
FITBITE: aplicación móvil para la recomendación de comidas saludables con funcionalidad de clasificación de imágenes

FITBITE: mobile application for the recommendation of healthy meals with image classification

Juan Martín Domínguez Matos

<https://orcid.org/0009-0006-9948-8617>
juanmartin.dominguez@unmsm.edu.pe

Jeanpiere Julian Palacios Barrutia

<https://orcid.org/0009-0008-8708-007X>
jeanpiere.palacios@unmsm.edu.pe

Flavia Francesa Abanto Salas

<https://orcid.org/0009-0006-6688-009X>
flavia.abanto@unmsm.edu.pe

Ivan Carlo Petrlik Azabache

<https://orcid.org/0000-0002-1201-2143>
ipetriika@unmsm.edu.pe

Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática
Lima, Perú

RECIBIDO: 25/10/2023 - ACEPTADO: 14/11/2023 - PUBLICADO: 30/12/2023

RESUMEN

La investigación que presentamos se enfoca en el desarrollo de una aplicación móvil diseñada para la clasificación de imágenes, utilizando un algoritmo supervisado. Esta funcionalidad es esencial para ofrecer recomendaciones de comida saludable, contribuyendo al mantenimiento de estándares de salud óptimos en los usuarios. Para el desarrollo de esta aplicación, adoptamos el framework Scrum, estructurándolo en cuatro sprints principales durante un período estimado de ocho semanas. En cuanto a su diseño metodológico, esta investigación es de tipo no experimental-descriptivo. Evaluamos la variable independiente a través de diversas dimensiones e indicadores, que se reflejan en las preguntas de un cuestionario aplicado a una muestra seleccionada de 17 individuos. Según los resultados obtenidos, el modelo de clasificación de imágenes implementado en la aplicación posee una precisión (exactitud) del 86,19%. Además, basándonos en el cuestionario System Usability Scale, la usabilidad de la aplicación se valoró en 81.029 de 100 puntos, lo que nos lleva a concluir que el aplicativo móvil alcanza un nivel de satisfacción de tipo A. Es importante mencionar que, una vez que se realiza el lanzamiento oficial de la aplicación, se evaluará su rendimiento utilizando la herramienta JMeter.

Palabras clave: Android, Flutter, aplicación móvil, comidas saludables, aprendizaje automático, clasificación de imágenes, salud.

ABSTRACT

The research we present focuses on the development of a mobile application designed for image classification, using a supervised algorithm. This functionality is essential to provide healthy food recommendations, contributing to the maintenance of optimal health standards in users. For the development of this application, we adopted the Scrum framework, structuring it in four main sprints during an estimated period of eight weeks. Regarding its methodological design, this research is non-experimental-descriptive. We evaluated the independent variable through various dimensions and indicators, which are reflected in the questions of a questionnaire applied to a selected sample of 17 individuals. According to the results obtained, the image classification model implemented in the application has a precision (accuracy) of 86.19%. Furthermore, based on the System Usability Scale questionnaire, the usability

of the application was rated at 81.029 out of 100 points, which leads us to conclude that the mobile application achieves a level of satisfaction of type A. It is important to mention that, once the application is officially launched, its performance will be evaluated using the JMeter tool.

Keywords: Android, Flutter, mobile app, healthy meals, machine learning, image classification, health.

I. INTRODUCCIÓN

Actualmente, en el mundo post-pandémico, la salud y una adecuada nutrición ha tomado gran relevancia. La alimentación saludable es vital para prevenir enfermedades no transmisibles (ECNT) y mejorar el bienestar general de las personas, más aún queda camino por recorrer. Según el informe titulado “Estado de la Seguridad Alimentaria y la Nutrición en el Mundo” (SOFI 2019), en años más recientes, hubo un alza constante en la cantidad de personas con desnutrición y, al mismo tiempo, se registra un número alarmante de personas obesas y con exceso de peso (FAO & OMS, 2020).

Esta situación puede conocerse a partir del Índice de Alimentación Saludable Alternativo (AHEI, por sus siglas en inglés). AHEI incluye una serie de elementos respecto a la ingesta de alimentos y hábitos vinculados a la prevención de ECNT (Méndez, I., 2021). AHEI se usa para evaluar la calidad de la alimentación, derivar el consejo de nutrición personalizado y cuantificar los cambios en la ingesta dietética (Zenun, F. et al., 2022). En 2018, la media global de la puntuación AHEI fue de 40,3 (Miller, V. et. al, 2022), para 11 componentes: 6 requieren mayor ingesta, 1, consumo moderado y 4, limitarse (Al-Ibrahim, A. & Jackson, R., 2019). Para estos, la puntuación de 10 indica que se cumplieron por completo las recomendaciones; una puntuación de 0 representa el comportamiento alimentario menos saludable (Akbaraly, T. et al., 2018).

Mediante la siguiente encuesta, se conocieron los principales hábitos alimenticios en el mundo:

- Prefiero comer bien que ser delgado (62%),
- Creo que la mayoría de los planes de dieta, finalmente, fallan (61%),
- Estoy contento con mi peso actual (55%),
- He intentado una dieta para perder peso (51%),
- Hago ejercicio para no tener que cuidar lo que como (45%) (Ipsos, 2019).

En países latinoamericanos y caribeños, el panorama es similar al global. En el año 2016, el 24.2% de la población adulta era obesa, muy por encima

del promedio mundial de 13,1% (FAO, IFAD, PAHO, UNICEF & WFP, 2023). Respecto al indicador AHEI, la media regional va desde 30.3, mientras que los puntajes más altos fueron para mayor cantidad de legumbres y nueces y menor contenido de sodio. (V. Miller et. al, 2022). Además, según la Organización Panamericana de la Salud (OPS), el consumo de productos ultraprocesados en Latinoamérica, entre 2008 y 2014, aumentó 8,3%, y se predijo un incremento, entre 2015 y 2019, del 7,8%, siendo este tipo de productos altamente relacionados con una mayor incidencia de sobrepeso y obesidad.

