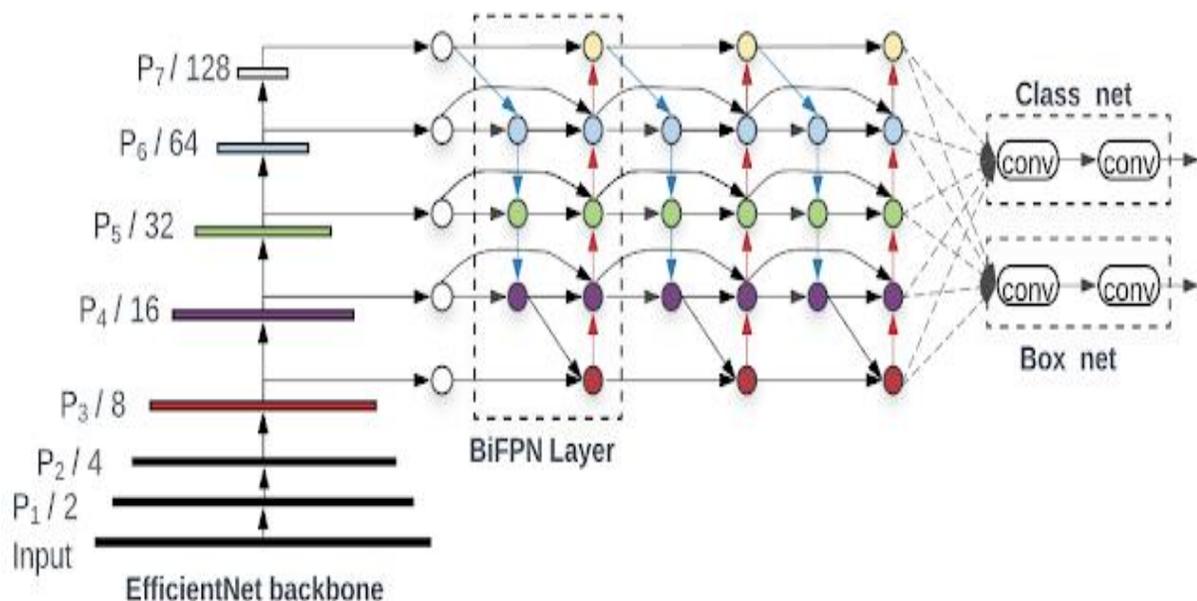
	A	B	C	D	E	F	G	H	I	J	K
A1	TRAIN,/content/drive/MyDrive/frutas/palta/142.jpg,Palta,0.037037037,0.325,,,0.591111111,0.683333333										
468	TRAIN,/content/drive/MyDrive/frutas/palta/100.jpg,Palta,0.423913043,0.693311582,,,0.895380435,0.94045677										
469	TRAIN,/content/drive/MyDrive/frutas/palta/100.jpg,Palta,0.513586957,0.454323002,,,1,0.693311582										
470	TRAIN,/content/drive/MyDrive/frutas/palta/15.jpg,Palta,0.390277778,0.180208333,,,0.602777778,0.34375										
471	TRAIN,/content/drive/MyDrive/frutas/palta/15.jpg,Palta,0.686111111,0.409375,,,0.940277778,0.572916667										
472	TRAIN,/content/drive/MyDrive/frutas/palta/15.jpg,Palta,0.005555556,0.046875,,,1,1										
473	TRAIN,/content/drive/MyDrive/frutas/palta/15.jpg,Palta,0.293055556,0.473958333,,,0.554166667,0.685416667										
474	TRAIN,/content/drive/MyDrive/frutas/palta/15.jpg,Palta,0.556944444,0.619791667,,,0.902777778,0.854166667										
475	TRAIN,/content/drive/MyDrive/frutas/palta/158.jpg,Palta,0.1015625,0.172767204,,,0.6796875,0.866764275										
476	TRAIN,/content/drive/MyDrive/frutas/palta/192.jpg,Palta,0.55954323,0.061990212,,,1,0.628058728										
477	TRAIN,/content/drive/MyDrive/frutas/palta/192.jpg,Palta,0.23817292,0.138662316,,,0.619902121,0.632952692										
478	TRAIN,/content/drive/MyDrive/frutas/palta/192.jpg,Palta,0.161500816,0.535073409,,,0.391517129,0.753670473										
479	TRAIN,/content/drive/MyDrive/frutas/palta/192.jpg,Palta,0.556280587,0.562805873,,,0.812398042,0.72593801										
480	TRAIN,/content/drive/MyDrive/frutas/palta/192.jpg,Palta,0,0.55954323,,,0.208809135,0.783034258										
481	TEST,/content/drive/MyDrive/frutas/palta/133.jpg,Palta,0,0.002083333,,,1,1										
482	TEST,/content/drive/MyDrive/frutas/palta/133.jpg,Palta,0.1921875,0.26875,,,0.7515625,0.777083333										
483	TEST,/content/drive/MyDrive/frutas/palta/7.jpg,Palta,0.015714286,0.277185501,,,0.408571429,0.897654584										
484	TEST,/content/drive/MyDrive/frutas/palta/7.jpg,Palta,0.258571429,0.509594883,,,0.677142857,1										
485	TEST,/content/drive/MyDrive/frutas/palta/156.jpg,Palta,0,0.018333333,,,0.975,1										
486	TEST,/content/drive/MyDrive/frutas/palta/52.jpg,Palta,0.371428571,0.114285714,,,0.957142857,0.9										
487	TEST,/content/drive/MyDrive/frutas/palta/95.jpg,Palta,0.115555556,0.3025,,,0.472592593,0.510833333										
488	TEST,/content/drive/MyDrive/frutas/palta/95.jpg,Palta,0.521481481,0.391666667,,,0.893333333,0.5775										
489	TEST,/content/drive/MyDrive/frutas/palta/95.jpg,Palta,0.299259259,0.641666667,,,0.724444444,0.866666667										
490	TEST,/content/drive/MyDrive/frutas/palta/157.jpg,Palta,0.505,0.095,,,0.90375,0.85										

3.2.3. Elegir modelo

Se hace uso de la arquitectura de modelo EfficientDet-Lite4, ya que este se adecua al problema que queremos resolver. EfficientDet-Lite [0-4] es una familia de modelos de detección de objetos aptos para dispositivos móviles derivados de la arquitectura EfficientDet-Lite.

Figura 13

Arquitectura de EfficientDet



Nota: Utiliza EfficientNet como red troncal, BiFPN como la red característica y red predicción de clase/caja compartida. Adaptada de *EfficientDet architecture – It*, por Sensio, 2020 (https://juansensio.com/blog/048_cv_detection).

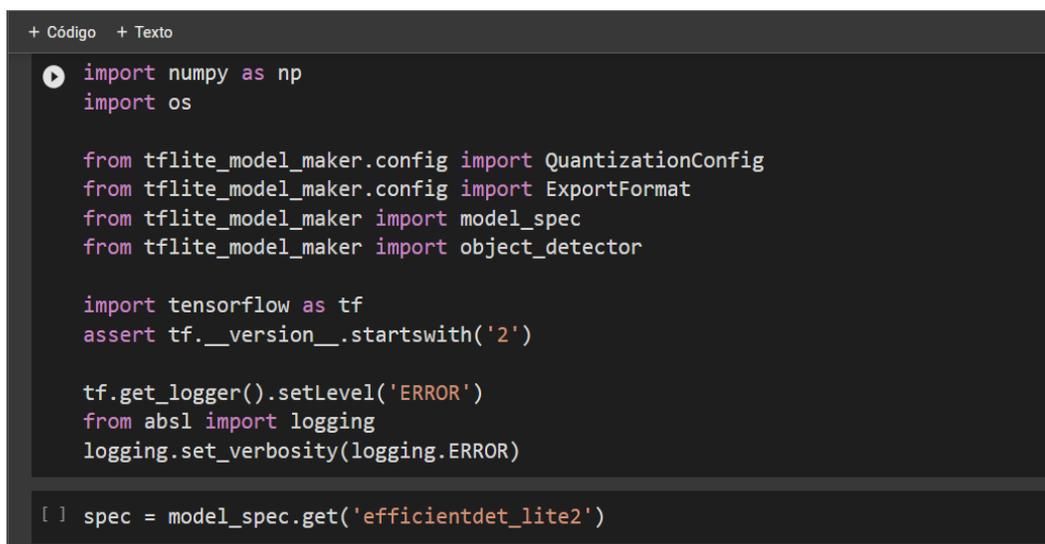
Tanto las capas BiFPN (red piramidal de características bidireccional ponderada) como las capas de red clase, se repiten varias veces en función de diferentes restricciones de recursos.

3.2.4. Entrenar modelo

Ahora se entrenará el modelo usando la herramienta Google colab, con el conjunto de datos preparado en el paso anterior.

Figura 14

Entrenamiento del modelo en Colaboratory



```

+ Código + Texto
▶ import numpy as np
import os

from tflite_model_maker.config import QuantizationConfig
from tflite_model_maker.config import ExportFormat
from tflite_model_maker import model_spec
from tflite_model_maker import object_detector

import tensorflow as tf
assert tf.__version__.startswith('2')

tf.get_logger().setLevel('ERROR')
from absl import logging
logging.set_verbosity(logging.ERROR)

[ ] spec = model_spec.get('efficientdet_lite2')

```

Antes de ejecutar el algoritmo se debe indicar algunos parámetros:

El parámetro “Epochs” indica el número de veces que se ejecutara el algoritmo EfficientDet-Lite4. Se puede observar la precisión de la validación durante el entrenamiento y detenerse antes para evitar el sobreajuste.

El parámetro “batch_size” indica la cantidad de pasos que se necesitan para recorrer la cantidad de imágenes del conjunto de datos.

El parámetro “train_whole_model=True”, es para ajustar todo el modelo en lugar de solo entrenar la capa principal para mejorar la precisión. Esto puede llevar más tiempo en entrenar el modelo.

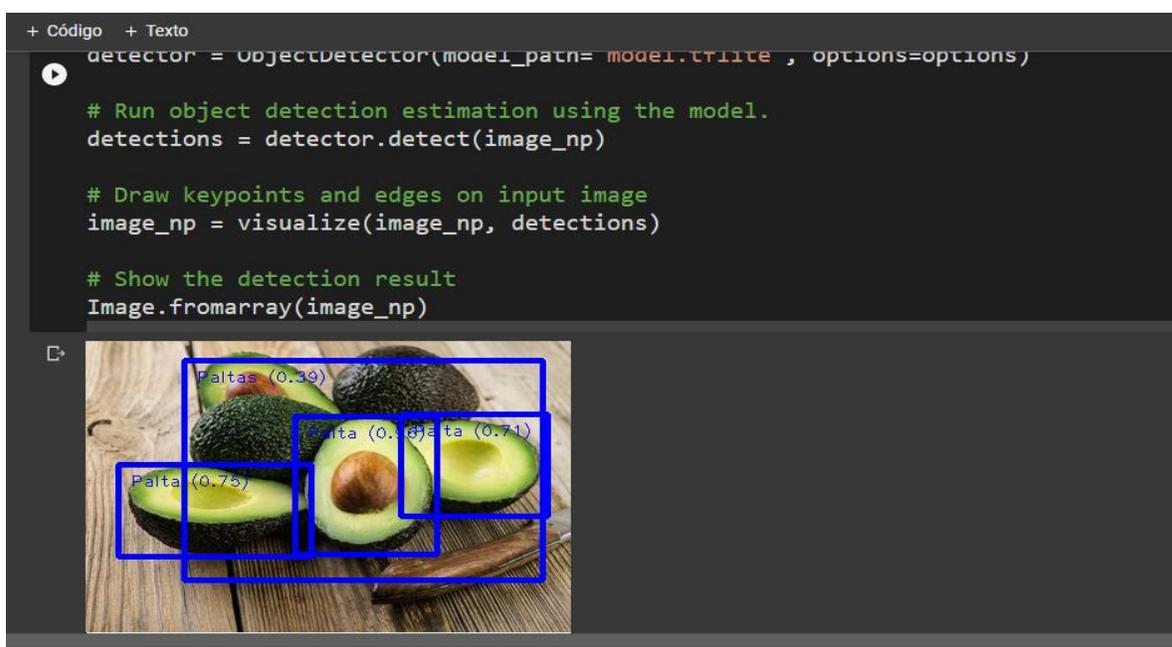
3.2.5. Evaluar modelo

Se debe evaluar el modelo creado con los datos de evaluación (conjunto de imágenes de prueba 10% del total) que contiene entradas que el modelo desconoce, para verificar la precisión del modelo ya entrenado. Si la exactitud es menor o igual al 50% el modelo no será útil y será necesario volver a iterar configurando algunos parámetros al momento de entrenar,

si se alcanza un 70% o más se considerará el modelo confiable.

Figura 15

Verificación del modelo con imágenes de prueba



Finalmente se exportará el modelo como model.tflite, para posteriormente integrarlo a la aplicación móvil.

3.3. DESARROLLO DE LA METODOLOGÍA DE SOFTWARE

3.3.1. Exploración

3.3.1.1. Recopilación de la información. La recopilación de datos de acuerdo a la metodología Mobile-D requiere, en principio, establecer reuniones con las personas interesadas. En esta fase inicial, se realizaron entrevistas y reuniones con la directora del colegio y se aplicaron cuestionarios sobre el tema de la educación nutricional en los estudiantes del colegio.

Reuniones con la Directora del Colegio. Se llevaron a cabo varias reuniones con la directora del colegio, quien proporcionó una visión general sobre las necesidades y expectativas del proyecto. Durante estas reuniones, se discutieron los objetivos de la

aplicación móvil, la importancia de motivar el aprendizaje en los estudiantes de primero de secundaria y las posibles características que la aplicación debería incluir para ser efectiva.

Cuestionarios a los Maestros. Se formularon y distribuyeron cuestionarios entre los maestros de la unidad educativa. Estos cuestionarios tenían como objetivo identificar las necesidades educativas específicas relacionadas con la educación nutricional y los alimentos naturales. Los maestros proporcionaron información valiosa sobre los problemas actuales que enfrentan los estudiantes en cuanto a su conocimiento y hábitos alimenticios (ver Anexo D)

Resultados de las Entrevistas y Cuestionarios. Los problemas detectados a partir de las respuestas de los cuestionarios incluyeron una falta de interés y conocimiento en los estudiantes sobre la importancia de una alimentación saludable. Los maestros también señalaron la necesidad de herramientas educativas interactivas y accesibles que puedan captar la atención de los estudiantes y facilitar el aprendizaje (Ver anexo 3).

Con base en los resultados obtenidos del cuestionario realizado a los maestros, se justifica la creación de una aplicación móvil orientada a la Educación Nutricional interactiva y educativa. A continuación, se muestra un resumen de los resultados obtenidos:

Tabla 5*Resultados de los cuestionarios*

<p>Nivel de Interés en la Nutrición (pregunta 5):</p>	<ul style="list-style-type: none"> ○ Muy interesados: 0 ○ Interesados: 0 ○ Poco interesados: 2 ○ Nada interesados: 3 	<p>Interpretación: Un alto porcentaje de estudiantes muestra poco o ningún interés en temas de nutrición, lo que indica la necesidad de métodos más atractivos y modernos, como una aplicación móvil, para fomentar su interés.</p>
<p>Frecuencia de Inclusión de Temas de Nutrición en Clases (pregunta 2):</p>	<ul style="list-style-type: none"> ○ Frecuentemente: 0 ○ Ocasionalmente: 0 ○ Raramente: 4 ○ Nunca: 1 	<p>Interpretación: La mayoría de los maestros abordan temas de nutrición raramente, lo que sugiere que los estudiantes reciben información insuficiente sobre el tema. Una aplicación móvil podría complementar la educación nutricional, proporcionando información constante y accesible.</p>
<p>Interés de los estudiantes por métodos educativos interactivos (pregunta 7):</p>	<ul style="list-style-type: none"> ○ Si: 5 ○ No: 0 ○ No estoy seguro: 0 	<p>Interpretación: Los resultados indican que los estudiantes tienen más interés en recursos educativos interactivos, Una aplicación móvil puede proporcionar información y actividades interactivas.</p>
<p>Percepción de la Oferta de Alimentos en el Colegio (pregunta 8):</p>	<ul style="list-style-type: none"> ○ Principalmente alimentos naturales y saludables: 0 ○ Mezcla equilibrada de alimentos naturales y procesados: 1 ○ Principalmente alimentos procesados: 4 	<p>Interpretación: Si un alto porcentaje de maestros percibe que el colegio vende principalmente alimentos procesados, es evidente la necesidad de educar a los estudiantes sobre la elección de alimentos saludables, lo que una aplicación móvil puede facilitar.</p>

Conclusión. Los resultados del cuestionario revelan una clara necesidad de mejorar la educación nutricional en los estudiantes. La implementación de una aplicación móvil puede ofrecer:

- Información interactiva y atractiva sobre alimentos naturales y su valor nutricional.
- Actividades y juegos educativos que mantengan el interés de los estudiantes.

En resumen, la fase de exploración permitió identificar las necesidades clave y los desafíos a abordar en el desarrollo de la aplicación móvil. La información recopilada será fundamental para guiar las siguientes fases del proyecto, asegurando que la aplicación cumpla con los objetivos educativos y motive a los estudiantes en su aprendizaje sobre nutrición y alimentos naturales.

3.3.1.2. Establecimiento de actores.

Tabla 6

Usuarios

USUARIO	DESCRIPCIÓN
Usuario de la aplicación móvil	Persona con un teléfono inteligente, que pretende informarse sobre los valores nutricionales de los alimentos naturales y que hará uso de la aplicación,

Tabla 7*Establecimiento de roles y responsabilidades*

ACTOR	ROL	RESPONSABILIDADES
Desarrollador de la aplicación	Diseñador y desarrollador de la aplicación móvil	<ul style="list-style-type: none"> • Diseño de la interfaz de usuario. • Desarrollo de la funcionalidad de la aplicación. • Pruebas y resolución de problemas.
Profesores	Facilitadores educativos	Integración de la aplicación en el plan de estudios.