El Perú presenta estadísticas alarmantes sobre la alimentación y estado nutricional poblacional. Del Estudio de Enfermedades No Transmisibles y Transmisibles (ENDES), se sabe que el sobrepeso afecta al 37.9% de personas mayores de 15 años, mientras que la obesidad afecta al 24,6% de dicho grupo etario. Los mayores porcentajes de sobrepeso, por departamento, son Tumbes (40.0%), Lima Metropolitana (40.2%) y Madre de Dios (43.9%). En Lima Metropolitana, el porcentaje de obesidad es poco mayor al 20%. Además, a nivel nacional, 4.5 días a la semana son aquellos en el que, mayores de 15 años, consumen frutas, sean troceadas o enteras y, dicho grupo etario, consume ensalada de verduras 3.4 días en ese mismo periodo de tiempo (Instituto Nacional de Estadística e Informática, 2021), en un contexto en el que, el 19.0% de la población no cuenta con los recursos económicos suficientes para alimentarse saludablemente (3.4 USD), a diferencia de una dieta mínima en calorías (0.7 USD) (FAO, FIDA, OPS, WFP & UNICEF, 2020). En Villa El Salvador, distrito de la capital peruana, de 329 hogares entrevistados al azar, el 22.8% cuenta con alimento seguro: la manera formal más frecuente de obtención de recursos alimentarios en los últimos 6 meses y 4 semanas fue el programa Qali Warma (42.0% para ambos); informalmente, un 20.0% consume comida preparada con alguien fuera del hogar (Brewer et. al, 2021).

En una encuesta sobre los hábitos saludables en el Perú, se obtuvieron estos hallazgos:

- El 72% de los encuestados señala que su salud personal es “muy saludable/ saludable”.

- El 69% de los encuestados afirman comer saludable “muy seguido/seguído”; el 27%, “a veces” y el 4% restante, “pocas veces/nunca”.
- Según el nivel de ingresos, el porcentaje menor de los encuestados que declaran consumir comida saludable de forma “muy seguido/seguído” corresponde a los de bajos ingresos (68%) (Win & Datum Internacional, 2021).

También, existen desafíos en la cultura alimentaria a causa del consumo de comidas poco saludables, una educación insuficiente sobre alimentación adecuada en los colegios y a los hábitos alimenticios inadecuados en los hogares peruanos (Rabanal, J., 2019).

En vista del análisis realizado, en la Figura 1, se presenta el Árbol del problema de forma general.

En las raíces del árbol están las causas del problema: “la inseguridad alimentaria en el Perú”, “cultura alimentaria de la población” y “consumo no frecuente de frutas y verduras”. Entre las consecuencias, se encuentran: “problemas de nutrición en la población”, “mayor riesgo de desarrollar ECNT relacionadas con la alimentación” y “posibilidad de mayores gastos en atenciones médicas”.

En estos últimos años, se ha evidenciado un gran avance respecto a la Inteligencia Artificial (IA), siendo adoptada en distintos sectores de la sociedad: el poder de las máquinas para el aprendizaje y la

toma de decisiones autónomas ha permitido una interacción mucho más significativa con la tecnología y en la manera de la ejecución de diversas tareas y procesos en nuestra vida cotidiana: desde la medicina y la atención sanitaria hasta el comercio electrónico, la industria manufacturera, el transporte y la logística, entre otros. Los sistemas de IA están ayudando a los médicos a diagnosticar enfermedades, a los científicos a analizar grandes volúmenes de datos y a las empresas a mejorar la eficiencia de sus operaciones. Además, los asistentes virtuales y los chatbots basados en IA se han vuelto comunes en el servicio al cliente y en la interacción con los usuarios en línea.

En el mundo actual, donde la salud y el bienestar se han vuelto prioridades, la adopción de una alimentación saludable es cada vez más relevante. Sin embargo, encontrar opciones de comidas nutritivas y equilibradas puede resultar complicado en medio de la amplia variedad de alimentos disponibles. Para abordar esta necesidad, ha surgido una aplicación móvil innovadora que, a partir de la Inteligencia Artificial, recomienda comidas saludables y facilita la toma de decisiones informadas en torno a la alimentación del usuario: "FITBITE".

La aplicación "FITBITE" se basa en algoritmos de *Machine Learning* y clasificación de imágenes para ofrecer a los usuarios recomendaciones de comidas saludables para las comidas del día. Al tomar una foto de sus insumos alimenticios, mediante los

Figura 1.
Árbol del problema identificado



algoritmos, se analiza y clasifica automáticamente al insumo y recomienda las comidas en base a este, proporcionando los ingredientes e información nutricional.

II. ANTECEDENTES

La influencia de la tecnología móvil en nuestras vidas ha abierto nuevas oportunidades para abordar el desafío de facilitar la toma de decisiones saludables. En este contexto, "FITBITE" ayudará a las personas en el descubrimiento y disfrute de comidas saludables de manera sencilla y conveniente. A partir de algoritmos inteligentes y una vasta base de datos de alimentos, esta aplicación proporciona recomendaciones saludables, información nutricional detallada y opciones de recetas saludables. En esta sección, exploraremos los antecedentes y la relevancia de esta aplicación móvil de recomendación de comidas saludables, destacando cómo puede marcar la diferencia en la vida de las personas y contribuir a una sociedad más saludable.

Ann et al. (2021) crearon la aplicación móvil Feast In, destacable por la personalización de recetas y el uso de Inteligencia Artificial para la obtención de recetas en base a una imagen del usuario, usando la API de TensorFlow. Asimismo, Cornejo & Urbano (2021) implementaron una app móvil con *Visual Recognition* para el monitoreo de una dieta saludable, ofreciendo una guía y recetas para mejorar sus hábitos alimenticios. El 94.2% y 89.8% de los encuestados sugieren y están satisfechas con la app, respectivamente. El 89.9% afirma que es precisa la funcionalidad de reconocimiento de comidas.

Castellanos y Niño (2021) realizaron un trabajo de tesis sobre una aplicación móvil Android que sugiere recetas culinarias basadas en búsquedas previas usando Inteligencia Artificial. El 93.8% de los encuestados señaló que usaría la aplicación para que les ayude a encontrar recetas. Por su lado, Campos y Rioseco (2020) desarrollaron un algoritmo de recomendación para que universitarios puedan llevar un estilo saludable de vida, sugiriendo recetas altamente nutricionales según sus gustos y preferencias. Por otro lado, Mejía e Insuasty (2022) desarrollaron la aplicación móvil HTMarket. Esta incorporó el modelo de *Machine Learning* Random Forest para las recomendaciones sobre productos de acuerdo a su nutrición, dieta y problemas de salud, en base a su historial de búsqueda.

Kul & Sayar (2022) diseñaron un Sistema Inteligente de Sugerencia de Recetas en Android. Su uso está orientado a las cocinas, permitiendo al usuario preparar recetas con los materiales disponibles.

Con TensorFlow Lite, una imagen del ingrediente se clasifica para recomendar las recetas, consultando a MongoDB para mostrar las recetas. Se calculó al 100% la búsqueda correcta del material buscado por el usuario, para los 78 ingredientes alimentarios conocidos. Además, Jiang, Wang, Liu, Nie, Duan & Xu (2019) presentaron el software móvil Market2Dish, la cual ayuda a las personas a desarrollar hábitos alimenticios saludables y a personalizar las recetas en base a su perfil. A partir de micro-videos, se identifican ingredientes gracias al modelo de clasificación de imágenes Inception-v3 Net, obteniendo un accuracy del 95.5%. Kong et al. (2023) desarrollaron la aplicación híbrida MyDietCam, la cual incluye una función de reconocimiento de alimentos para ofrecer información nutricional. Entre las comidas recomendadas, están las frutas, verduras, carne, granos y cereales, etc. La aplicación fue calificada con 5.13/7 en el test de usabilidad.

III. MATERIALES Y MÉTODOS

A. Metodología

Scrum es un enfoque ágil que, mediante soluciones adaptativas, promueve la generación de valor para problemas complejos, beneficiando a individuos, equipos y organizaciones (Schwaber, K. & Sutherland, J., 2020).

A.1. Planificación

El proyecto se ha planificado en 4 *sprints*. Estos eventos tienen una duración limitada de un mes o menos con el fin de crear consistencia (Schwaber, K. & Sutherland, J., 2020). En la Tabla 1, se explica la planificación de los *sprints*.

Tabla 1
Planificación de los sprints del proyecto.

Nº Sprint	Actividades principales	Duración
1	Documentación inicial Prototipado de las interfaces en Figma Diseño de la base de datos	21 días
2	Desarrollo de las interfaces con el framework Flutter Consulta de datasets para crear el modelo de clasificación	21 días
3	Registro de comidas saludables en la base de datos Visualización y obtención de recomendaciones de comidas saludables Información nutricional de las comidas saludables	21 días
4	Opción "Favoritos" para las comidas saludables Ejecución de pruebas	21 días

A.2. Historias de usuario

La Tabla 2 ilustra las historias de usuario de mayor importancia.

Tabla 2
Historias de usuario del proyecto.

ID	Característica	Razón
HU-001	Como usuario quiero registrar mis datos en la aplicación	Acceder a las funciones de la app
HU-002	Como usuario quiero obtener información nutricional sobre las comidas saludables	Planificar mi alimentación
HU-003	Como usuario quiero obtener recomendaciones de comidas saludables	Incorporar nuevas comidas a mi alimentación
HU-004	Como usuario quiero añadir comidas saludables como Favoritas	Consultarlas de forma rápida

B. Esquema conceptual

En la Figura 2, se esquematiza los pasos para la creación del modelo de clasificación de imágenes y su relación con la app móvil en uso.

La sección verde inicia con la recolección de imágenes para generar el modelo de clasificación de imágenes. 31 clases fueron consideradas, a partir

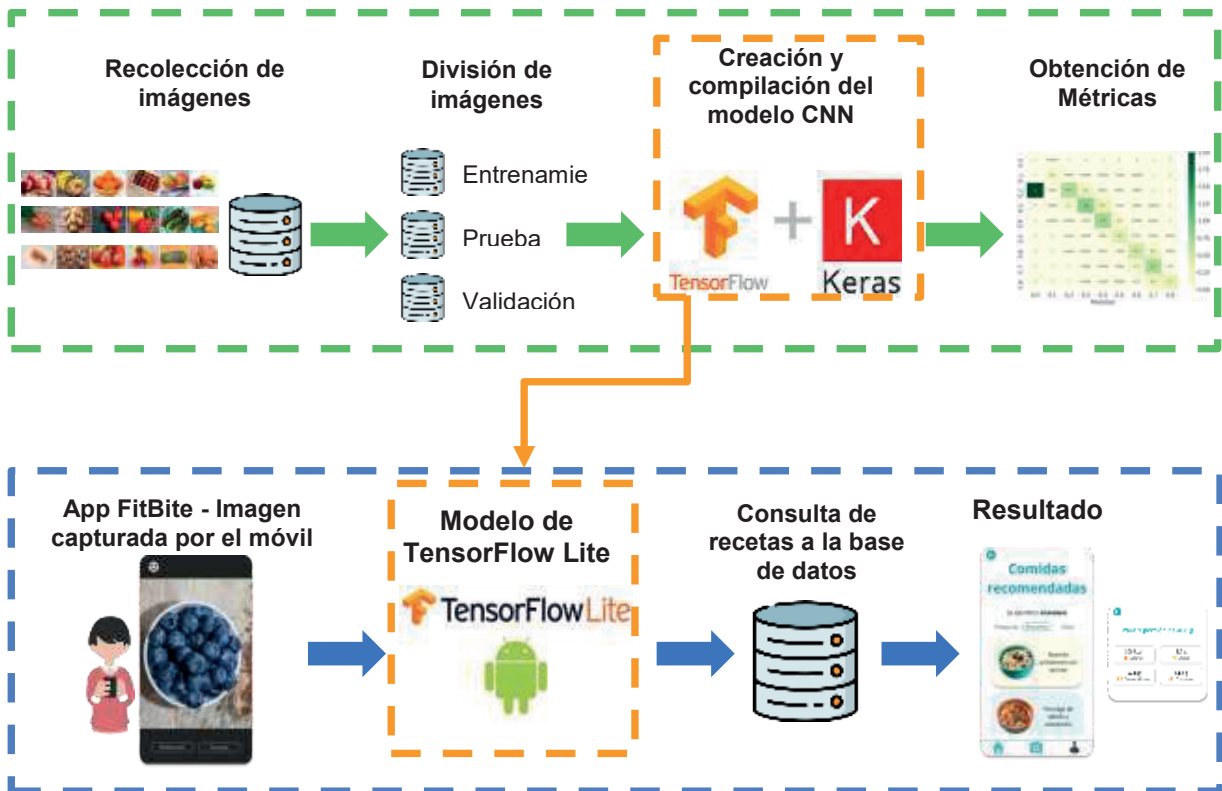
de datos del MIDAGRI (2022) sobre las frutas y verduras más producidas en Perú. En la Tabla 3 se nombran los datasets consultados.