3.3.1.3. Definición del alcance.**Limitaciones:**

- Esta aplicación se limitará a dispositivos móviles Android, con versión 8.0 en adelante.
- Esta aplicación va directamente dirigido a estudiantes de Primero de Secundaria (12 y 15 años).
- La aplicación móvil, al utilizar la cámara del dispositivo móvil, solo reconocer frutas y verduras.

La aplicación móvil no podrá:

- Reconocer o informar sobre alimentos compuestos (conformado por diversas sustancias alimenticias), como son: galletas, panes, cereales, tortas, yogurt, etc.
- Reconocer un conjunto de sustancias alimenticias procesadas o preparadas, como son: ensaladas, jugos, arroz con pollo, hamburguesa, etc.
- Estará disponible solo en idioma español.

3.3.1.4. Especificación de requisitos. Para cumplir los objetivos del sistema, las funcionalidades que la aplicación debe proporcionar a los usuarios son los siguientes:

Tabla 8*Requisitos funcionales:*

ID	DESCRIPCIÓN
RF1	La aplicación móvil debe contar con una pantalla para usar la cámara del dispositivo.
RF2	La aplicación debe reconocer en tiempo real la fruta o verdura que se ve a través de la cámara.
RF3	Visualizar información detallada sobre el alimento detectado.
RF4	La aplicación debe contar con una pantalla donde se pueda consultar la información de los alimentos.
RF5	La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional.
RF6	La aplicación debe estar disponible en la plataforma Play Store.
RF7	La aplicación debe contar con técnicas de gamificación.
RF8	La aplicación debe contar con un módulo para el cálculo del índice de masa corporal
RF9	La aplicación debe funcionar sin necesidad de conexión a internet
RF10	La aplicación debe tener una opción de retroalimentación para que los usuarios puedan reportar problemas o sugerencias.
RF11	La aplicación debe ofrecer opciones de accesibilidad, como texto a voz, para usuarios con discapacidades visuales.

Tabla 9*Requerimientos no funcionales*

ID	DESCRIPCIÓN
RNF1	La aplicación debe responder a las solicitudes internas en menos de 5 segundos.
RNF2	Deberá ser compatible con versiones a partir de la versión 8.0
RNF3	La aplicación debe adaptarse a diferentes resoluciones de pantallas.
RNF4	El tiempo de carga inicial de la aplicación no debe exceder los 3 segundos en dispositivos móviles con al menos 2 GB de RAM

3.3.2. Inicialización

3.3.2.1. Establecimiento de recursos. Durante del desarrollo de la esta aplicación se utilizará un ordenador con sistema operativo Windows 10 de 64 bits, junto a los requisitos mínimos que exige Android Studio para el uso eficiente de esta herramienta.

- Windows 8/10/11 (64 bits).
- 8 GB de RAM recomendado (3 GB mínimo).
- 8 GB de espacio disponible en el disco como mínimo (IDE + SDK de Android + Android Emulador).
- Resolución mínima de 1280 x 800.
- Dispositivo móvil con sistema operativo Android para ejecutar la aplicación.

3.3.2.2. Software requerido. El software requerido para esta aplicación es:

- IDE Android Studio versión Arctic Fox 2020.3.1
- Java SE 8
- Python 3.9
- Plataforma Roboflow
- Google Colaboratory

3.3.3. Producción

En esta fase se debe repetir iterativamente la programación (planificación – trabajo – liberación) hasta implementar todas las funcionalidades. A continuación, se describen las iteraciones para el desarrollo de la aplicación.

3.3.3.1. Iteración 1.

Planificación

Tabla 10

Planificación Iteración 1

ITERACIÓN 1	CARACTERÍSTICAS
Interfaz al ingresar a la aplicación.	<p>Al ingresar a la aplicación móvil, el usuario accederá a una interfaz principal intuitiva y fácil de navegar, donde podrá seleccionar entre las siguientes seis opciones:</p> <ol style="list-style-type: none"> 1. Cámara: Activar la cámara del dispositivo para escanear y reconocer alimentos en tiempo real. 2. Información de Alimentos: Acceder a una base de datos detallada sobre diferentes tipos de alimentos, incluyendo su valor nutricional y beneficios para la salud. 3. Mini Juego 1: Participar en un divertido mini juego educativo enfocado en el aprendizaje de nutrición. 4. Mini Juego 2: Disfrutar de otro mini juego interactivo diseñado para reforzar los conocimientos sobre alimentación saludable. 5. Mini Juego 3: Jugar un tercer mini juego que combina entretenimiento y educación nutricional para mantener el interés y motivación del usuario. 6. Calculadora Índice de Masa Corporal.

Trabajo

Se realizó el diseño y desarrollo de la interfaz principal, la cual consiste en:

- Botón 1, para seleccionar y usar la cámara del dispositivo.
- Botón 2, para consultar la información de los alimentos.
- Botón 3, mini juego uno.
- Botón 4, mini juego dos.
- Botón 5, mini juego tres.
- Botón 6. Calcular Índice de Masa Corporal (IMC)

Figura 16

Clase MainActivity.kt (interfaz principal)

```
22 class MainActivity : AppCompatActivity() {
23
24     override fun onCreate(savedInstanceState: Bundle?) {
25         super.onCreate(savedInstanceState)
26         setContentView(R.layout.activity_main)
27         hideSystemUI()
28     }
29
30
31     fun tomarFoto(view: View?) {
32         val intent = Intent(packageContext: this, ActivityCamara::class.java)
33         startActivity(intent)
34     }
35
36     fun Aprender(view: View?) {
37         val intent = Intent(packageContext: this, ActivityAprender::class.java)
38         startActivity(intent)
39     }
40
41     fun Jugar1(view: View?) {
42         val intent = Intent(packageContext: this, ActivityJuego1::class.java)
43         //val intent = Intent(this, ActivityMemoria::class.java)
44         startActivity(intent)

```

MainActivity > Aprender()

Liberación

Figura 17

Interfaz principal



3.3.3.2. Iteración 2. En esta iteración, se describe el proceso de elaboración de la interfaz que utilizará la cámara del dispositivo. También se integrará el modelo de reconocimiento previamente explicado, y se empleará realidad aumentada para visualizar una breve información del alimento.

Planificación

Tabla 11

Planificación Iteración 2

ITERACIÓN 2	CARACTERÍSTICAS
<p>Botón 1 de la interfaz principal.</p>	<p>Este módulo incluirá una funcionalidad avanzada que utiliza la cámara del dispositivo para reconocer los alimentos que el usuario está observando. Además, ofrecerá la opción de desplegar información detallada sobre el alimento seleccionado mediante realidad aumentada. Esta característica permitirá a los usuarios:</p> <ul style="list-style-type: none"> • Activar la cámara del dispositivo para escanear cualquier alimento. • Reconocer automáticamente el alimento a través de la cámara. • Acceder a información nutricional y educativa sobre el alimento en tiempo real. • Visualizar datos interactivos y enriquecidos sobre el alimento usando realidad aumentada, proporcionando una experiencia de aprendizaje más inmersiva y atractiva.

Trabajo

Se llevó a cabo el diseño y desarrollo de la interfaz que utilizará la cámara de la aplicación. Para ello, se realizaron las siguientes tareas:

- **Creación de la Interfaz de Cámara:** Se diseñó una interfaz intuitiva y fácil de usar que permite a los usuarios activar la cámara y escanear los alimentos. La interfaz es capaz de reconocer automáticamente los alimentos que están siendo observados por la cámara.

- Integración del Modelo de Detección de Alimentos: Se integró un modelo avanzado de detección de alimentos utilizando la Biblioteca de Tareas TFLite. Este modelo permite identificar con precisión los alimentos capturados por la cámara.
- Diseño de Objetos para Realidad Aumentada: Se diseñaron objetos interactivos para la realidad aumentada, los cuales se mostrarán sobre los alimentos reconocidos. Estos objetos proporcionan información detallada y visualmente atractiva sobre el valor nutricional y otros datos relevantes del alimento.

Estas tareas garantizaran que la interfaz de cámara sea funcional, eficiente y ofrezca una experiencia de usuario enriquecida mediante el uso de tecnología de realidad aumentada y modelos de detección de alimentos.

Figura 18

Clase ObjectDetectorHelper.kt (donde se integrará el modelo)

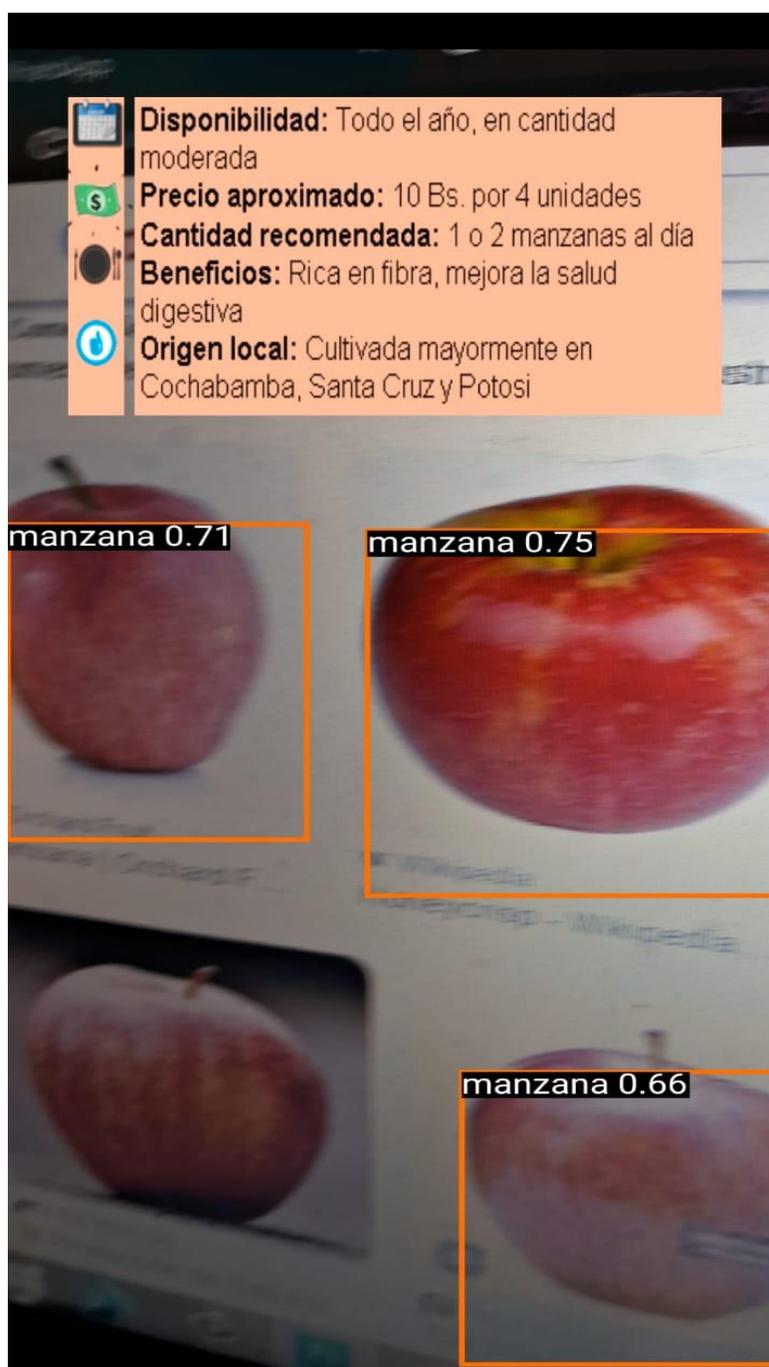
```
 3  import ...
 4
 5
 6
 7
 8
 9
10
11
12
13
14
15
16
17  class ObjectDetectorHelper(
18      var threshold: Float = 0.5f,
19      var numThreads: Int = 2,
20      var maxResults: Int = 3,
21      var currentDelegate: Int = 0,
22      var currentModel: Int = 0,
23      val context: Context,
24      val objectDetectorListener: DetectorListener?
25  ) {
26
27      private var objectDetector: ObjectDetector? = null
28
29      init {
30          setupObjectDetector()
31      }
32
33      fun clearObjectDetector() {
34          objectDetector = null
35      }
36
37      fun setupObjectDetector() {
```

ObjectDetectorHelper

Liberación

Figura 19

Pantalla que usa la cámara del dispositivo



3.3.3.3. Iteración 3. En esta iteración se realizará las interfaces donde se

visualizarán los alimentos organizados en grupos y algunos conceptos dentro de la nutrición.

Planificación

Tabla 12

Planificación Iteración 3

ITERACIÓN 3	CARACTERÍSTICAS
<p>Botón 2 de la interfaz principal.</p>	<ul style="list-style-type: none"> • Pantalla de Grupos de Alimentos: La aplicación mostrará una pantalla inicial con varios botones, cada uno representando un grupo de alimentos específico. Además, habrá un botón dedicado a desplegar conceptos clave en educación nutricional. • Navegación por Grupos de Alimentos: Al seleccionar uno de los botones de los grupos de alimentos, se abrirá una nueva interfaz que mostrará una lista completa de todos los alimentos pertenecientes al grupo seleccionado. Cada alimento incluirá información detallada sobre su valor nutricional y beneficios para la salud.

Trabajo

- Se desarrolló una interfaz con seis botones, cada botón representa un grupo de alimento. Además de un botón que desplegara conceptos clave en nutrición.
- Se desarrolló las interfaces en donde se despliegan los alimentos que se encuentran dentro de cada grupo.
- Botón de retroceso a la pantalla anterior.

Liberación

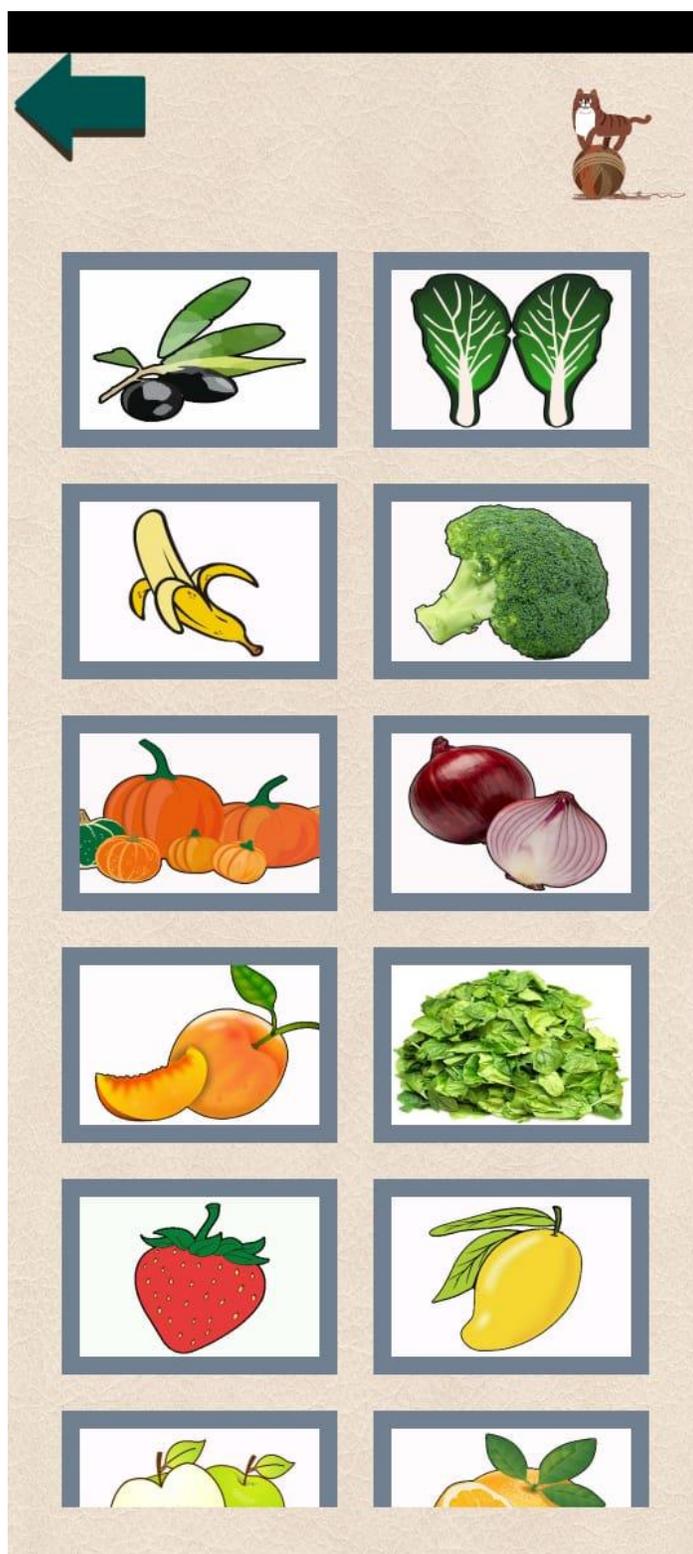
Figura 20

Pantalla de navegación por Grupos de Alimentos



Figura 21

Pantalla donde se visualizan los alimentos pertenecientes al grupo seleccionado



3.3.3.4. Iteración 4. En esta iteración se realizará la interfaz para mostrar la información de un alimento seleccionado.

Planificación

Tabla 13

Planificación Iteración 4

ITERACIÓN 4	CARACTERÍSTICAS
Alimento seleccionado de la interfaz	Al presionar un alimento (botón) se visualizará la información del alimento seleccionado, tales como los beneficios del alimento y la tabla nutricional. También se realizará una pantalla para desplegar conceptos clave en nutrición.