Tabla 3
Datasets usados para el modelo de clasificación.

Autor	Nombre del dataset	Estructura
Oltean (2021)	Fruits 360	Dataset de prueba, entrenamiento y validación
Zhang (2023)	Southeast Asia Common Fruit and Vegetables	Dataset de entrenamiento y validación
Seth (2022)	Fruits and Vegetables Image Recognition Dataset	Dataset de prueba, entrenamiento y validación
Ahmed et al. (2022)	Vegetable Image Dataset	Dataset de prueba, entrenamiento y validación
Images.cv	Datasets Varios	Dataset de prueba, entrenamiento y validación
Minut (2021)	Fruits-262	No estructurado

Con las imágenes divididas en conjuntos de prueba, entrenamiento y validación, se crea el modelo de redes neuronales convolucionales (CNN) y Keras. Por CNN, se interpreta como un tipo de arquitectura

Figura 2
Esquema de la clasificación de imágenes y su uso en la app



de red neuronal de alimentación directa, capaz de identificar particularidades de los datos (Z. Li et al., 2022). Keras es un marco de alto nivel que ofrece una interfaz sencilla e intuitiva con el fin de construir y entrenar modelos de *Deep Learning* (Rao, 2023). Después, se obtienen las métricas y se guarda el modelo de TensorFlow Lite (.tflite). Esta es una versión ligera de TensorFlow que ayuda a implementar modelos de *Machine Learning* en dispositivos móviles (Singh & Bhadani, 2020).

La sección azul se relaciona con el manejo de la aplicación FitBite por el usuario. Este accede a la funcionalidad de clasificación de imágenes y, la imagen capturada es la entrada al modelo. Vea que la relación entre ambas secciones es mediante este modelo, la sección naranja. Luego, la imagen recibe una etiqueta, la cual se consulta a la base de datos. Las recomendaciones de recetas saludables serán mostradas con su información nutricional.

C. Clasificación de Imágenes

La clasificación de imágenes es una tarea fundamental en el campo de la visión por computadora, que busca asignar etiquetas o categorías a las imágenes de manera automática.

Esta tarea ha experimentado un gran avance en los últimos años gracias al desarrollo de técnicas de aprendizaje profundo, particularmente las redes neuronales convolucionales.

La clasificación de imágenes ha experimentado un gran avance en los últimos años gracias al desarrollo de técnicas basadas en redes neuronales. Las redes neuronales convolucionales (CNN) se han convertido en un enfoque fundamental en esta tarea, ya que permiten extraer características relevantes de las imágenes y realizar una clasificación precisa. Según el estudio de He et al. (2016), la red neuronal convolucional conocida como ResNet ha logrado resultados destacados en competencias de clasificación de imágenes, superando incluso a la capacidad humana en ciertos casos. Además, el trabajo de Szegedy et al. (2017) ha demostrado la efectividad de las redes neuronales con arquitecturas más profundas, como InceptionV3, para lograr un mejor rendimiento en la clasificación de imágenes. Estos avances en el uso de redes neuronales han impulsado la aplicación de técnicas de clasificación de imágenes en campos como la visión por computadora, la medicina y la seguridad. En resumen, las redes neuronales convolucionales han revolucionado la clasificación de imágenes en los últimos años, ofreciendo resultados sobresalientes

y abriendo nuevas posibilidades en diversas áreas de aplicación.

Según el estudio de Deng et al. (2019), las CNN están diseñadas específicamente para trabajar con datos de imágenes y han demostrado un rendimiento sobresaliente en la clasificación de imágenes. Estas arquitecturas aprovechan las operaciones de convolución y pooling para extraer características relevantes de las imágenes, permitiendo una representación jerárquica de los datos visuales. Las capas convolucionales detectan patrones locales, mientras que las capas de pooling reducen la dimensionalidad y preservan las características más importantes. Las redes neuronales convolucionales también pueden contener capas completamente conectadas para realizar la clasificación final. Su arquitectura profunda permite aprender características cada vez más abstractas a medida que la información fluye a través de la red.

La arquitectura de redes neuronales convolucionales (CNN) se ilustra en la Figura 3. Como entrada, está la imagen capturada por el móvil del usuario y como salida, la etiqueta que recibirá la imagen. Dicha imagen es de 3 canales y se prueba en el modelo. tflite con dimensiones 224 x 224. El modelo se crea con las siguientes capas: convolucionales, maxPool, Flatten y Dense.

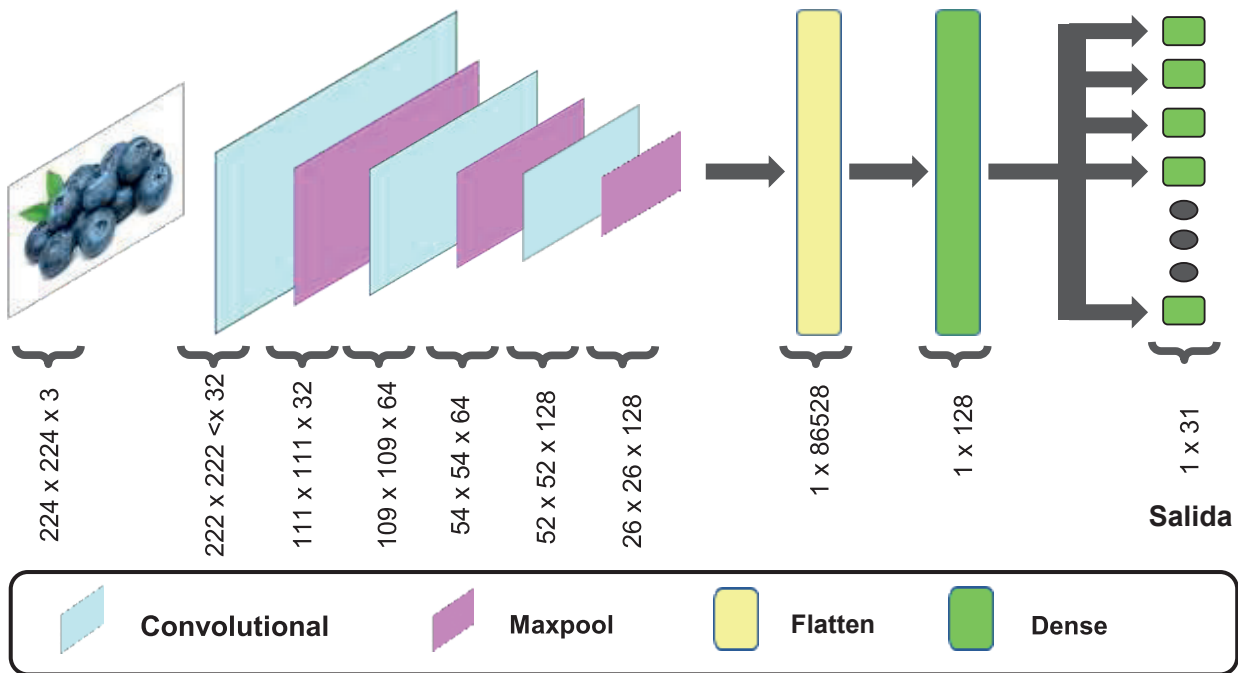
D. Funcionalidad de Clasificación de Imágenes

Esta funcionalidad se refiere a la capacidad de un sistema o algoritmo para asignar etiquetas o categorías, automáticamente, a las imágenes en función de su contenido visual. Esta tarea es esencial en aplicaciones como la organización de grandes conjuntos de imágenes, la búsqueda visual y el etiquetado automático de imágenes en plataformas en línea.

La clasificación de imágenes mediante el uso de redes neuronales ha demostrado ser una técnica efectiva y versátil en el campo de la visión por computadora. Según el estudio de Liu et al. (2020), las redes neuronales convolucionales (CNN) han alcanzado un alto rendimiento en la clasificación precisa de imágenes en diversas aplicaciones. Estas redes pueden aprender automáticamente características relevantes de las imágenes y realizar una asignación adecuada a diferentes categorías. Además, las técnicas de aprendizaje profundo aplicadas a las redes neuronales, como la arquitectura de redes residuales, han mejorado significativamente la capacidad de generalización y robustez de los modelos de clasificación de imágenes (Huang et al., 2020). Esta funcionalidad

Figura 3

Arquitectura de redes neuronales convolucionales (CNN)



ha impulsado el desarrollo de sistemas de clasificación de imágenes en campos como la medicina, la seguridad y la industria del entretenimiento. En resumen, la combinación de la clasificación de imágenes y las redes neuronales ha permitido avances significativos en la identificación y categorización automática de imágenes en diversas aplicaciones.

Proceso de IA en la Clasificación de Imágenes

Según el estudio de Chen et al. (2022), el proceso de IA para la clasificación de imágenes generalmente consta de tres partes principales que pueden ser divididas en subtarefas para un mejor entendimiento del flujo: preprocesamiento de datos, entrenamiento del modelo y evaluación del rendimiento. En la etapa de preprocesamiento, se realizan tareas como la normalización, redimensionamiento y eliminación de ruido para preparar los datos de entrada. Luego, en la etapa de entrenamiento, se utiliza un conjunto de datos etiquetados para entrenar el modelo de IA, generalmente mediante el uso de redes neuronales convolucionales, finalmente, en la etapa de evaluación, se analiza el rendimiento del modelo utilizando métricas.

1. Preparación de datos: En esta etapa, se recopila y prepara los datos, en este caso, las imágenes que conformarán el modelo. Su selección debe conformar un conjunto diverso y

representativo, con la etiqueta correspondiente a su clase para su posterior división en conjuntos de entrenamiento, validación y prueba.

Según el estudio de Hossain et al. (2020), la preparación de datos en la clasificación de imágenes puede involucrar técnicas como la normalización de píxeles, el redimensionamiento de imágenes y la eliminación de ruido. Estas operaciones son importantes para mejorar la calidad de los datos.

2. Extracción de características: Esto se logra mediante el uso de las redes neuronales, precisamente, gracias a las capas que conforman dicha red. Se aplican filtros para identificar patrones visuales en distintas escalas y niveles de abstracción.