Trabajo

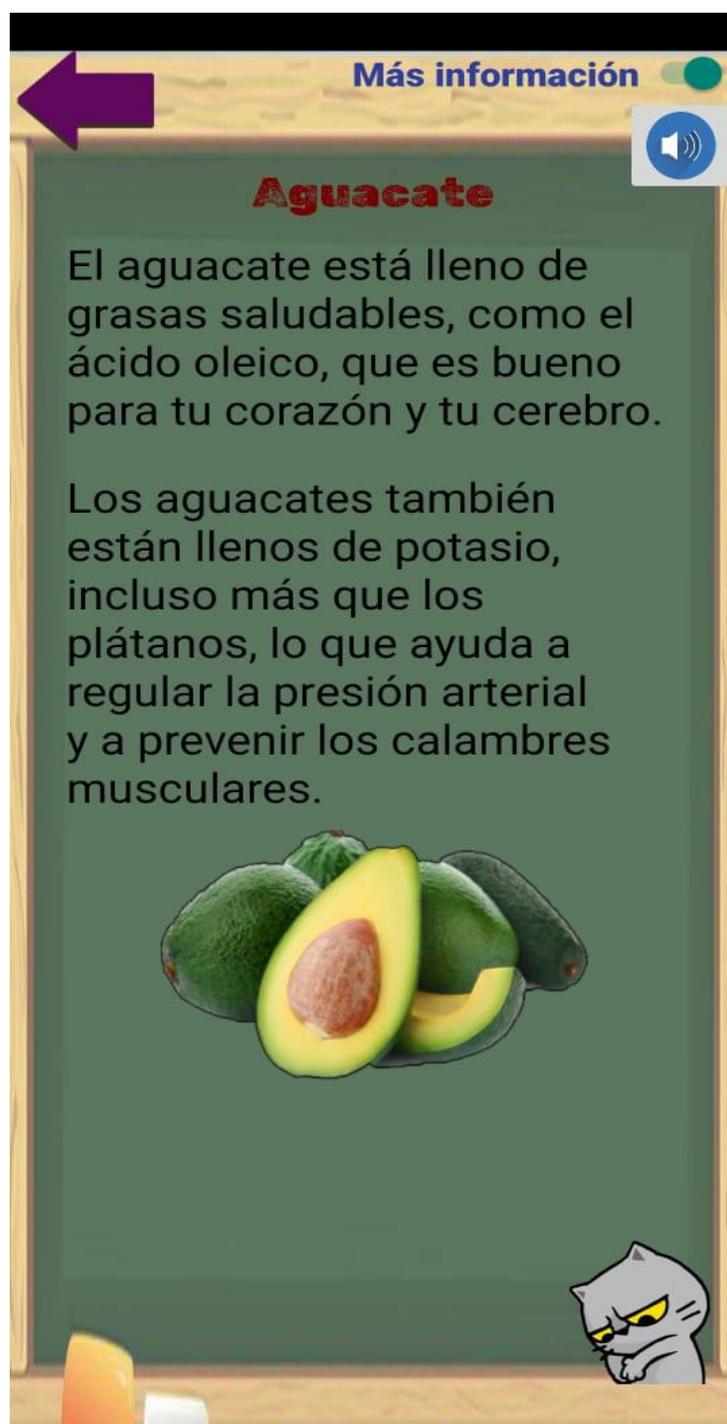
Las tareas que se realizaron son:

- Creación de la base de datos (SQLite) para almacenar y consultar información sobre los alimentos.
- Implementación y funcionalidad de la interfaz para visualizar la información de un alimento.
- Inclusión de un botón para retroceder a la pantalla anterior.
- Implementación y funcionalidad para mostrar la tabla nutricional del alimento seleccionado.
- Desarrollo de una pantalla para desplegar conceptos clave en nutrición.
- Inclusión de audios sobre los alimentos.

Liberación

Figura 22

Beneficios del alimento seleccionado



The image is a screenshot of a mobile application interface. At the top, there is a purple arrow pointing left and the text "Más información" next to a green toggle switch. Below this, the word "Aguacate" is written in a bold, red, serif font. The main content consists of two paragraphs of text on a dark green background. The first paragraph states that avocados are full of healthy fats like oleic acid, which is good for the heart and brain. The second paragraph mentions that avocados are also high in potassium, more than bananas, which helps regulate blood pressure and prevent muscle cramps. Below the text is an illustration of several avocados, including one cut in half to show the pit. In the bottom right corner, there is a small cartoon illustration of a grey cat with yellow eyes, looking towards the left. At the bottom left, there is a small orange slice and a white object, possibly a piece of paper or a small container.

Aguacate

El aguacate está lleno de grasas saludables, como el ácido oleico, que es bueno para tu corazón y tu cerebro.

Los aguacates también están llenos de potasio, incluso más que los plátanos, lo que ayuda a regular la presión arterial y a prevenir los calambres musculares.

Figura 23

Tabla nutricional del alimento seleccionado



Más información

Aguacate 100 g

Nutriente	Cantidad	% VD Hombre
Calorías	160.00 kcal	8.00%
Grasa Total	15.00 g	21.43%
Grasa Monoinsaturada	9.79 g	40.79%
Grasa Saturada	2.12 g	10.60%
Potasio	485.00 mg	16.17%
Fósforo	52.00 mg	4.16%
Magnesio	29.00 mg	7.07%
Cobre	190.00 mcg	21.35%
Manganeso	0.20 mg	9.09%
Vitamina K	21.00 mcg	28.00%
Vitamina C	10.00 mg	13.33%
Vitamina E	2.10 mg	14.00%
Vitamina B6	0.25 mg	19.23%
Vitamina B5	1.38 mg	27.60%
Vitamina B9	81.00 mcg	20.25%

Los porcentajes de Valores Diarios están basados en un consumo de 2000 Kcal. Sus valores diarios pueden ser menores o mayores dependiendo de sus necesidades calóricas. Los valores recomendados se basan en las directrices proporcionadas por la Organización Mundial de la Salud (OMS) y el Instituto Nacional de Salud (NIH)

3.3.3.5. Iteración 5.

Planificación

Tabla 14

Planificación Iteración 5

ITERACIÓN 5	CARACTERÍSTICAS
<p>Botón 3 de la interfaz principal</p>	<p>Este módulo incluirá un mini juego diseñado para encontrar pares de imágenes de alimentos antes de que se acabe el tiempo. Al terminar el juego se desplegará información o dato sobre algún alimento. Este juego no solo es divertido, sino que también ofrece múltiples beneficios educativos y cognitivos:</p> <ul style="list-style-type: none"> • Entrenamiento de la Memoria Visual: Los jugadores deben recordar la ubicación de las imágenes, lo que mejora la memoria visual. • Ejercicio para el Cerebro: La actividad desafía al cerebro, promoviendo el ejercicio mental y la agilidad cognitiva. • Mejora de la Concentración: El juego requiere atención y concentración, ayudando a los usuarios a desarrollar estas habilidades. • Educación Nutricional: Después de completar cada juego, la aplicación ofrecerá una oportunidad para aprender más sobre un alimento aleatorio. Esta característica proporcionará información valiosa y educativa para complementar la experiencia del juego. Algunas opciones para la visualización de información podrían incluir: ficha de alimento, curiosidad, recetas saludables.

Trabajo

Las tareas que se realizaron dentro de esta iteración son:

- Desarrollo de Interfaz de Niveles para el Mini Juego:

Se diseñó y desarrolló una interfaz que muestra varios niveles para el mini juego. El usuario tendrá la capacidad de seleccionar cualquier nivel disponible.

- Implementación de Funcionalidad de Niveles:

Al seleccionar un nivel, se inicia el juego con un número inicial de imágenes. A medida que el usuario avanza en el juego, el número de imágenes aumenta progresivamente. Además, se garantizó que las imágenes estén relacionadas con diferentes tipos de alimentos.

- Pantalla de Visualización al Finalizar cada Juego:

Se agregó una pantalla de visualización que muestra el puntaje obtenido por el usuario al finalizar cada juego. Además, se proporciona información sobre algún alimento específico, enriqueciendo así la experiencia educativa y lúdica del juego.

Liberación

Figura 24

Pantalla juego 1



Figura 25

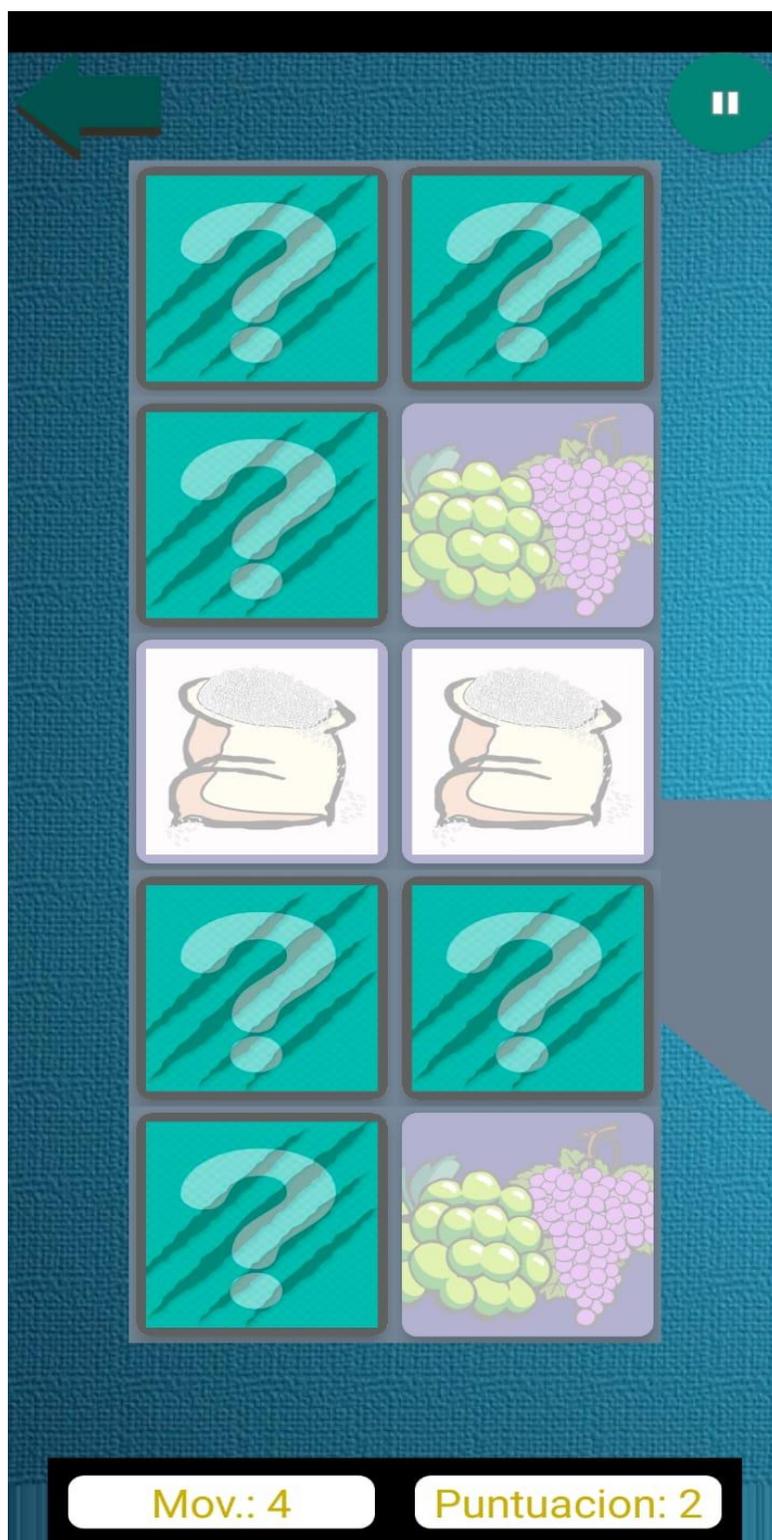
Funcionamiento del juego

Figura 26

Visualización de información de alimento al finalizar cada juego



3.3.3.6. Iteración 6.

Planificación

Tabla 15

Planificación Iteración 6

ITERACIÓN 6	CARACTERÍSTICAS
<p>Botón 4 de la interfaz principal</p>	<p>La aplicación contará con un emocionante mini juego de rompecabezas diseñado específicamente para promover la motricidad y la coordinación de los ojos y las manos, al tiempo que ofrece una experiencia educativa sobre nutrición. Estas son las características clave del juego:</p> <ul style="list-style-type: none"> • Temática Nutricional: Todas las imágenes utilizadas en el rompecabezas estarán relacionadas con alimentos saludables y nutrición, lo que permitirá a los usuarios aprender sobre diferentes tipos de alimentos mientras juegan. • Desarrollo de Habilidades Motoras: El juego proporciona una plataforma divertida y estimulante para mejorar la motricidad fina y la coordinación mano-ojo al manipular las piezas del rompecabezas. • Interfaz Atractiva y Accesible: La interfaz del juego será intuitiva y fácil de usar, con gráficos vibrantes y atractivos que cautivarán la atención de los usuarios y los mantendrán comprometidos con el juego.

Trabajo

- Se diseñó las imágenes que usaran en este mini juego.
- Se desarrolló una interfaz donde se visualizarán todas las imágenes que se puedan seleccionar para jugar el rompecabezas.
- Si el usuario presiona una de estas imágenes, se mostrará otra interfaz con la imagen seleccionada dividida en varias piezas, las cuales deben ser manipuladas para poder

armar una imagen.

- Botón de retroceso a la pantalla anterior.

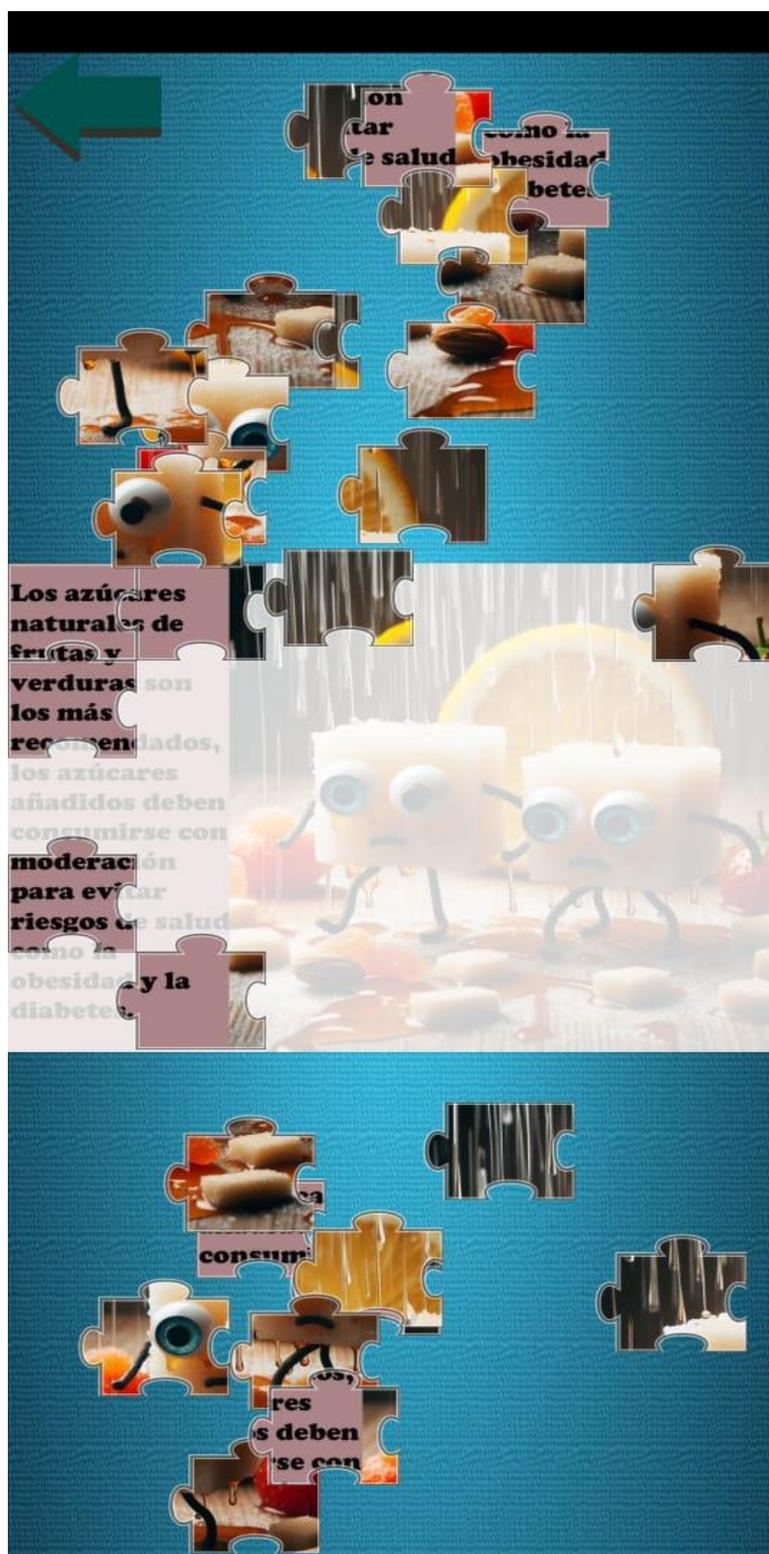
Liberación

Figura 27

Pantalla juego 2



Figura 28

Funcionamiento del juego 2

3.3.3.7. Iteración 7.

Planificación

Tabla 16

Planificación Iteración 7

ITERACIÓN 7	CARACTERÍSTICAS
<p>Botón 5 de la interfaz principal</p>	<p>Este módulo incluirá un mini juego interactivo de preguntas y respuestas que desafiará el conocimiento de los usuarios sobre temas relacionados con la nutrición y la alimentación saludable. Estas son las características clave del juego:</p> <ul style="list-style-type: none"> • Amplia Variedad de Preguntas: El juego ofrecerá una amplia gama de preguntas relacionadas con nutrición, alimentos saludables, grupos alimenticios, beneficios para la salud, y más. • Selección de Respuesta Múltiple: Cada pregunta presentará múltiples opciones de respuesta, de las cuales los usuarios deberán seleccionar la correcta. • Sistema de Puntuación: Los usuarios acumularán puntos basados en el número de respuestas correctas. Esto les permitirá realizar un seguimiento de su progreso y competir con ellos mismos.