Según el estudio de Li et al. (2020), existen diferentes enfoques para la extracción de características en la clasificación de imágenes, siendo una de ellas las redes neuronales convolucionales (CNN), las cuales permiten resaltar patrones y características discriminativas en las imágenes, mejorando el rendimiento del modelo de clasificación.

3. Aprendizaje y entrenamiento: Se usa el conjunto de datos de entrenamiento y las características previamente extraídas para el entrenamiento del

modelo. En esta etapa, se pueden utilizar algoritmos de optimización, como la retropropagación del error.

Según el estudio de Chen et al. (2021), se utiliza un conjunto de datos etiquetados para entrenar el modelo utilizando algoritmos de Deep Learning, como las redes neuronales convolucionales (CNN). El modelo se ajusta a los datos de entrenamiento, mejorando sus parámetros para minimizar la función de pérdida y su capacidad de clasificación. A medida que el modelo se entrena, adquiere la capacidad de reconocer patrones y generalizar a nuevas imágenes, lo que permite realizar clasificaciones precisas.

- 4. Validación y ajuste de hiperparámetros:** Mediante el conjunto de validación, se evalúa el desempeño del modelo y se ajustan los hiperparámetros.

Según el estudio de Belilovsky et al. (2020), durante esta etapa, se utiliza validación cruzada y técnicas como la búsqueda en cuadrícula (grid search) o la optimización bayesiana para encontrar la combinación óptima de hiperparámetros que maximice la precisión y la capacidad de generalización del modelo. Esta etapa de validación y ajuste es esencial para obtener un modelo de clasificación de imágenes con un rendimiento óptimo.

- 5. Evaluación y prueba:** Con el conjunto de prueba, se evalúa el rendimiento del modelo entrenado. Es aquí donde se obtendrán las predicciones para las imágenes de prueba, en base a comparaciones con las etiquetas reales, obteniéndose así distintas métricas para evaluar el modelo.

Según Loshchilov y Hutter (2020), durante esta etapa, se utiliza un conjunto de datos de prueba separado para evaluar y comparar el desempeño de diferentes combinaciones de hiperparámetros. La etapa de prueba de hiperparámetros permite seleccionar los valores óptimos para lograr un modelo de clasificación de imágenes con un rendimiento óptimo en conjuntos de datos no vistos previamente.

Una vez explicado esto, se explicará el código referente a la clasificación de imágenes. Este primer código corresponde a la pantalla “Recomendacion-Saludable”. Es aquí donde el usuario presiona el botón “Tomar Foto” para usar la cámara. En la Figura 4, se aprecia que se crea un objeto picker de la librería ImagePicker para obtener la imagen que capture el usuario (ImageSource: camera)

La Figura 5 muestra el código para otorgar permisos de uso de la cámara. Nótese que la imagen se pasa como parámetro a la pantalla “ResultadoClasificacion”.

En la pantalla “ResultadoClasificacion” se carga el modelo .tflite y el archivo .txt donde se guardan las etiquetas de las frutas/verduras consideradas en el modelo. El código referente a la carga de ambos se aprecia en la Figura 6.

En la Figura 7, se muestra el método “obtenerEtiqueta”, que contiene al método de TensorFlow Lite “runModelOnImage”. Este ejecuta el modelo. tflite previamente entrenado. Seguidamente, se establece que se quiere obtener solo 1 resultado, es decir, solo 1 etiqueta, y que tenga una confianza en la clasificación mayor igual al 50% para que aparezca como resultado.

Figura 4

Código para la obtención de la imagen capturada por el móvil del usuario.

```
child: GestureDetector(
  onTap: () async {
    final picker = ImagePicker();
    final pickedFile = await picker.getImage(source: ImageSource.camera);

    if (pickedFile != null) {
      imagePath = pickedFile.path;
      navigateToResultadoClasificacion(context, imagePath!);
    } else {
      // Manejo de error si no se capturó una imagen
    }
  },
),
```


Figura 5

Código para pasar la imagen a la pantalla de resultados.

```
class CaptureScreen extends StatelessWidget {
  Future<void> navigateToResultadoClasificacion(BuildContext context, String imagePath) async {
    final status = await Permission.camera.request();
    if (status.isGranted) {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) => ResultadoClasificacionScreen(
            imagePath: imagePath,

```

Figura 6

Código para cargar el modelo y archivo de etiquetas.

```
Future<void> loadModel() async {
  try {
    await Tflite.loadModel(
      model: 'assets/mml_clasificador_fitbite.tflite',
      labels: 'assets/etiquetas_clasificador_fitbite.txt',
    );

```

Figura 7

Código para cargar el modelo y archivo de etiquetas.

```
child: GestureDetector(
  onTap: () async {
    final picker = ImagePicker();
    final pickedFile = await picker.getImage(source: ImageSource.camera);

    if (pickedFile != null) {
      imagePath = pickedFile.path;
      navigateToResultadoClasificacion(context, imagePath!);
    } else {
      // Manejo de error si no se capturó una imagen
    }
  },

```

Finalmente, la Figura 8 muestra cómo se asigna la etiqueta a una imagen mediante la variable “etiqueta”, la cual fue declarada en el código de la Figura 7.

E. Pantallas de la aplicación móvil

A continuación, se presentarán las pantallas de la aplicación más relevantes:

En la Figura 9, se aprecia la interfaz para iniciar sesión, conformada por los campos “correo electrónico” y “contraseña” y el botón “Ingresar” (pantalla izquierda). Si el usuario no tiene una cuenta, puede realizar el registro para usar la aplicación móvil (pantalla derecha).

La Figura 10 trata de la interfaz de tomar foto para, a partir de la imagen captura, el usuario obtenga las recomendaciones de comidas saludables.