Trabajo

Las tareas que se realizaron en este módulo son:

- Crear un diseño intuitivo y atractivo para la pantalla del juego que incluya el enunciado de la pregunta y las opciones de respuesta.
- Diseñar elementos visuales claros y distintivos para cada pregunta y opción de respuesta.
- Diseñar elementos gráficos adicionales para mejorar la estética y la experiencia del usuario durante el juego.
- Implementar la lógica para seleccionar aleatoriamente las preguntas de una base de datos o lista predefinida.
- Desarrollar la funcionalidad para mostrar el enunciado de la pregunta y las opciones de respuesta en la pantalla del juego.
- Crear la lógica para verificar si la respuesta seleccionada por el usuario es correcta o incorrecta.
- Implementar la funcionalidad para proporcionar retroalimentación inmediata al usuario sobre la respuesta seleccionada.
- Desarrollar la funcionalidad para llevar un registro de la puntuación del usuario a lo largo del juego.

Liberación

Figura 29

Pruebas de conocimiento

Pruebas de conocimiento

REPASAR	Nutrición Básica
REPASAR	Nutrición
	Nutrición

Figura 30

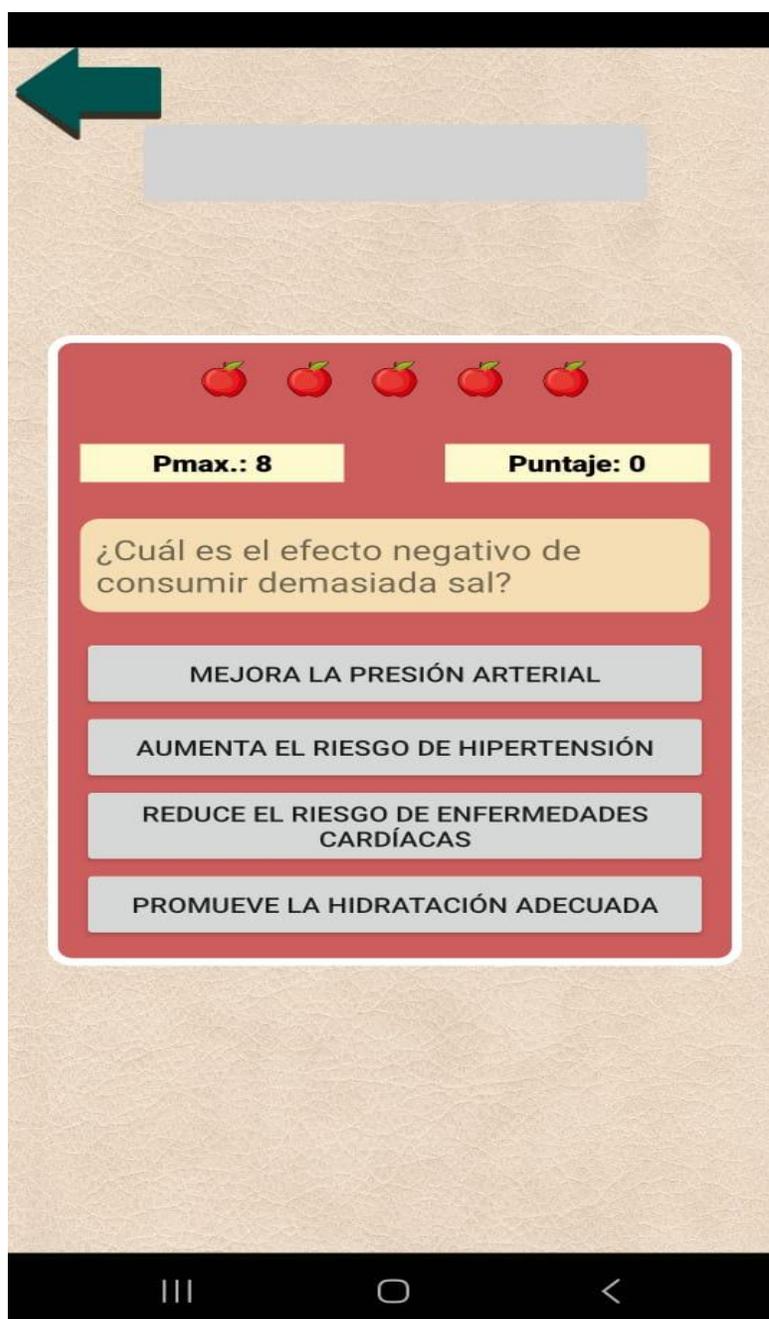
Funcionamiento juego 3

Figura 31

Algunos conceptos básicos sobre nutrición



NUTRICIÓN

La nutrición es el proceso biológico mediante el cual los organismos obtienen, asimilan y utilizan los nutrientes necesarios para el crecimiento, el mantenimiento y la reparación de los tejidos, y para la realización de las funciones vitales. Incluye la ingesta de alimentos, su digestión, absorción, transporte, utilización y excreción.

1. Principales Componentes de la Nutrición

Macronutrientes: Nutrientes que el cuerpo necesita en grandes cantidades para proporcionar energía y construir estructuras corporales.

Carbohidratos: Fuente primaria de energía.
Proteínas: Esenciales para

3.3.3.8. Iteración 8.

Planificación

Tabla 17

Planificación Iteración 8

ITERACIÓN 7	CARACTERÍSTICAS
Botón 6 de la interfaz principal	Este módulo contará con una calculadora de índice de masa corporal (IMC), que permitirá al usuario ingresar su peso y altura para realizar el cálculo. Basándose en el resultado del IMC obtenido, se ofrecerán recomendaciones personalizadas correspondientes a su estado de salud y composición corporal.

Trabajo

Las tareas que se realizaron en este módulo son:

- Implementación de la lógica necesaria para calcular el IMC utilizando la fórmula estándar: $IMC = \text{peso (kg)} / (\text{altura (m)})^2$.
- Desarrollo de código para validar los datos ingresados por el usuario y realizar el cálculo del IMC en función de estos datos.
- Desarrollo de código para determinar la categoría de peso del usuario en función del resultado del IMC calculado.
- Desarrollo de lógica para proporcionar recomendaciones personalizadas al usuario basadas en su categoría de peso y otros factores relevantes (por ejemplo, recomendaciones dietéticas, consejos de ejercicio, etc.).

Liberación

Figura 32

Interfaz para el cálculo del IMC



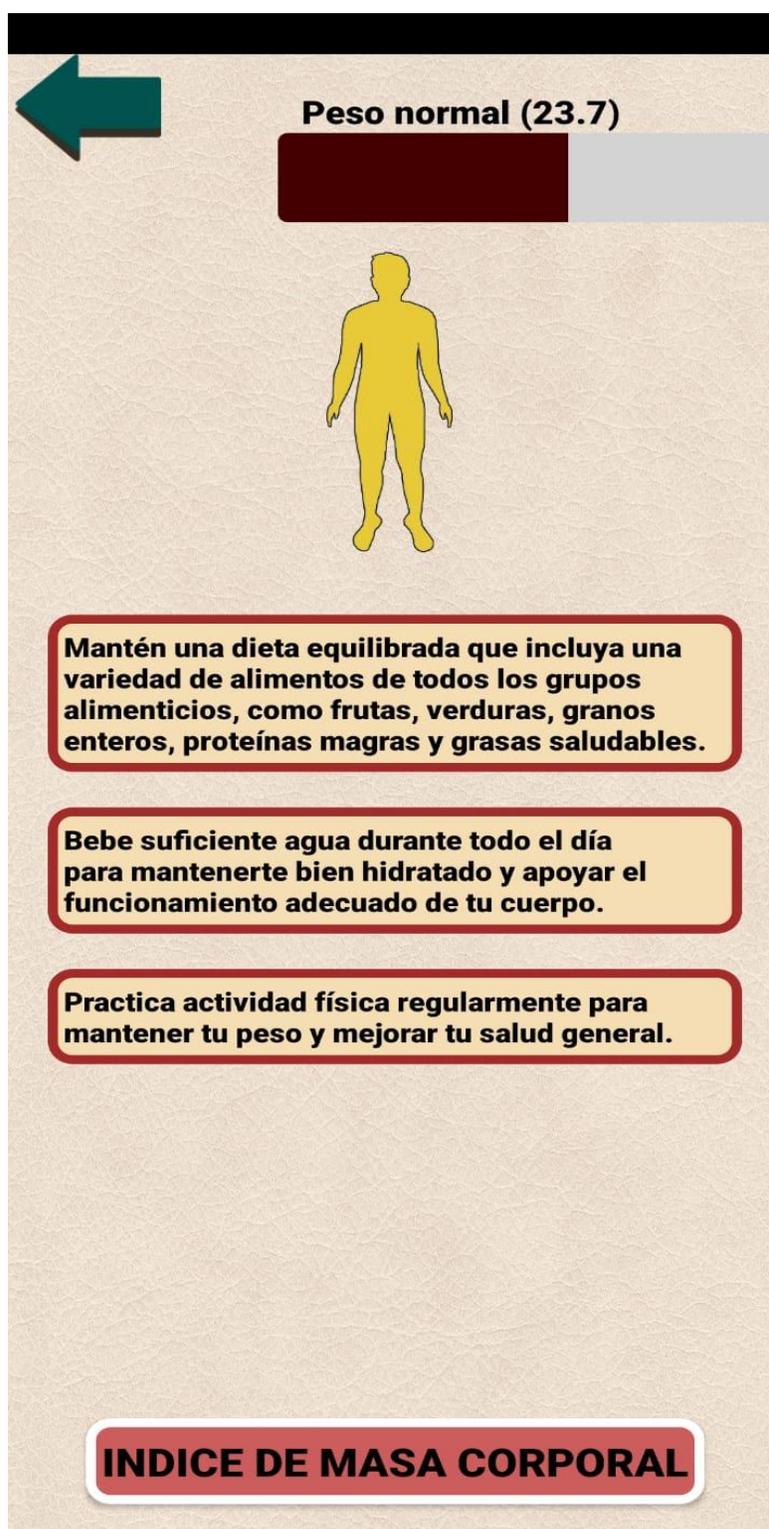
INDICE DE MASA CORPORAL

Edad:	25
Altura en cm:	148
Peso en kg:	52

CALCULAR IMC

Figura 33

Recomendaciones de acuerdo a la categoría de peso



3.3.4. Estabilización

En esta fase se aseguró que todos los módulos desarrollados funcionen correctamente y estén integrados de manera adecuadas en la aplicación final. además, se llevó a cabo la elaboración de la documentación de la presente investigación. Se realizaron las siguientes tareas:

Verificación del Funcionamiento de los Módulos:

- Se realizaron pruebas exhaustivas en cada módulo individualmente para confirmar que funcionan según lo previsto.
- Se verificó que todas las funciones y características de cada módulo respondan adecuadamente y sin errores.
- Se identificó y corrigió cualquier problema o fallo encontrado.

Elaboración de Documentación:

- Se preparó la documentación detallada sobre cada módulo y su funcionamiento.
- Se documentó los resultados de las pruebas realizadas en cada módulo, incluyendo los errores encontrados y las soluciones aplicadas.

Integración de los Módulos en la Aplicación:

- Se estableció un plan de integración para unir todos los módulos desarrollados en una única aplicación cohesiva.
- Se realizaron pruebas de integración para verificar que los módulos se comunican entre sí de manera efectiva y que la aplicación funcione como se espera.
- Se aseguró que la interfaz de usuario sea consistente en toda la aplicación y que los diferentes módulos se complementen entre sí de manera lógica y fluida.
- Se corrigió cualquier problema de compatibilidad o conflicto que surja durante la integración de los módulos.

3.3.5. Pruebas

En la fase de pruebas, el objetivo principal es asegurarse de que la aplicación sea completamente funcional, libre de errores y cumpla con los requisitos establecidos, todas las tareas de esta fase se describen y desarrolla en el Cap. 4 del presente informe. Algunas actividades importantes que se realizaron en esta fase son:

3.3.5.1. Actividades en la fase de pruebas:

Pruebas de Funcionalidad:

- Se verificó que todas las funciones de la aplicación, como la calculadora de IMC, el mini juego y la visualización de información nutricional, funcionen correctamente según lo esperado.
- Se realizó pruebas exhaustivas en todas las características de la aplicación para asegurarse de que respondan adecuadamente a las entradas del usuario y produzcan resultados precisos.

Pruebas de Usabilidad:

- Se evaluó la facilidad de uso de la aplicación, observando cómo los usuarios interactúan con ella y si encuentran algún obstáculo o confusión en la navegación.
- Se recolectó comentarios de los usuarios sobre la experiencia de uso de la aplicación y utilizar esta retroalimentación para realizar mejoras en la interfaz de usuario si es necesario.

Pruebas de Seguridad:

- Se realizó pruebas de seguridad para identificar posibles vulnerabilidades en la aplicación, como la protección de datos del usuario y la prevención de accesos no autorizados.

Prueba de Compatibilidad

Se realizó esta prueba para verificar que la aplicación móvil funciona en dispositivos móviles de diferentes dimensiones, resoluciones, resoluciones y versiones. Para esto se tomó

en cuenta dispositivos reales con diferentes características de software y hardware. A continuación, se describe las características de los dispositivos donde se verificó el cumplimiento de esta prueba y su funcionalidad correcta.

Tabla 18

Prueba de compatibilidad

Marca	Modelo	Versión Android	Memoria RAM	Resolución de pantalla
Samsung	Galaxy A32	11	4 GB	1080 x 2400 px
Samsung	Galaxy A20	9	3 GB	720 x 1560 px
Samsung	Galaxy S10	9.0	8 GB	1440 x 3040 px

CAPÍTULO IV

4. CALIDAD, COSTO, SEGURIDAD Y PRUEBAS

4.1. MÉTRICA DE CALIDAD

Como métrica de calidad de software, se utilizará la ISO/IEC 25010, ya que es una herramienta esencial para evaluar y mejorar la calidad del software en cualquier proyecto. Proporciona un enfoque integral, orientado al usuario y basado en estándares para garantizar que el software cumpla con los más altos niveles de calidad y satisfacción del cliente. La norma ISO 25010 define ocho características de calidad de software, los cuales son: funcionalidad, fiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad, seguridad y compatibilidad.

4.1.1. Usabilidad

En la siguiente tabla se presentan los criterios de usabilidad expresados en porcentajes, reflejando el nivel de comprensibilidad de la aplicación para el usuario. Además, se muestra el puntaje final de usabilidad de la app. Esta evaluación se llevó a cabo mediante una encuesta dirigida a 8 estudiantes (ver Anexo E), abordando aspectos como la facilidad de uso, la comprensión y la capacidad de aprendizaje del sistema.

Tabla 19*Resultados encuesta Usabilidad*

No.	PREGUNTA	SI	NO	PORCENTAJE
1	¿La navegación dentro de la aplicación es intuitiva y fácil de entender?	7	1	87.5%
2	¿Encuentras fácilmente la información que buscas dentro de la aplicación?	6	2	75%
3	¿Las instrucciones y la información proporcionada en la aplicación son claras y fáciles de entender?	6	2	75%
4	¿Es fácil comprender los conceptos relacionados con la educación nutricional presentados en la aplicación?	6	2	75%
5	¿El diseño visual de la aplicación es atractivo y estéticamente agradable?	8	0	100%
6	¿Las funciones interactivas de la aplicación, como los juegos y las calculadoras, son fáciles de usar?	7	1	87.5%
7	En general, ¿calificarías tu experiencia con la aplicación en términos de usabilidad como positiva?	7	1	87.5%
8	¿Recomendarías esta aplicación a otros estudiantes interesados en aprender sobre nutrición?	8	0	100%
PROMEDIO				86%

De acuerdo a los datos recopilados en la tabla anterior, podemos concluir que: la aplicación móvil alcanza un nivel de usabilidad del 86%. Esto significa que, de cada 100 usuarios que utilizan la aplicación, 86 indican que es fácil de comprender y manejar.

4.1.2. Funcionalidad

La evaluación de la funcionalidad consiste en determinar si la Aplicación móvil cumple con los requisitos funcionales previstos. El propósito es detectar posibles problemas y errores en su funcionamiento, garantizando su conformidad con el comportamiento deseado por el usuario.

Tabla 20

Requisitos funcionales cumplidos que se han implementado

	DESCRIPCIÓN	%
FUNCIÓN 1	La aplicación móvil debe contar con una pantalla para usar la cámara del dispositivo.	100%
FUNCIÓN 2	La aplicación debe reconocer en tiempo real la fruta o verdura que se ve a través de la cámara.	82%
FUNCIÓN 3	Visualizar información detallada sobre el alimento detectado.	95%
FUNCIÓN 4	La aplicación debe contar con una pantalla donde se pueda consultar la información de los alimentos.	92%
FUNCIÓN 5	La aplicación debe poder utilizarse sin necesidad de instalar ningún software adicional.	100%
FUNCIÓN 6	La aplicación debe estar disponible en la plataforma Play Store.	100%
FUNCIÓN 7	La aplicación debe contar con técnicas de gamificación	96%
FUNCIÓN 8	La aplicación debe contar con un módulo para el cálculo del índice de masa corporal	100%
FUNCIÓN 9	La aplicación debe funcionar sin necesidad de conexión a internet	100%
TOTAL		96%

$$\text{Porcentaje de Funcionalidad} = (9/9) \times 100 = 100\%$$

Esto significa que la aplicación móvil tiene un 100% de funcionalidad, lo que indica que todos los requisitos funcionales han sido cumplidos satisfactoriamente.