La Figura 11 muestra la interfaz de comidas recomendadas según un ingrediente o insumo

F. Arquitectura de la Aplicación

Según el estudio de Rodríguez et al. (2022), la arquitectura MVC se basa en la separación de las responsabilidades en tres componentes principales: el modelo, que representa los datos y la lógica de negocio; la vista, encargada de la presentación de la información al usuario; y el controlador, que maneja las interacciones y coordina las acciones entre el modelo y la vista. Esta separación de preocupaciones facilita la modularidad, el mantenimiento y la reutilización del código, lo que resulta en un desarrollo más eficiente y escalable de aplicaciones. Además, el trabajo de García et al. (2022) destaca que la arquitectura MVC promueve la claridad y la estructura en el desarrollo de software, permitiendo

Figura 8

Código para dar etiqueta a una imagen.

```
RichText(
  textAlign: TextAlign.center,
  text: TextSpan(
    text: 'Se identificó como: ',
    style: TextStyle(
      fontSize: 16.0,
      color: Colors.black,
    ), // TextStyle
  ),
  children: [
    TextSpan(
      text: etiqueta,
      style: TextStyle(
        fontWeight: FontWeight.bold,
      ), // TextStyle
    ),
  ],
)
```

Figura 9

Pantalla Login/Registro de la aplicación móvil



Figura 10

Tomar fotos de insumos de comidas saludables



Figura 11

Lista de comidas recomendadas



un mejor control sobre el flujo de datos y la interacción con el usuario.

En la Figura 12, se muestra la arquitectura seguida: Modelo Vista Controlador (MVC). En esta imagen aparece una arquitectura de 3 capas Modelo - Vista - Controlador, el dispositivo que usa el usuario es la capa de interfaz, todos los procesos que realiza el app es la capa lógica y visualizamos una tercera capa, la cual es la conexión a la base de datos, realizado con PHP, la vista vendría a ser la app, los controladores que utiliza son un controlador de eventos que realiza las ejecuciones en la app y además existe un controlador a la base de datos que se le conoce como database controller y otro database controller para mostrar las recetas de la app.

Figura 12. Arquitectura del software

Se notan sus tres componentes: Modelo, Vista y Controlador. Para “modelo”, se usó el SGBD MySQL y TensorFlow Lite. Para “vista”, se usó el framework Flutter, basado en el lenguaje Dart. Para “controlador”, se usó el lenguaje PHP, para realizar consultas a la base de datos.

G. Tecnologías y herramientas utilizadas

A continuación, se enumeran las tecnologías y herramientas utilizadas en las áreas clave del proyecto:

G.1. Desarrollo de la Aplicación Móvil

- **Android Studio:** Según el artículo de Serrano et al. (2021), Android Studio ofrece un conjunto completo de herramientas y funcionalidades que facilitan el desarrollo de aplicaciones, como la edición de código, la depuración, la administración de proyectos y la interfaz gráfica de usuario (GUI) para diseñar y construir interfaces de usuario interactivas.
- **Trello:** Trello es una herramienta de gestión de proyectos basada en la web que permite a los equipos organizar y colaborar en sus tareas de manera visual y eficiente. Según el estudio de Corral et al. (2020), Trello utiliza un enfoque de tableros y listas para ayudar a los usuarios a visualizar y organizar sus proyectos y ofrece integraciones con otras herramientas y servicios para mejorar la colaboración y la productividad.

- **GitHub:** GitHub es una plataforma web de desarrollo colaborativo que permite a los desarrolladores alojar, gestionar y compartir proyectos de software de forma eficiente. Según el estudio de Gousios et al. (2020), GitHub proporciona un entorno centralizado para almacenar y versionar el código fuente, lo que facilita la colaboración entre desarrolladores y equipos distribuidos.
- **Figma:** Figma es una herramienta de diseño de interfaz de usuario (UI) basada en la web que permite a los diseñadores crear y colaborar en proyectos de diseño de manera eficiente. Según el estudio de Ye et al. (2021), Figma ofrece una plataforma en la nube que permite a los equipos trabajar en tiempo real en un entorno colaborativo. Los usuarios pueden crear diseños de interfaces interactivas, prototipos y animaciones utilizando una interfaz intuitiva y una amplia gama de herramientas y funciones.
- **Flutter:** Es un framework de desarrollo de aplicaciones móviles multiplataforma desarrollado por Google que permite crear aplicaciones nativas para iOS y Android utilizando un único código base. Según el estudio de Cheng et al. (2021), Flutter utiliza el lenguaje de programación Dart y proporciona un conjunto de widgets y herramientas que facilitan el desarrollo de interfaces de usuario atractivas y de alto rendimiento.
- **MySQL:** Es un sistema de gestión de bases de datos relacionales ampliamente utilizado en aplicaciones web y empresariales. Según el artículo de Elnikety et al. (2018), MySQL ofrece una plataforma robusta y confiable para almacenar, organizar y recuperar grandes volúmenes de datos de manera eficiente.
- **PHP (Hypertext Preprocessor):** Es un lenguaje de programación de código abierto ampliamente utilizado para el desarrollo de aplicaciones web dinámicas. Según el estudio de Wirth et al. (2020), PHP ofrece una sintaxis sencilla y fácil de aprender, lo que lo convierte en una opción popular entre los desarrolladores web.

Además, se listan las dependencias utilizadas en la aplicación móvil:

- **camera: ^0.10.0:** permite acceder a la(s) cámara(s) del móvil.
- **image_picker: ^0.8.4+4:** permite tomar nuevas fotos con la cámara del móvil.
- **mysql1:** permite conexión con MySQL.
- **permission_handler: ^10.2.0:** permite solicitar permisos de acceso a la cámara.
- **flutter_image_picker: ^1.1.2:** permite acceder a la API de TensorFlow Lite, la cual, admite clasificación de imágenes.

G.2. Modelo de clasificación de imágenes

Durante el desarrollo del proyecto, se empleó Google Colab, una poderosa plataforma para la ejecución de código Python en línea, según el estudio de Rahman et al. (2018), Google Colab proporciona un entorno de programación interactivo y gratuito que ofrece capacidades de ejecución de código, visualización de datos y documentación en un solo lugar.