4.1.3. Confiabilidad

La confiabilidad permite evaluar la relación entre el nivel de funcionalidad y la cantidad de recursos utilizados. Esto incluye la medición del tiempo durante el cual el software está disponible para su uso y la probabilidad de que presente fallas. La evaluación de la confiabilidad se basa en dos aspectos principales:

- Comportamiento con respecto al tiempo: Incluye atributos del software relacionados con los tiempos de respuesta y el procesamiento de datos.
- Comportamiento con respecto a los recursos: Incluye atributos del software relacionados con la cantidad de recursos utilizados y la duración de su uso durante la realización de funciones.

Donde se fija lo siguiente:

$$P(T \leq t) = F(t) \quad \text{probabilidad de hallar una falla}$$

$$P(T < t) = 1 - F(t) \quad \text{probabilidad de no hallar una falla}$$

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla
Conociendo la funcionalidad del 100% del sistema calculamos para el periodo establecido.

Donde:

Funcionalidad = 1

$\lambda = 0.01$ (es decir 1 error en cada 10 ejecuciones)

T = 20

Hallamos la confiabilidad del sistema:

$$F(t) = (\text{Funcionalidad}) * e^{-\lambda t}$$

$$F(12) = 1 * e^{-(1/10) * 20}$$

$$F(12) = 0.135 * 100 = 13.5$$

Reemplazando en la fórmula de probabilidades tenemos lo siguiente:

$$P(T \leq t) = F(t)$$

$$P(T \leq t) = 1 - F(t)$$

$$P(T \leq t) = 0.135 = 13.5\%$$

$$P(T \leq t) = 1 - 0.135$$

$$P(T \leq t) = 0.865 = 86.5\%$$

Interpretando los datos se puede decir que la confiabilidad es de 86.5% en promedio de 20 días como tiempo de prueba.

4.1.4. Mantenibilidad

La mantenibilidad se refiere a los atributos que permiten evaluar el esfuerzo necesario para modificar la aplicación o software, ya sea para corregir errores o para agregar nuevas funcionalidades. Para evaluar la mantenibilidad del sistema, se utiliza el Índice de Madurez de Software (IMS), que indica la estabilidad de un producto de software. El IMS se determina mediante la siguiente fórmula:

Ecuación 1

Índice de Madurez de Software

$$IMC = \frac{Mt - (Fc + Fa + FE)}{MT}$$

Donde:

Mt: Numero de módulos total de la versión actual.

Fc: Numero de módulos de la versión actual que se cambiaron.

Fa: Numero de módulos de la versión actual que se añadieron.

FE: Numero de módulos de la versión anterior que se eliminaron en la versión actual.

$$IMC = \frac{6 - (1 + 0 + 0)}{6} = 0.83$$

Para la interpretación del resultado multiplicamos por 100 para sacar el porcentaje, de la siguiente forma:

$$\text{MANTENIBILIDAD} = 0.83 * 100$$

$$\text{MANTENIBILIDAD} = 83\%$$

De esta forma se concluye que la aplicación móvil tiene una mantenibilidad del 83%.

4.1.5. Portabilidad

La portabilidad se refiere a la capacidad del software para ser trasladado de un entorno a otro, considerando los siguientes aspectos:

i. Adaptabilidad:

- **Compatibilidad con Diferentes Dispositivos:** Verifica si la aplicación funciona correctamente en una variedad de dispositivos con diferentes tamaños de pantalla, resoluciones y capacidades de hardware.
- **Compatibilidad con Diferentes Sistemas Operativos:** Asegura de que la aplicación sea compatible con las versiones principales de los sistemas operativos móviles (por ejemplo, Android e iOS).
- **Requisitos de Software:** Evalúa si la aplicación necesita dependencias específicas y si estas están disponibles en diferentes entornos.

ii. Facilidad de Instalación:

- **Proceso de Instalación:** Evalúa la simplicidad y claridad del proceso de instalación de la aplicación en diferentes dispositivos y sistemas operativos.
- **Tamaño del Archivo:** Considera el tamaño del archivo de la aplicación y su impacto en la facilidad de descarga e instalación.

iii. Conformidad:

- **Adherencia a Estándares:** Verifica si la aplicación cumple con los estándares y convenciones establecidos para el desarrollo de aplicaciones móviles, como las directrices de las tiendas de aplicaciones (Google Play Store, Apple App Store).
- **Licencias y Permisos:** Asegura de que la aplicación cumpla con todas las licencias y permisos necesarios para operar en diferentes entornos.

Tabla 21

Resultados de las pruebas de portabilidad de la aplicación

FACTOR EVALUADO	ESCENARIO DE PRUEBA	RESULTADO	PORCENTAJE
Adaptabilidad	<ul style="list-style-type: none"> Prueba en diferentes dispositivos. 	Aprobado	95%
Facilidad de Instalación	<ul style="list-style-type: none"> Proceso de instalación. Tamaño del archivo. 	Aprobado	92%
Conformidad	<ul style="list-style-type: none"> Adherencia a estándares. Licencias y permisos 	Aprobado	90%
TOTAL			90.4%

4.2. ESTIMACIÓN DEL COSTO

Para estimar el costo de desarrollo de la aplicación móvil se utilizó el modelo COCOMO II, debido a razones como: precisión en la Estimación, adaptabilidad a Diferentes Proyectos, transparencia y justificación. Para esto COCOMO II utiliza la siguiente formula:

Ecuación 2

Fórmula que utiliza COCOMO II

$$Esfuerzo (PM) = a * (Tamaño)^b * EAF$$

Donde:

- **a** es una constante empírica, el valor típico es 2.4.
- **b** es un exponente que representa la escala, el valor típico es 1.05.
- **Tamaño** se mide en KLOC (mil líneas de código).
- **EAF** (Effort Adjustment Factor) es el producto de multiplicar los factores de costo.

Tabla 22

Tabla de coeficientes

Proyecto de software	a	b	c	d
Orgánico	2.4	1.05	2.2	0.38
Semi – acoplado	3.0	1.12	2.5	0.35
Empotrado	3.80	1.20	2.5	0.32

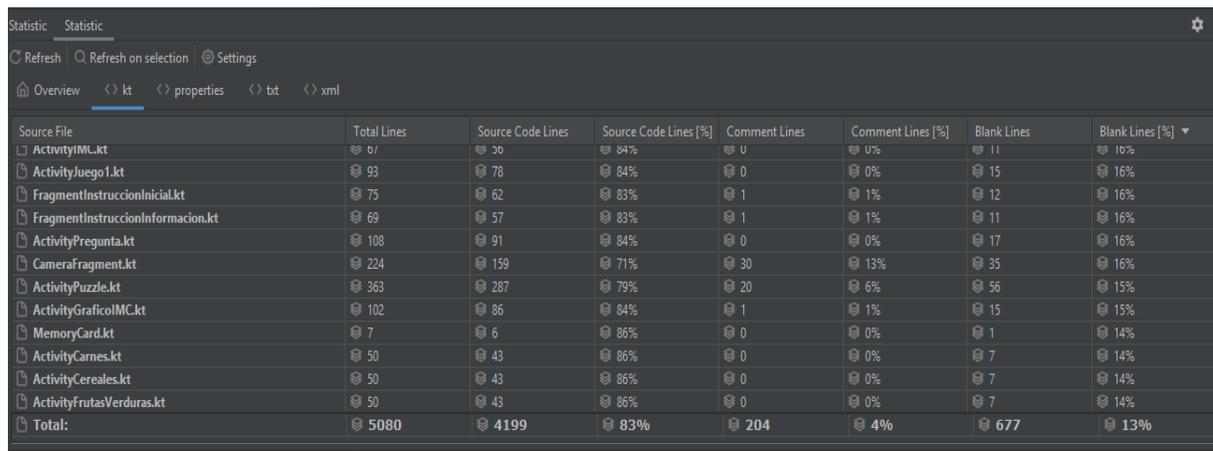
Nota: (Pressman, 2010)

Calcular el *Tamaño* en KLOC:

Se utilizó el pluggin Statistic para obtener el número de líneas de código, dando como resultado 5080 LOC.

Figura 34

Cálculo de líneas de código con Statistic



Source File	Total Lines	Source Code Lines	Source Code Lines [%]	Comment Lines	Comment Lines [%]	Blank Lines	Blank Lines [%]
ActivityIML.kt	0/	0	84%	0	0%	11	10%
ActivityJuego1.kt	93	78	84%	0	0%	15	16%
FragmentInstruccionInicial.kt	75	62	83%	1	1%	12	16%
FragmentInstruccionInformacion.kt	69	57	83%	1	1%	11	16%
ActivityPregunta.kt	108	91	84%	0	0%	17	16%
CameraFragment.kt	224	159	71%	30	13%	35	16%
ActivityPuzzle.kt	363	287	79%	20	6%	56	15%
ActivityGraficoIMC.kt	102	86	84%	1	1%	15	15%
MemoryCard.kt	7	6	86%	0	0%	1	14%
ActivityCarnes.kt	50	43	86%	0	0%	7	14%
ActivityCereales.kt	50	43	86%	0	0%	7	14%
ActivityFrutasVerduras.kt	50	43	86%	0	0%	7	14%
Total:	5080	4199	83%	204	4%	677	13%

Ecuación 3

Ecuación para calcular tamaño en KLOC

$$KLOC = \frac{LOC}{1000}$$

$$KLOC = \frac{5080}{1000} = 5.08$$

4.2.1. Cálculo del esfuerzo y tiempo del proyecto

A continuación, se hace el cálculo del esfuerzo necesario para la programación de la aplicación. La ecuación para este cálculo es:

Ecuación 4

Esfuerzo expresado en personas por mes

$$\text{Esfuerzo (PM)} = a * (\text{KLOC})^b, \text{ en personas por mes}$$

Donde:

E es el esfuerzo expresado en personas por mes

a es una constante empírica igual a 2.4

b es una constante empírica igual a 1.05

KLOC es el número estimado de líneas de código

Reemplazando en la ecuación anterior tenemos:

Ecuación 5

Esfuerzo persona por mes

$$\text{Esfuerzo} = 2.4 * (5.08)^{1.05}$$

$$\text{Esfuerzo} = 13.22 \frac{\text{personas}}{\text{mes}}$$

Calculando el tiempo de desarrollo con $c = 2.2$ y $d = 0.38$:

Ecuación 6

Tiempo de desarrollo en meses

$$\text{Tiempo Desarrollo} = c(\text{Esfuerzo})^d \text{ en meses}$$

$$\text{Tiempo Desarrollo} = 2.2 * (13.22)^{0.38}$$

$$\text{Tiempo Desarrollo} = 5.86 \cong 6 \text{ meses}$$

Cálculo de personal requerido para el desarrollo de la app:

Ecuación 7

Cantidad de personal requerido para el desarrollo

$$P = \frac{\text{Esfuerzo}}{\text{Tiempo Desarrollo}} \text{ en personas}$$

$$P = \frac{13.22}{5.86} \text{ en personas}$$

$$P = 2.25 \cong 2 \text{ programadores}$$

4.2.2. Costo total del proyecto

El salario promedio de un programador junior oscila entre los 2000 Bs y 4000 Bs., se tomará un promedio de 3000 Bs, a partir de este monto podemos calcular el costo del software desarrollado:

Ecuación 8

Costo del proyecto por mes

$$\text{Costo de desarrollo por mes} = P * \text{salario de un programador}$$

$$\text{Costo de desarrollo por mes} = 2 * 3000$$

$$\text{Costo de desarrollo por mes} = 6000 \text{ Bs.}$$

Como el desarrollo de software se lo estima en 6 meses tendremos:

Ecuación 9

Costo total del proyecto

$$\text{Costo total de proyecto} = \text{Costo de desarrollo por mes} * \text{Numero de meses}$$

$$\text{Costo total de proyecto} = 6000 * 6$$

$$\text{Costo total del proyecto} = 36000 \text{ Bs}$$

4.3. SEGURIDAD

Se adopto la norma ISO/IEC 27034, la cual es una norma internacional que aborda la

seguridad en el desarrollo de aplicaciones de software. Esta norma proporciona un marco integral para integrar la seguridad en todas las etapas del ciclo de vida del desarrollo de aplicaciones, desde la concepción hasta la operación y el mantenimiento.

- **Estandarización de la Seguridad.** Aunque la aplicación es offline y no requiere inicio de sesión, ISO/IEC 27034 proporciona un marco estandarizado para garantizar que todas las prácticas de seguridad se apliquen de manera consistente. Esto asegura que la app mantenga un nivel adecuado de protección contra amenazas.
- **Gestión de Riesgos.** Se identifico y mitigo riesgos potenciales que pueden afectar a la aplicación offline. Esto incluye amenazas como la manipulación de datos locales, acceso no autorizado al dispositivo, y ataques de malware que podrían comprometer la integridad de la aplicación.
- **Mejores Prácticas de Desarrollo Seguro.** Se siguieron directrices para integrar prácticas de seguridad durante todo el ciclo de vida del desarrollo de software, desde el diseño hasta el mantenimiento. Esto asegura que la aplicación se desarrolle con un enfoque de seguridad proactivo.

Ofuscación de Datos

Para mejorar la seguridad del software, se decidió utilizar herramientas de ofuscación de código como ProGuard y R8. Estas herramientas son esenciales para proteger la aplicación contra la ingeniería inversa, dificultando que los atacantes puedan entender y modificar el código fuente. A continuación, se detalla el proceso:

Configuración Inicial: Primero, se configura el entorno de desarrollo para usar ProGuard/R8. Dado que Android Studio ya incluye estas herramientas, se aseguró de que estuvieran activadas en el archivo build.gradle de la aplicación, añadiendo las siguientes líneas en la sección correspondiente del archivo:

Figura 35

Configuración en el archivo `build.gradle`

```
buildTypes {
    release {
        minifyEnabled true
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'progu
    }
}
```

Creación del Archivo de Reglas: Luego, se creó un archivo de reglas de ProGuard, `proguard-rules.pro`, donde se definió las configuraciones específicas para la ofuscación y optimización. Este archivo permitió especificar qué partes del código debían ofuscarse y cuáles debían mantenerse legibles, como las APIs públicas o ciertas clases críticas.

Ejecución de la Ofuscación: Al compilar la aplicación en modo de lanzamiento (release), ProGuard/R8 ejecutó automáticamente el proceso de ofuscación, optimización y reducción del código. Esto incluyó la eliminación de código no utilizado, la renombración de clases, métodos y variables a nombres más cortos y confusos, y la reorganización del código para hacerlo más eficiente.

Verificación y Pruebas: Después de la ofuscación, se realizó pruebas exhaustivas de la aplicación para asegurar de que funcionara correctamente y que no se hubieran introducido errores durante el proceso. Se verificó que todas las funcionalidades clave estuvieran intactas y que el rendimiento hubiera mejorado.

Despliegue: Finalmente, se empaquetó la aplicación ofuscada para su distribución. Este paso aseguró que el código resultante fuera más seguro y difícil de descompilar.

El uso de ProGuard/R8 no solo hizo que la aplicación fuera hasta un 90% más pequeña y hasta un 20% más rápida, sino que también añadió una capa significativa de seguridad al ofuscar los nombres de las clases, campos y métodos. Esto proporcionó una protección adicional contra la ingeniería inversa, dificultando que cualquier atacante pudiera comprender y modificar el código de la aplicación.

Encriptación de datos

Para asegurar la confidencialidad de los datos almacenados localmente en la aplicación, se utilizó SQLCipher para encriptar la base de datos SQLite. SQLCipher es una extensión de código abierto para SQLite que proporciona encriptación de datos usando el algoritmo AES-256. A continuación se detalla el proceso:

Configuración del Entorno: Primero, se integró SQLCipher en el proyecto Android añadiendo la dependencia necesaria en el archivo `build.gradle` de la aplicación.

Inicialización de SQLCipher: Luego, se aseguró que SQLCipher estuviera correctamente inicializado en mi aplicación. Esto implicó cargar las bibliotecas nativas de SQLCipher antes de acceder a la base de datos:

Creación y Apertura de la Base de Datos Encriptada: Se utilizó una contraseña para encriptar la base de datos al crearla o abrirla. Este paso aseguró que todos los datos almacenados estuvieran cifrados y protegidos:

Acceso y Manejo de Datos: Con la base de datos encriptada, se continuó accediendo y manejando los datos de manera normal, asegurando que todas las operaciones estuvieran cifradas de forma transparente:

Verificación y Pruebas: Después de implementar la encriptación, se realizaron pruebas exhaustivas para verificar que los datos estuvieran efectivamente encriptados y que la aplicación funcionara correctamente.

Despliegue: Finalmente, se desplegó la aplicación con la base de datos encriptada, asegurando que los datos estuvieran protegidos.

Implementar SQLCipher en la base de datos SQLite no solo añadió una capa crucial de seguridad al cifrar los datos almacenados, sino que también garantizó la integridad y confidencialidad de la información, protegiéndola contra accesos no autorizados.

4.4. PRUEBAS AL SOFTWARE

4.4.1. Pruebas de caja blanca

Para las pruebas de caja blanca se tomó en cuenta los algoritmos de lógica para el cálculo del Índice de Masa Corporal.

Se diseñaron casos de prueba para verificar la funcionalidad de cada componente, asegurando de cubrir:

- Pruebas de Condiciones: Verifica todas las condiciones en las estructuras de control.
- Pruebas de Caminos: Asegúrate de que todos los caminos posibles a través del código sean probados.
- Pruebas de Bucles: Valida la ejecución de bucles, incluyendo casos de cero iteraciones, una iteración y múltiples iteraciones.
- Pruebas de Valores Límites: Verifica cómo el código maneja valores en los límites de los rangos permitidos.

Caso de prueba para el método que calculara el índice de masa corporal:

Figura 36

Método para el cálculo del IMC

```
class EMICalculadora {  
    fun calcular(weight: Double, height: Double): Double {  
        if (height <= 0) {  
            throw IllegalArgumentException("La altura debe ser mayor a 0")  
        }  
        val bmi = weight / (height * height)  
        return String.format("%.2f", bmi).toDouble()  
    }  
  
    fun getCategoria(bmi: Double): String {  
        return when {  
            bmi < 18.5 -> "Peso bajo"  
            bmi < 24.9 -> "Peso normal"  
            bmi < 29.9 -> "Sobrepeso"  
            else -> "Obesidad"  
        }  
    }  
}
```

Figura 37

Pruebas unitarias para el módulo de cálculo del Índice de Masa Corporal utilizando Junit.

```

class BMICalculadoraTest {

    private val bmiCalculadora = BMICalculadora()

    @Test
    fun testCalculadoraBMI() {
        val weight = 70.0 // kg
        val height = 1.75 // meters
        val expectedBmi = 22.86
        val actualBmi = bmiCalculadora.calculate(weight, height)
        assertEquals(expectedBmi, actualBmi, 0.01)
    }

    @Test
    fun testCalcularBMIHeightCero() {
        val weight = 70.0 // kg
        val height = 0.0 // meters
        assertThrows(IllegalArgumentException::class.java) {
            bmiCalculadora.calculate(weight, height)
        }
    }

    @Test
    fun testGetCategoriaBajoPeso() {
        val bmi = 17.5
        val expectedCategory = "bajo peso"
        val actualCategory = bmiCalculadora.getCategory(bmi)
        assertEquals(expectedCategory, actualCategory)
    }

    @Test
    fun testGetCategoriaPesoNormal() {
        val bmi = 22.0
        val expectedCategory = "Peso normal"
        val actualCategory = bmiCalculadora.getCategoria(bmi)
        assertEquals(expectedCategory, actualCategory)
    }

    @Test
    fun testGetCategoriaSobrepeso() {
        val bmi = 27.0
        val expectedCategory = "Sobrepeso"
        val actualCategory = bmiCalculadora.getCategoria(bmi)
        assertEquals(expectedCategory, actualCategory)
    }

    @Test
    fun testGetCategoryObesidad() {
        val bmi = 32.0
        val expectedCategory = "Obesidad"
        val actualCategory = bmiCalculadora.getCategoria(bmi)
        assertEquals(expectedCategory, actualCategory)
    }
}

```

Explicación de las Pruebas:

- testCalculadoraBMI: Esta prueba verifica que el cálculo del IMC sea correcto para un peso de 70 kg y una altura de 1.75 m. La salida esperada es 22.86.
- testCalcularBMIHeightZero: Esta prueba verifica que si la altura es 0, se lance una excepción IllegalArgumentException. Esto es importante para asegurarse de que el método maneje adecuadamente entradas no válidas.
- testGetCategoriaBajoPeso: Esta prueba verifica que un IMC de 17.5 se clasifique correctamente como "Bajo peso".
- testGetCategoriaPesoNormal: Esta prueba verifica que un IMC de 22.0 se clasifique correctamente como "Peso normal".
- testGetCategoriaSobrepeso: Esta prueba verifica que un IMC de 27.0 se clasifique correctamente como "Sobrepeso".
- testGetCategoriaObesidad: Esta prueba verifica que un IMC de 32.0 se clasifique correctamente como "Obesidad".

Al ejecutar las pruebas en android studio, se verifico y confirmo que el código funciona correctamente.

4.4.2. Pruebas de caja negra

4.4.2.1. Módulo cámara del dispositivo para escanear y reconocer alimentos.

Caso de pruebas 1. Escaneo de alimento

- Título: Escaneo y reconocimiento de un alimento válido.
- Descripción: Verificar que la cámara del dispositivo puede escanear y reconocer correctamente un alimento conocido.
- Precondiciones: La cámara del dispositivo debe estar funcionando.
- Entradas: Imagen de una manzana.

- Pasos:
 - Abrir la aplicación y seleccionar la opción para escanear alimentos.
 - Enfocar la cámara en una manzana y escanear.
- Resultados Esperados: La aplicación reconoce la manzana y muestra una breve información sobre el alimento.

Para evaluar la efectividad del módulo de reconocimiento de alimentos en la aplicación móvil, se realizó pruebas mediante el escaneo de 10 imágenes para cada tipo de alimento, estas imágenes incluían tanto fotos reales como imágenes en libros e internet. Utilizando la aplicación, se escaneó cada imagen y fruta, para luego registrar los resultados en tres categorías: detectado correctamente, detectado incorrectamente y no reconocido. Los resultados se recopilaron en una tabla estadística que muestra la precisión del reconocimiento, así como los errores y fallas.

Tabla 23

Resultados del escaneo de frutas y verduras

Alimento	Total escaneado	Frutas reales	Frutas en imagen	Detectado correctamente	Detectado incorrectamente	No reconocido
Manzana	10	6	4	7	0	3
Banana	10	7	3	7	0	3
Sandía	10	6	4	8	0	2
Zanahoria	10	5	5	7	0	3
Palta	10	6	4	8	0	2
Tomate	10	5	5	7	1 (rabano)	2
Brócoli	10	5	5	8	0	2
Piña	10	5	5	9	0	1
Frutilla	10	7	3	9	0	1
Durazno	10	6	4	6	2 (mango)	2
PROMEDIO				7.6 (76%)	0.3 (3%)	2.1 (21%)

4.4.2.2. Módulo de información sobre alimentos

Caso de Prueba 1: Visualización de Información de un Alimento natural

- Título: Visualización de información nutricional de un alimento.
- Descripción: Verificar que la aplicación muestra correctamente la información de un alimento conocido.
- Precondiciones: El alimento debe estar registrado en la base de datos de la aplicación.
- Entradas: Solicitud de información sobre "Manzana".
- Pasos:
 - Abrir la aplicación y seleccionar la opción de información sobre alimentos.
 - Buscar y seleccionar "Manzana" en la lista de alimentos.
- Resultados Esperados: La aplicación muestra las ventajas y la tabla nutricional de la manzana.

Para evaluar la funcionalidad del módulo de información sobre alimentos, se seleccionaron alimentos representativos de cada uno de los seis grupos alimenticios y se realizaron pruebas de caja negra para verificar la correcta visualización de la información correspondiente. Cada prueba consistió en seleccionar un alimento específico en la aplicación y comprobar que se mostraran correctamente su imagen, ventajas nutricionales y tabla nutricional. Los resultados se registraron en una tabla que muestra si cada uno de estos aspectos se visualizó correctamente, identificando así las áreas de mejora en el módulo de información sobre alimentos. Esta metodología permite una evaluación precisa y detallada del funcionamiento del módulo, asegurando que la información proporcionada a los usuarios sea exacta y completa.

Tabla 24*Resultados sobre el módulo Información de Alimentos*

Alimento	Grupo	Ventajas Nutricionales Correctas	Tabla Nutricional Correcta	Resultado
Manzana	Frutas y Verduras	Sí	Sí	Correcto
Plátano	Frutas y Verduras	Si	Sí	Correcto
Brócoli	Frutas y Verduras	Sí	Sí	Correcto
Zanahoria	Frutas y Verduras	Sí	Si	Correcto
Pollo	Carnes	Sí	Sí	Correcto
Pescado	Carnes	Sí	Sí	Correcto
Arroz	Cereales	Sí	Sí	Correcto
Yogur	Lácteos	Sí	Sí	Correcto
Queso	Lácteos	Sí	Si	Correcto
Nueces	Frutos secos	Sí	Sí	Correcto

4.4.2.3. Calculadora de Índice de Masa Corporal (IMC)

Caso de Prueba :

- Entrada: Usuario ingresa peso y altura que resultan en un IMC dentro del rango normal.
- Salida Esperada: Se muestra el IMC calculado y un mensaje de recomendaciones saludables.

Caso de Prueba :

- Entrada: Usuario ingresa peso y altura que resultan en un IMC fuera del rango normal (sobrepeso o bajo peso).
- Salida Esperada: Se muestra el IMC calculado y un mensaje de recomendaciones específicas para el sobrepeso o bajo peso.

Para evaluar la funcionalidad del módulo de cálculo del índice de masa corporal (IMC) de la aplicación, se definieron diversos casos de prueba con combinaciones específicas de peso y altura, abarcando diferentes categorías de IMC: bajo peso, peso normal, sobrepeso y obesidad. En cada caso de prueba, se ingresaron los valores de edad, peso y altura en la aplicación y se verificó que el IMC se calculara correctamente, y que la categoría de IMC mostrada fuera precisa y que las recomendaciones personalizadas fueran adecuadas según el resultado. Los resultados se registraron en una tabla que muestra el peso y la altura ingresados, el IMC calculado, la categoría de IMC esperada, la categoría de IMC mostrada por la aplicación y si las recomendaciones fueron correctas. Este proceso permite evaluar la precisión y utilidad del módulo de IMC, asegurando que los usuarios reciban información y consejos precisos basados en sus datos ingresados.

Tabla 25

Resultados para el Caso de Prueba 2

Caso de Prueba	Peso (kg)	Altura (m)	IMC Calculado	Cat. IMC Esperada	Cat. IMC Mostrada	Recomendaciones Correctas	Resultado
1	50	1.60	19.53	Peso Normal	Peso Normal	Sí	Correcto
2	45	1.60	17.58	Bajo Peso	Bajo Peso	Sí	Correcto
3	70	1.70	24.22	Peso Normal	Peso Normal	Sí	Correcto
4	85	1.75	27.76	Sobrepeso	Sobrepeso	Sí	Correcto
6	110	1.70	38.06	Obesidad	Obesidad	Sí	Correcto
7	60	1.50	26.67	Sobrepeso	Sobrepeso	Sí	Correcto
8	75	1.60	29.30	Sobrepeso	Sobrepeso	Sí	Correcto
9	100	1.90	27.70	Sobrepeso	Sobrepeso	Sí	Correcto
10	80	1.55	33.28	Obesidad	Obesidad	Sí	Correcto

4.4.3. Pruebas de rendimiento

A continuación, se detallan las pruebas de rendimiento realizadas en la aplicación móvil. Se evaluaron tanto el rendimiento del dispositivo como el rendimiento de la aplicación para garantizar su eficiencia y estabilidad.

4.4.3.1. Pruebas de Rendimiento del Dispositivo

Tabla 26

Tiempo de arranque

DESCRIPCIÓN	DISPOSITIVO	RESULTADOS
Se evaluó el tiempo que tarda la aplicación en iniciarse completamente en diferentes dispositivos. Se realizaron múltiples pruebas para calcular el tiempo de arranque promedio.	Dispositivo 1	3.5 segundos
	Dispositivo 2	4.2 segundos
	Dispositivo 3	3.8 segundos

Tabla 27

Uso de Memoria

DESCRIPCIÓN	RESULTADOS
Se monitorizó el uso de memoria RAM y almacenamiento interno del dispositivo mientras la aplicación estaba en funcionamiento.	Uso promedio de memoria RAM: 150 MB Uso promedio de almacenamiento interno: 50 MB

4.4.3.2. Pruebas de Rendimiento de la Aplicación

Tabla 28

Tiempo de respuesta y estabilidad de la ampliación móvil

DESCRIPCIÓN	RESULTADOS
Tiempo de respuesta Se midió el tiempo de respuesta de la aplicación al realizar diversas acciones como navegar entre pantallas y realizar operaciones dentro de la aplicación.	Tiempo de respuesta promedio: 0.8 segundos

Estabilidad	Se crearon casos de prueba para evaluar la capacidad de la aplicación para manejar errores y excepciones sin colapsar.	La aplicación demostró una alta estabilidad y pudo manejar errores sin interrupciones significativas para el usuario.
--------------------	--	---

En conclusión, las pruebas de rendimiento indican que la aplicación móvil tiene un tiempo de arranque aceptable, un uso eficiente de la memoria y una buena capacidad de respuesta. Además, se observó que la aplicación es estable y puede manejar errores de manera efectiva.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Una vez finalizado el desarrollo e implementación de la “APLICACIÓN MÓVIL ORIENTADA A LA EDUCACIÓN NUTRICIONAL DE ALIMENTOS NATURALES PARA MOTIVAR EL APRENDIZAJE EN ESTUDIANTES DE PRIMERO DE SECUNDARIA”, para la Unidad Educativa “Mariscal de Zepita Mañana”, se ha logrado alcanzar el objetivo principal planteado a través de un enfoque innovador que combina la tecnología de Visión Artificial, Realidad Aumentada y técnicas de gamificación, la aplicación ha demostrado ser una herramienta efectiva para el aprendizaje eficiente de los estudiantes de primero de secundaria en el ámbito de la nutrición. Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Conocimiento Nutricional. La aplicación ha contribuido a aumentar el conocimiento nutricional de los estudiantes al proporcionar información detallada sobre los requerimientos relacionados con el valor nutricional, como vitaminas, fibras y minerales.
- La integración de un sistema de Visión Artificial ha permitido a los estudiantes identificar fácilmente alimentos naturales utilizando la cámara de sus dispositivos móviles. Esta funcionalidad ha mejorado significativamente la experiencia de usuario al proporcionar acceso instantáneo a información nutricional relevante sobre los alimentos escaneados.
- Se ha desarrollado una interfaz de usuario que es tanto intuitiva como amigable para los estudiantes de Primero de Secundaria. Esta interfaz facilita la interacción con la aplicación, asegurando que los usuarios jóvenes puedan navegar y utilizar las diferentes funcionalidades sin dificultad.
- Se han incorporado elementos de gamificación de forma exitosa, haciendo que el

aprendizaje sea más divertido e interesante para los estudiantes. Características como recompensas y niveles han incrementado el interés y la participación de los usuarios, haciendo que el proceso de aprendizaje sea más atractivo y efectivo.

- **Realidad Aumentada para Información Nutricional en Tiempo Real:** Las funcionalidades de Realidad Aumentada han permitido superponer información nutricional relevante sobre los alimentos identificados en tiempo real. Esta característica proporciona una manera visualmente atractiva de presentar datos teóricos, facilitando una mejor comprensión y retención de la información nutricional.

5.2. RECOMENDACIONES

- **Continuidad y Mejora.** Para garantizar la continuidad y el éxito a largo plazo de la aplicación, se recomienda mantener un proceso de actualización constante que incluya mejoras en la interfaz de usuario, incorporación de nuevas funcionalidades y actualización de contenidos nutricionales según las últimas investigaciones y recomendaciones.
- **Evaluación y Retroalimentación.** Es importante realizar evaluaciones periódicas del impacto de la aplicación en el conocimiento y comportamiento alimenticio de los usuarios, así como recopilar feedback y sugerencias para identificar áreas de mejora y nuevas oportunidades de desarrollo.
- **Promoción y Difusión.** Se recomienda promover activamente la aplicación entre la comunidad estudiantil, educadores y profesionales de la salud para aumentar su adopción y maximizar su impacto.
- **Colaboraciones Interdisciplinarias.** Para enriquecer aún más la aplicación y garantizar su relevancia y precisión, se sugiere establecer colaboraciones con expertos en nutrición, educadores y profesionales de la salud para incorporar sus conocimientos y experiencia en el desarrollo y actualización de la aplicación.

REFERENCIAS BIBLIOGRÁFICAS

- Amaya, Y. (2013). *Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual.*
- A. Mamani (2020). *Tutor Inteligente para el Aprendizaje de Matemáticas en Sexto de Primaria* (Tesis de Grado). Universidad Pública de El Alto, Bolivia. Repositorio Institucional – Universidad Pública de El Alto.
- Amazon Web Services (AWS). (2023). *¿Qué es el machine learning? - Explicación sobre el machine learning empresarial - AWS.* Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/machine-learning/>
- Android Studio. (2024, febrero 29). *Introducción a Android Studio.* Android Developers.
<https://developer.android.com/studio/intro?hl=es-419>
- ATLANTE. (2016). *Los beneficios del uso de las aplicaciones móviles en las instituciones.*
<http://www.eumed.net/rev/atlante/2016/11/aplicaciones.html>
- AWS. (2023). *¿Qué es el aprendizaje por transferencia?: Explicación del aprendizaje por transferencia en machine learning: AWS.* Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/transfer-learning/>
- Azuma, R. T. (1997). *A Survey of Augmented Reality.*
- Borrella Petisco, B. (2022). *Introducción a la visión artificial: Procesos y aplicaciones.*
<https://hdl.handle.net/20.500.14352/3290>
- Casas (2022). *Tutor Inteligente con Realidad Virtual para la Enseñanza de Conceptos y Teorías de la Biología Celular a Jóvenes Estudiantes de 15 a 18 años en las Unidades Educativas de La-Paz Bolivia* (Tesis de Grado). Universidad Mayor de San Andrés, Bolivia. Repositorio Institucional – Universidad Mayor de San Andrés.
- Caribe, C. E. para A. L. y el. (2018, abril 2). *Malnutrición en niños y niñas en América Latina y el Caribe* [Text]. Comisión Económica para América Latina y el Caribe.

- <https://www.cepal.org/es/enfoques/malnutricion-ninos-ninas-america-latina-caribe>
- Comité Municipal de Seguridad Alimentaria de Sucre. (2021).
- CIRAD et al., (2024). *PLANTNET* (3.4.4) [Aplicación móvil]. Google Play. Recuperado de https://play.google.com/store/search?q=plantnet&c=apps&hl=es_419&gl=US
- Corporation Microsoft, M. (s. f.). *¿Qué es la realidad aumentada (AR)? | Microsoft Dynamics* 365. Recuperado 27 de febrero de 2024, de <https://dynamics.microsoft.com/es-es/mixed-reality/guides/what-is-augmented-reality-ar/>
- De la Horra Villace, I. (2017). *Realidad Aumentada, una Revolución Educativa*.
- Edgar Mozas Fenoll (Director). (2016, abril 4). *TIC4.1: Qué es la realidad aumentada*. https://www.youtube.com/watch?v=jpd5E_Dh9PU
- Elvira y Peñalosa (2020). *Impacto de una App Móvil en la Promoción de Hábitos Alimenticios Saludables en estudiantes de Grado Séptimo de dos Instituciones Educativas Públicas* (Trabajo de Grado). Universidad Santander UDES, Colombia. Repositorio digital – Universidad de Santander
- ELIKA. (2018, diciembre 28). Nutrientes. *ELIKA Alimentación Saludable*. <https://alimentacionsaludable.elika.eus/nutrientes/>
- Fernandez Garcia, N. L. (s. f.). *Tema 1.- Introducción a la Visión Artificial—Visión Artificial Avanzada*.
- Gavin, M. L. (2022, mayo). *Grasas (para Padres)—Nemours KidsHealth*. <https://kidshealth.org/es/parents/fat.html>
- Gómez, A., Migani, S., & Otazú, A. (s. f.). *UN MODELO DE ESTIMACION DE PROYECTOS DE SOFTWARE*.
- Google. (s. f.). *Google Colab*. Recuperado 1 de marzo de 2024, de <https://research.google.com/colaboratory/intl/es/faq.html>
- Grapsas, T. (2019, diciembre 15). Realidad aumentada: ¿qué es, cómo funciona y para qué

- sirve? *Rock Content - ES*. <https://rockcontent.com/es/blog/realidad-aumentada/>
- Guillermo. (2024, marzo 26). *Alimentos Naturales*. <https://blog.joselito.com/es/blog/alimentos-naturales-caracteristicas-ejemplos>
- Herazo, L. (2022). *¿Qué es una aplicación móvil?* <https://anincubator.com/que-es-una-aplicacion-movil/>
- Herrera Bernal, J. A. (2008). *El m-learning o aprendizaje móvil se habla de algo espontaneo, informal, personal, aprendizaje situado, situaciones reales, etc., el m-learning se basa en instrucciones, utilizando audio vos, imágenes en movimiento y aprendizaje de campo.*
- Hurtado Soler, A. (2013). *LA SALUD*. <https://www.uv.es/hort/alimentacion/alimentacion.html>
- International Business Machines (IBM). (s. f.). *¿Qué es la visión artificial? | IBM*. Recuperado 22 de abril de 2024, de <https://www.ibm.com/mx-es/topics/computer-vision>
- Mefood (2020). *Oorenji & Ciencia y Nutrición (2.0.3)* [Aplicación móvil]. Google Play. Recuperado de https://play.google.com/store/search?q=oorenji&c=apps&hl=es_419&gl=US
- MedlinePlus. (2021, agosto 11). *¿Qué son las proteínas y qué es lo que hacen?: MedlinePlus Genetics*. <https://medlineplus.gov/spanish/genetica/entender/comofuncionangenes/proteina/>
- Mena, M. (2023, marzo 30). *Infografía: El mapa mundial de Android e iOS*. Statista Daily Data. <https://es.statista.com/grafico/29620/sistema-operativo-movil-con-la-mayor-cuota-de-mercado-por-pais>
- Mendoza (2020). *Tutor Inteligente Móvil para la Enseñanza de la Estática y Dinámica en Estudiantes de Quinto de Secundaria* (Proyecto de Grado). Universidad Pública de El Alto, Bolivia. Repositorio Institucional – Universidad Pública de El Alto.
- Ministerio de Salud. (2018, mayo 22). *Alimentación saludable*. Argentina.gob.ar.

<https://www.argentina.gob.ar/salud/alimentacion-saludable>

N. Mamani (2020). *Videojuego Educativo para Concientizar sobre la Importancia del Reciclaje* (Tesis de Grado). Universidad Mayor de San Andrés, Bolivia. Repositorio Institucional – Universidad Mayor de San Andrés.

NESTLE. (s. f.). *Todo sobre los minerales en la alimentación | Nestle contigo*. Recuperado 28 de febrero de 2024, de <https://www.nestle-contigo.co/elige-tu-medida/todo-sobre-los-minerales>

NormasISO.org. (2023a, mayo 23). *Guía ISO 27034: Seguridad de la Información en Aplicaciones*. NormasISO.org. <https://normasiso.org/norma-iso-27034/>

NormasISO.org. (2023b, septiembre 24). *ISO 25010: Mejora calidad y satisfacción del usuario en software*. NormasISO.org. <https://normasiso.org/norma-iso-25010/>

Organización Mundial de la Salud, FAO. (2007). *Glosario*.

Quiroz, A. (2022, junio 29). *¿Qué es una aplicación móvil y para qué sirve? B2Chat*. <https://www.b2chat.io/blog/marketing/aplicacion-movil-que-para-que-sirve/>

Recoletas Salud. (2023, febrero 24). *¿En qué se basa la educación nutricional?* Noticias Grupo Recoletas. <https://www.gruporecoletas.com/noticias/bases-educacion-nutricional/>

Rodríguez, M., Mancha, C.-L., Pedreira, Ó., & Fernández, C. M. (2015). *Certificación de la Mantenibilidad del Producto Software: Un Caso Práctico*. *Revista Latinoamericana de Ingeniería de Software*.

Rueda, J. F. V. (2019, noviembre 4). *CRISP-DM: Una metodología para minería de datos en salud*. *healthdataminer.com*. <https://healthdataminer.com/data-mining/crisp-dm-una-metodologia-para-mineria-de-datos-en-salud/>

Sanchez Arias, S. (2023, diciembre 13). *9 consecuencias de una mala alimentación en los niños*. Eres Mamá. <https://eresmama.com/consecuencias-una-mala-alimentacion-en-niños>

los-ninos/

Siles (2020). *Desarrollo de Aplicación Móvil para Identificar los Rasgos del Rostro y brindar Asesoramiento en Maquillaje, utilizando Inteligencia Artificial* (Proyecto de Grado). Universidad Mayor de San Simón, Bolivia. Repositorio – Universidad Mayor de San Simón.

Silva (2020). *Aplicación de Realidad Aumentada como Herramienta Orientada al Proceso de Enseñanza y Aprendizaje* (Tesis de Grado). Universidad Pública de El Alto, Bolivia. Repositorio Institucional – Universidad Pública de El Alto.

Sociedad Mexicana de Nutrición y Endocrinología. (2021). *Qué son los carbohidratos?*

Softcorp. (2019, febrero 16). *Definición y cómo funcionan las aplicaciones móviles.*

<https://servisoftcorp.com/definicion-y-como-funcionan-las-aplicaciones-moviles/>

TensorFlow. (2022, mayo 26). *Creador de modelos TensorFlow Lite*. TensorFlow.

https://www.tensorflow.org/lite/models/modify/model_maker?hl=es-419

Universidad Internacional de Valencia. (2021, abril 27). *Educación nutricional ¿Qué es? Y ¿Por qué es tan importante?* VIU Colombia.

<https://www.universidadviu.com/co/actualidad/nuestros-expertos/educacion-nutricional-que-es-y-por-que-es-tan-importante>

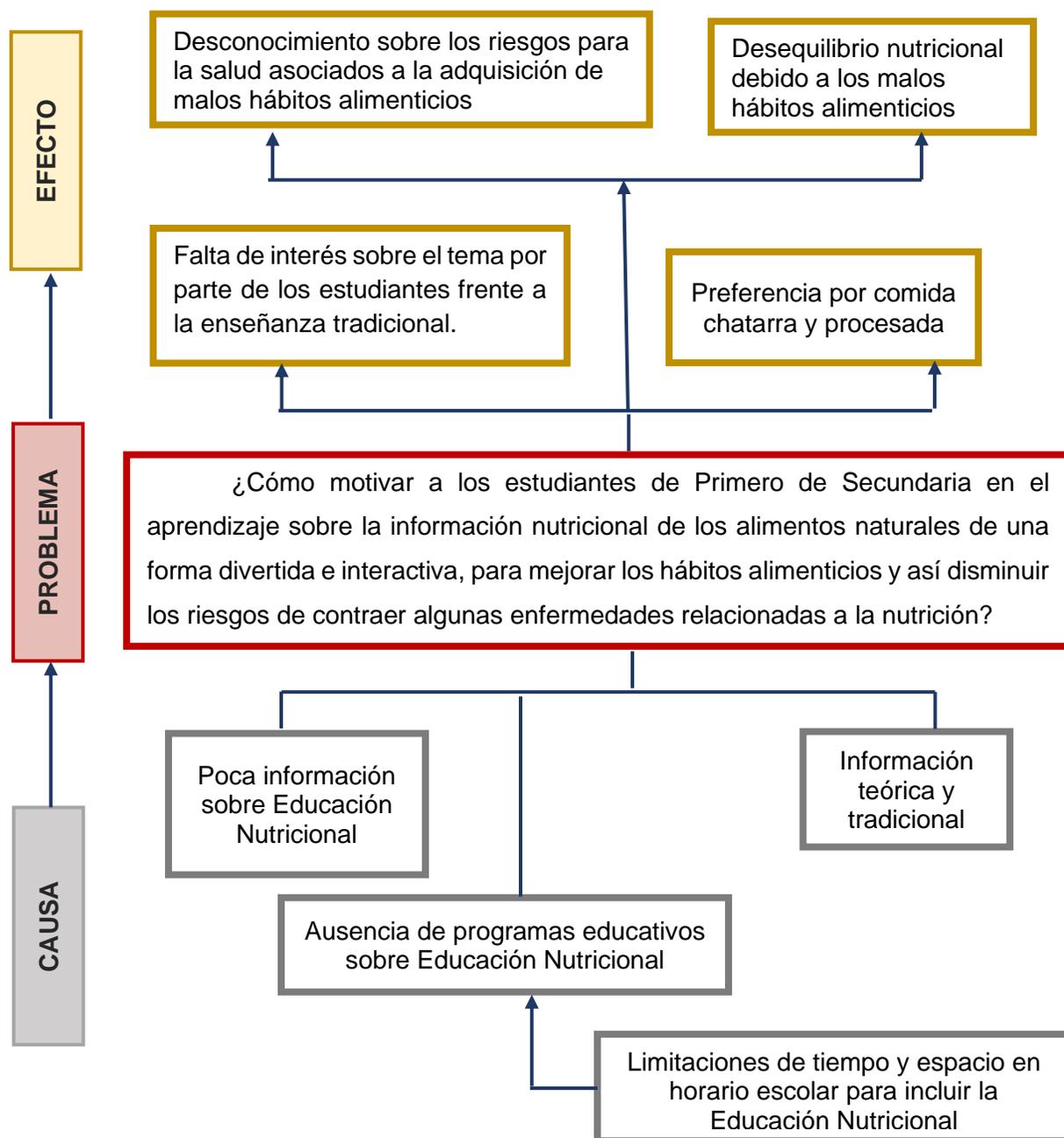
Wikipedia. (2024). Kotlin (lenguaje de programación). En *Wikipedia, la enciclopedia libre*.

[https://es.wikipedia.org/w/index.php?title=Kotlin_\(lenguaje_de_programaci%C3%B3n\)&oldid=158600774](https://es.wikipedia.org/w/index.php?title=Kotlin_(lenguaje_de_programaci%C3%B3n)&oldid=158600774)

ANEXOS

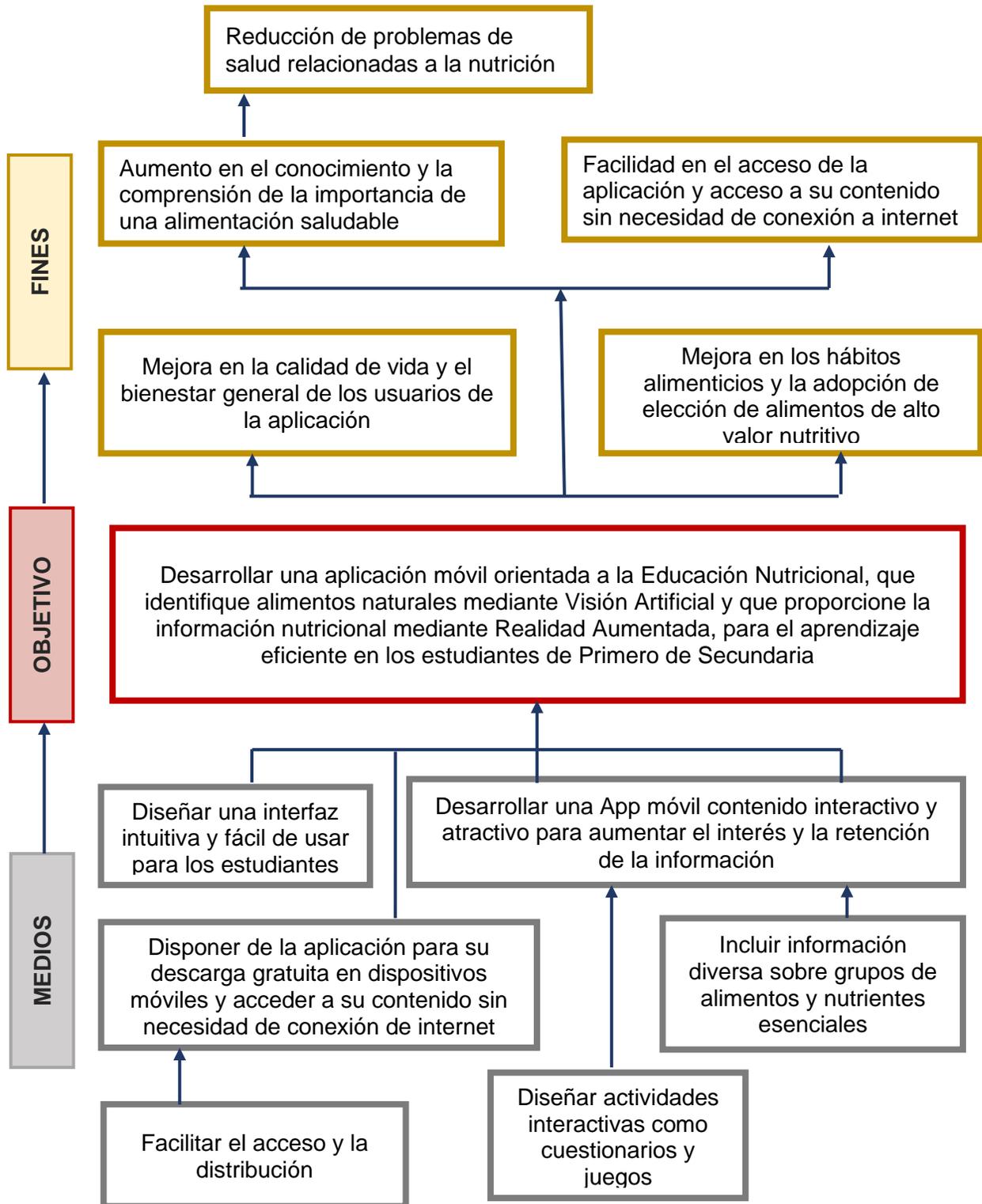
ARBOL DE PROBLEMAS

ANEXO A



ARBOL DE OBJETIVOS

Anexo B



Anexo C

CUESTIONARIO

Objetivo: Recopilar información sobre las necesidades y la percepción de los maestros en relación a la información de la educación nutricional y la importancia del consumo de alimentos naturales, así como identificar la falta de interés de los estudiantes y la prevalencia de alimentos procesados en el colegio.

Instrucciones: Por favor, responda las siguientes preguntas con la mayor sinceridad posible. Sus respuestas ayudarán a desarrollar una aplicación móvil que motive el aprendizaje sobre nutrición en los estudiantes de primero de secundaria.

Datos Generales:

Nombre del Maestro/a: _____

Materia que imparte: _____

Sección 1: Educación Nutricional

1. ¿Qué tan importante considera la educación nutricional en el desarrollo integral de los estudiantes?
 - Muy importante
 - Importante
 - Poco importante
 - No importante
2. ¿con qué frecuencia incluye temas de nutrición en su clase?
 - Frecuentemente
 - Ocasionalmente
 - Raramente
 - Nunca
3. ¿Qué recursos utiliza actualmente para enseñar sobre nutrición?
 - Libros de texto
 - Videos educativos
 - Charlas y conferencias
 - Actividades prácticas
 - Otros (por favor, especifique): _____

Sección 2: Necesidades Educativas

4. ¿Qué tipo de herramientas educativas cree que podrían mejorar el aprendizaje sobre nutrición en los estudiantes?
- Aplicaciones móviles interactivas
 - Talleres y actividades prácticas
 - Juegos educativos
 - Material audiovisual

Sección 3: Interés de los Estudiantes

5. ¿Cómo describiría el nivel de interés de los estudiantes de primero de secundaria en temas de nutrición?

1. Muy interesados
2. Interesados
3. Poco interesados
4. Nada interesados

6. ¿Qué factores cree que contribuyen a la falta de interés de los estudiantes en la nutrición?

1. Falta de información atractiva y accesible
2. Predominio de alimentos procesados en el colegio
3. Influencia de hábitos familiares

7. ¿Ha notado alguna diferencia en el interés de los estudiantes cuando se utilizan métodos educativos interactivos?

1. Sí
2. No
3. No estoy seguro/a

Sección 4: Alimentación en el Colegio

8. ¿Cómo describiría la oferta de alimentos en la cafetería del colegio?

- Principalmente alimentos naturales y saludables

- Mezcla equilibrada de alimentos naturales y procesados
- Principalmente alimentos procesados

9. ¿Cree que la oferta de alimentos en el colegio influye en los hábitos alimenticios de los estudiantes?

4. Sí, positivamente
5. Sí, negativamente
6. No estoy seguro/a

CUESTIONARIO PARA LA EVALUACION DE USABILIDAD

Anexo D

No.	PREGUNTA	SI	NO
1	¿La navegación dentro de la aplicación es intuitiva y fácil de entender?		
2	¿Encuentras fácilmente la información que buscas dentro de la aplicación?		
3	¿Las instrucciones y la información proporcionada en la aplicación son claras y fáciles de entender?		
4	¿Es fácil comprender los conceptos relacionados con la educación nutricional presentados en la aplicación?		
5	¿El diseño visual de la aplicación es atractivo y estéticamente agradable?		
6	¿La disposición de los elementos en la pantalla facilita la interacción con la aplicación?		
7	¿Las funciones interactivas de la aplicación, como los juegos y las calculadoras, son fáciles de usar?		
8	En general, ¿calificarías tu experiencia con la aplicación en términos de usabilidad como positiva?		
9	¿Recomendarías esta aplicación a otros estudiantes interesados en aprender sobre nutrición?		
10	¿Hay algún aspecto específico de la aplicación que te gustaría destacar, ya sea positivo o negativo?		

ANEXO E

Diagrama de barras sobre los resultados de las Métricas de Calidad



ANEXO F

Gráfico de barras sobre la usabilidad de la aplicación móvil

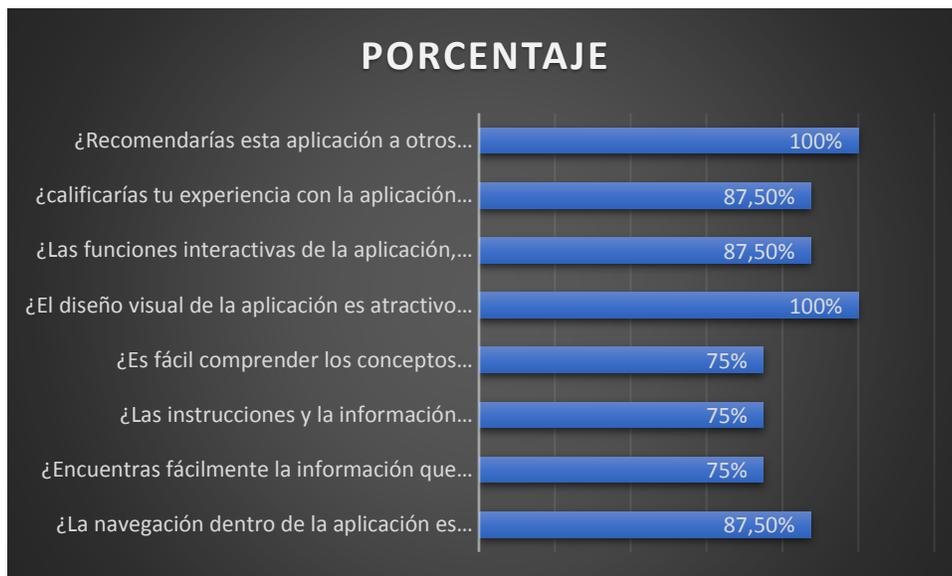
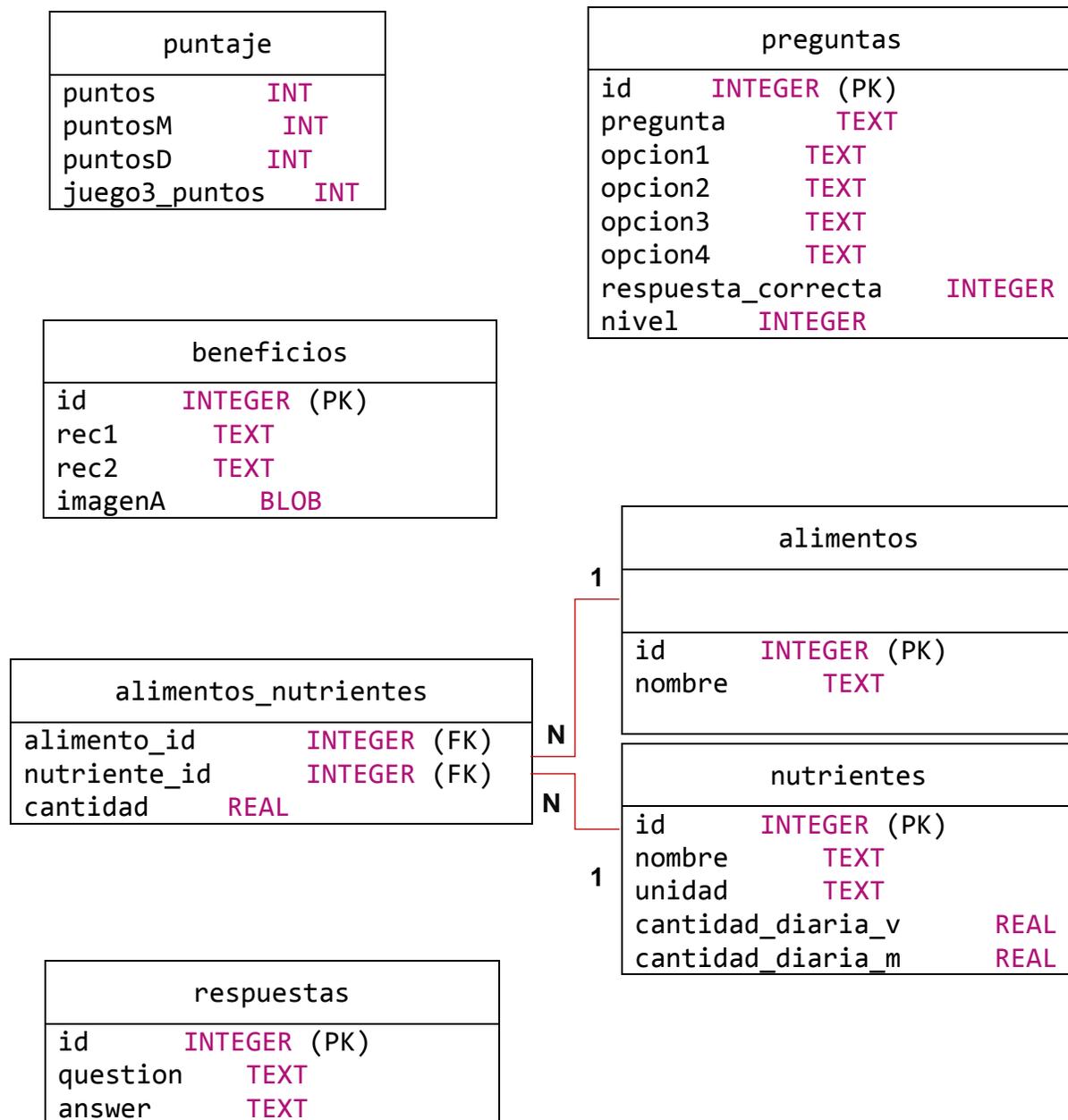


DIAGRAMA ENTIDAD RELACIÓN

ANEXO G



DICcionario DE DATOS

ANEXO H

Alimentos

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
nombre	text	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

alimentos_nutrientes

Columna	Tipo	Nulo	Predeterminado	Comentarios
alimento_id	int(11)	Sí	NULL	
nutriente_id	int(11)	Sí	NULL	
cantidad	double	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
alimento_id	BTREE	No	No	alimento_id		A	Sí	
nutriente_id	BTREE	No	No	nutriente_id		A	Sí	

beneficios

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
rec1	text	Sí	NULL	
rec2	text	Sí	NULL	
imagenA	blob	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

nutrientes

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
nombre	text	Sí	NULL	
unidad	text	Sí	NULL	
cantidad_diaria_v	double	Sí	NULL	
cantidad_diaria_m	double	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

preguntas

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
pregunta	text	Sí	NULL	
opcion1	text	Sí	NULL	
opcion2	text	Sí	NULL	
opcion3	text	Sí	NULL	
opcion4	text	Sí	NULL	
respuesta_correcta	int(11)	Sí	NULL	
nivel	int(11)	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

puntaje

Columna	Tipo	Nulo	Predeterminado	Comentarios
puntos	int(11)	Sí	NULL	
puntosM	int(11)	Sí	NULL	
puntosD	int(11)	Sí	NULL	
juego3_puntos	int(11)	Sí	NULL	

respuestas

Columna	Tipo	Nulo	Predeterminado	Comentarios
id (<i>Primaria</i>)	int(11)	No		
question	text	Sí	NULL	
answer	text	Sí	NULL	

Índices

Nombre de la clave	Tipo	Único	Empaquetado	Columna	Cardinalidad	Cotejamiento	Nulo	Comentario
PRIMARY	BTREE	Sí	No	id	0	A	No	

ANEXO I



Manual de Usuario de la Aplicación Móvil "NutriEduca"

Índice

1. Introducción
2. Instalación de la Aplicación
3. Navegación de la Aplicación
4. Funciones Principales
 - i. Escaneo de alimentos con la cámara del dispositivo
 - ii. Grupo de alimentos
 - iii. Juego 1 con tematica nutricional
 - iv. Juego 2 con temática nutricional
 - v. Prueba de conocimientos y lecciones
 - vi. Calculo del Indice de Masa Corporal

1. INTRODUCCIÓN

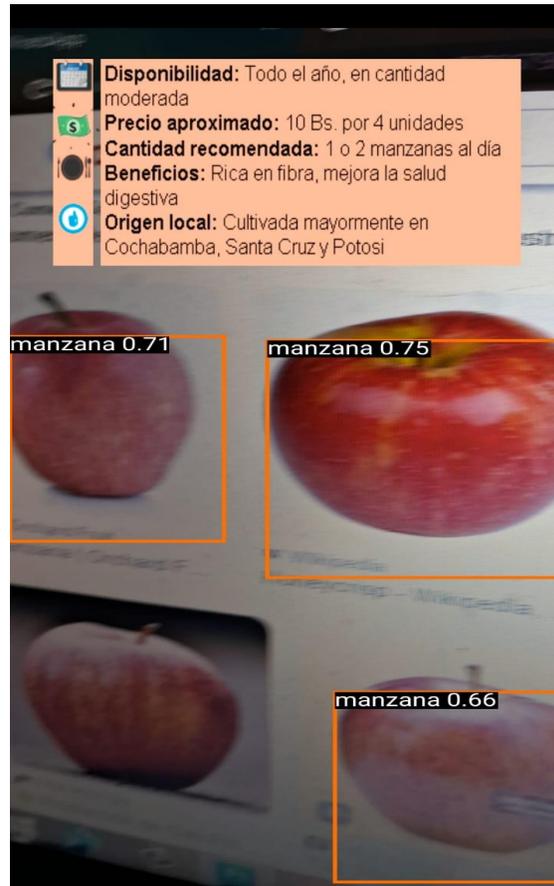
¡Bienvenidos a "NutriEduca"! Esta aplicación está diseñada para ayudar a los estudiantes de primero de secundaria a aprender sobre nutrición de manera divertida e interactiva. Aquí encontrarás lecciones, información y algunos juegos que te ayudaran a aprender más sobre los alimentos naturales y nutrición.

2. INSTALACIÓN DE LA APLICACIÓN

- Abre la Play Store en tu dispositivo.
- Busca "NutriEduca".
- Haz clic en "Instalar".

3. FUNCIONES PRINCIPALES

i. Escaneo de Alimentos con la Cámara del Dispositivo



- **Funcionalidad:**

La función de escaneo de alimentos utiliza técnicas de visión artificial y realidad aumentada para reconocer alimentos a través de la cámara del dispositivo móvil. Una vez reconocido un alimento, la aplicación muestra información breve sobre el mismo.

- **Instrucciones de Uso:**

- 1) Abre la función de escaneo desde el menú principal.
- 2) Apunta la cámara hacia el alimento que deseas escanear.
- 3) Trata de mantener el dispositivo lo mas quieto posible.
- 4) La aplicación utilizará visión artificial para reconocer el alimento.
- 5) Una vez reconocido, se mostrará información básica sobre el alimento reconocido.

ii. Grupo de Alimentos



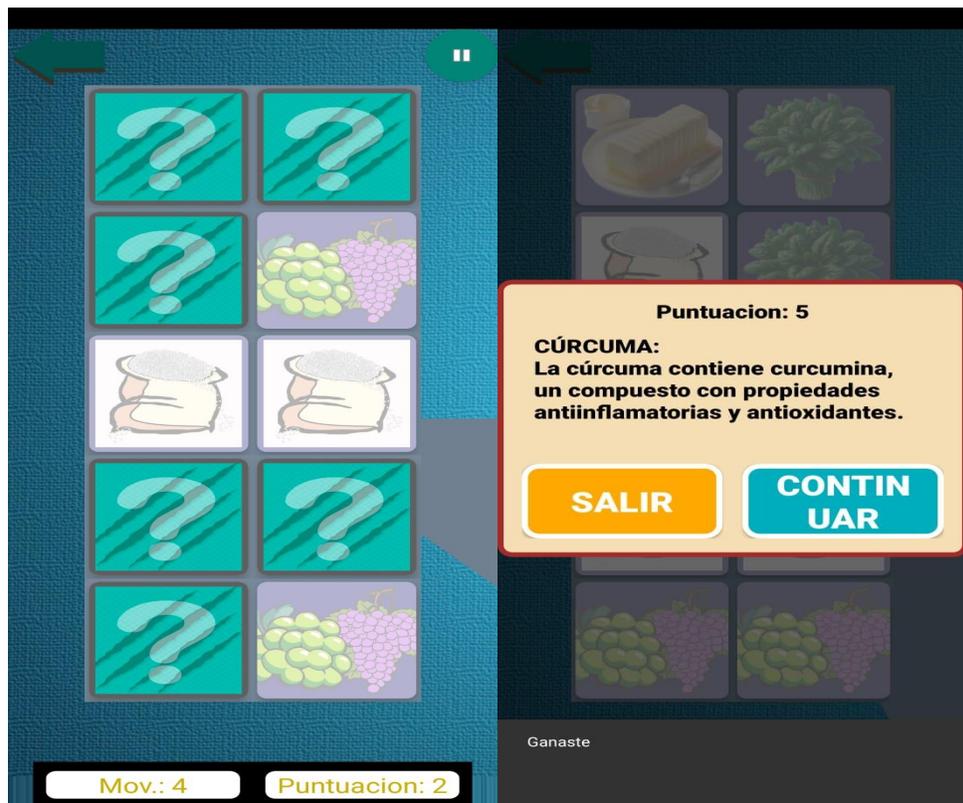
- **Funcionalidad:**

En esta sección, se presentan los 6 grupos de alimentos básicos. Al seleccionar un grupo específico, se abrirá una ventana con los alimentos que conforman ese grupo. Al seleccionar un alimento en particular, se visualiza información detallada y la tabla nutricional del mismo, además de un audio explicativo.

- **Instrucciones de Uso:**

- 1) Desde el menú principal, selecciona "Grupo de Alimentos".
- 2) Selecciona uno de los 6 grupos alimenticios disponibles (ej. frutas, cereales, lácteos, etc.).
- 3) Se mostrará una lista de alimentos dentro del grupo seleccionado.
- 4) Selecciona un alimento específico para ver detalles como la información nutricional y escuchar una explicación en audio sobre sus beneficios.

iii. JUEGO 1 CON TEMÁTICA NUTRICIONAL (JUEGO DE MEMORIA)



- **Funcionalidad:**

Este juego consiste en un juego de memoria por niveles, donde el usuario debe emparejar dos imágenes iguales relacionadas con alimentos. Al completar cada juego, se muestra información o datos relevantes sobre algún alimento relacionado.

- **Instrucciones de Uso:**

- 1) Accede al juego desde el menú principal seleccionando "Juego de Memoria".
- 2) Selecciona el nivel de dificultad deseado.
- 3) Encuentra las imágenes iguales de alimentos en el menor número de intentos.
- 4) Al completar el juego, se desbloquea información nutricional o datos interesantes sobre los alimentos emparejados.

v. PRUEBA DE CONOCIMIENTOS Y LECCIONES



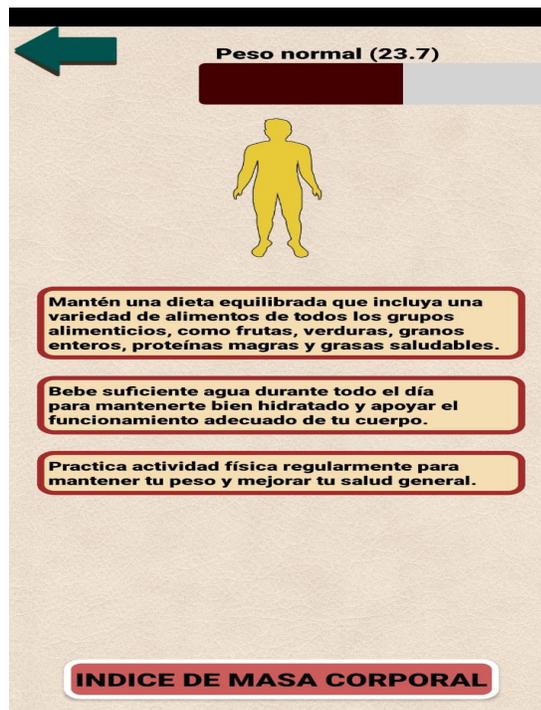
- **Funcionalidad:**

Esta sección ofrece niveles de pruebas de conocimiento sobre nutrición. Cada nivel incluye preguntas relacionadas con la nutrición y la salud. Además, se proporcionan lecciones de aprendizaje asociadas. Si el usuario responde incorrectamente, pierde una vida y se muestra la respuesta correcta.

- Instrucciones de Uso:

- 1) Selecciona "Prueba de Conocimientos" desde el menú principal.
- 2) Elige el nivel de dificultad o tema que deseas explorar.
- 3) Responde las preguntas seleccionando la respuesta correcta.
- 4) Si respondes incorrectamente, se mostrará la respuesta correcta y podrás aprender de tus errores.

vi. CÁLCULO DEL ÍNDICE DE MASA CORPORAL (IMC)



- **Funcionalidad:**

Esta función permite calcular el Índice de Masa Corporal (IMC) del usuario. Además, ofrece recomendaciones personalizadas según el resultado del IMC obtenido, promoviendo hábitos de alimentación saludables.

- **Instrucciones de Uso:**

- 1) Accede a la función de cálculo del IMC desde el menú principal.
- 2) Ingresa tu peso y altura en las casillas proporcionadas.
- 3) La aplicación calculará automáticamente tu IMC.
- 4) Basado en tu IMC, recibirás recomendaciones personalizadas para mejorar tus hábitos alimenticios y de salud.