Además, se aprovecharon herramientas clave como TensorFlow, un marco de trabajo de aprendizaje automático de código abierto desarrollado por Google que ofrece un entorno eficiente para construir y entrenar modelos de aprendizaje automático. Según el estudio de Abadi et al. (2022), TensorFlow proporciona una amplia gama de herramientas y recursos para la implementación de modelos de aprendizaje automático en diversos dominios, como visión por computadora, procesamiento de lenguaje natural y análisis de datos.

También se utilizaron TensorFlow Lite y Keras, las cuales fueron previamente presentadas y explicadas en la sección del esquema conceptual. TensorFlow Lite es una versión optimizada y ligera del marco de trabajo de aprendizaje automático TensorFlow, diseñada específicamente para la implementación de modelos de aprendizaje automático en dispositivos móviles y sistemas integrados. Según el estudio de Mehta et al. (2022), TensorFlow Lite ofrece un entorno eficiente para la inferencia de modelos de aprendizaje automático en plataformas con recursos computacionales limitados.

Keras es una biblioteca de aprendizaje profundo de código abierto que se ejecuta sobre TensorFlow. Proporciona una interfaz sencilla y fácil de usar para la construcción y el entrenamiento de redes neuronales. Según el estudio de Chollet et al. (2022), Keras está diseñado para facilitar el proceso de desarrollo de modelos de aprendizaje profundo, al proporcionar capas predefinidas, optimizadores y métricas comunes, lo que permite a los desarrolladores crear redes neuronales de forma rápida y eficiente.

Estas tecnologías y librerías de aprendizaje automático desempeñaron un papel fundamental en la implementación y el entrenamiento de los modelos utilizados en el proyecto.

G.3. Evaluación de la usabilidad

A continuación, se detallará la integración de diversas herramientas utilizadas para evaluar la usabilidad de la aplicación:

Se realizó un de usabilidad moderado a distancia, en el cual participaron individuos de entre 18 y 50 años. Para este propósito, se utilizó la plataforma de reuniones en línea de Google Meet. Durante las sesiones, se presentó a los participantes el propósito de la aplicación, se proporcionaron las instrucciones necesarias y se evaluó la usabilidad utilizando el System Usability Scale (SUS). Este cuestionario consta de 10 enunciados diseñados para medir la percepción general de usabilidad de la aplicación por parte de los participantes. Para recopilar las respuestas, se utilizó la herramienta Google Forms.

Los enunciados del cuestionario SUS se clasifican en una escala de puntuación del 1 al 5, donde 1 representa "muy en desacuerdo", 2 indica "en desacuerdo", 3 significa "ni de acuerdo ni en desacuerdo", 4 refleja "de acuerdo" y 5 representa "muy de acuerdo". En la Tabla 4 se detallan los enunciados específicos que se utilizaron en el cuestionario.

Tabla 4
Enunciados del System Usability Scale (SUS)

N°	Enunciado
E1	Me gustaría usar este sistema más a menudo.
E2	Me parece que el sistema es más complicado de lo que debería ser.
E3	Creo que el sistema es sencillo y fácil de usar.
E4	Necesito apoyo técnico para utilizar este sistema.
E5	Creo que el sistema funciona bien y está bien integrado.
E6	Creo que hay muchas irregularidades en el sistema.
E7	Creo que la mayoría de la gente puede aprender este sistema rápidamente.
E8	Creo que este sistema requiere mucho tiempo.
E9	Me siento seguro al utilizar este sistema.
E10	Creo que hay muchas cosas que aprender antes de poder empezar a utilizar este sistema.

El puntaje de usabilidad con SUS es calculado con la siguiente fórmula:

$$SUS = \frac{\sum_{i=1}^n (S_i - 5) + \sum_{i=1}^n (25 - S_i)}{n} * 2.5$$

Donde $\sum_{i=1}^n$ se refiere a la suma de los puntajes de los participantes a los 10 enunciados, $\sum_{i=1}^n S_i$ es la suma de los puntajes de enunciados impares, $\sum_{i=1}^n S_i$ es la suma de los puntajes de enunciados pares y n el número de participantes del estudio.

La Tabla 5 representa la escala para SUS propuesta por Lewis & Sauro (2018).

Tabla 5
Escala de calificación curva para el SUS (Lewis & Sauro, 2018)

Calificación	SUS	Rango Percentil
A+	84.1 - 100	96 - 100
A	80.8 - 84.0	90 - 95
A-	78.9 - 80.7	85 - 89
B+	77.2 - 78.8	80 - 84
B	74.1 - 77.1	70 - 79
B-	72.6 - 74.0	65 - 69
C+	71.1 - 72.5	60 - 64
C	65.0 - 71.0	41 - 59
C-	62.7 - 64.9	35 - 40
D	51.7 - 62.6	15 - 34
F	0 - 51.6	0 - 14

H. Métricas de Evaluación

Se consideraron las siguientes métricas para evaluar el modelo de clasificación de imágenes:

- *Recall*: conocido en español como sensibilidad, es entendida como una medida que evalúa la proporción de casos positivos que fueron clasificados de forma correcta (Borja et al., 2020).
- *Accuracy*: es una métrica de facilidad de cálculo, usada para comprender y evaluar la efectividad general de un modelo de Machine Learning.

En cuanto al *recall*, puede representarse de forma visual, gracias a la matriz de confusión. La matriz de confusión muestra la distribución de las predicciones sobre todas las clases en una vista concisa (Heydarian et al., 2022). Se distinguen 2 tipos de matriz de confusión, pero, en este caso, se usará la matriz de confusión normalizada, en vez de la no normalizada pues la primera permite conocer el porcentaje de predicciones verdaderas y falsas; la segunda, realiza un conteo por clase.

IV. RESULTADOS

Se analizaron los resultados obtenidos en esta sección, en torno a:

- Modelo de clasificación de imágenes
- Usabilidad
- Rendimiento

A. Modelo de clasificación de imágenes

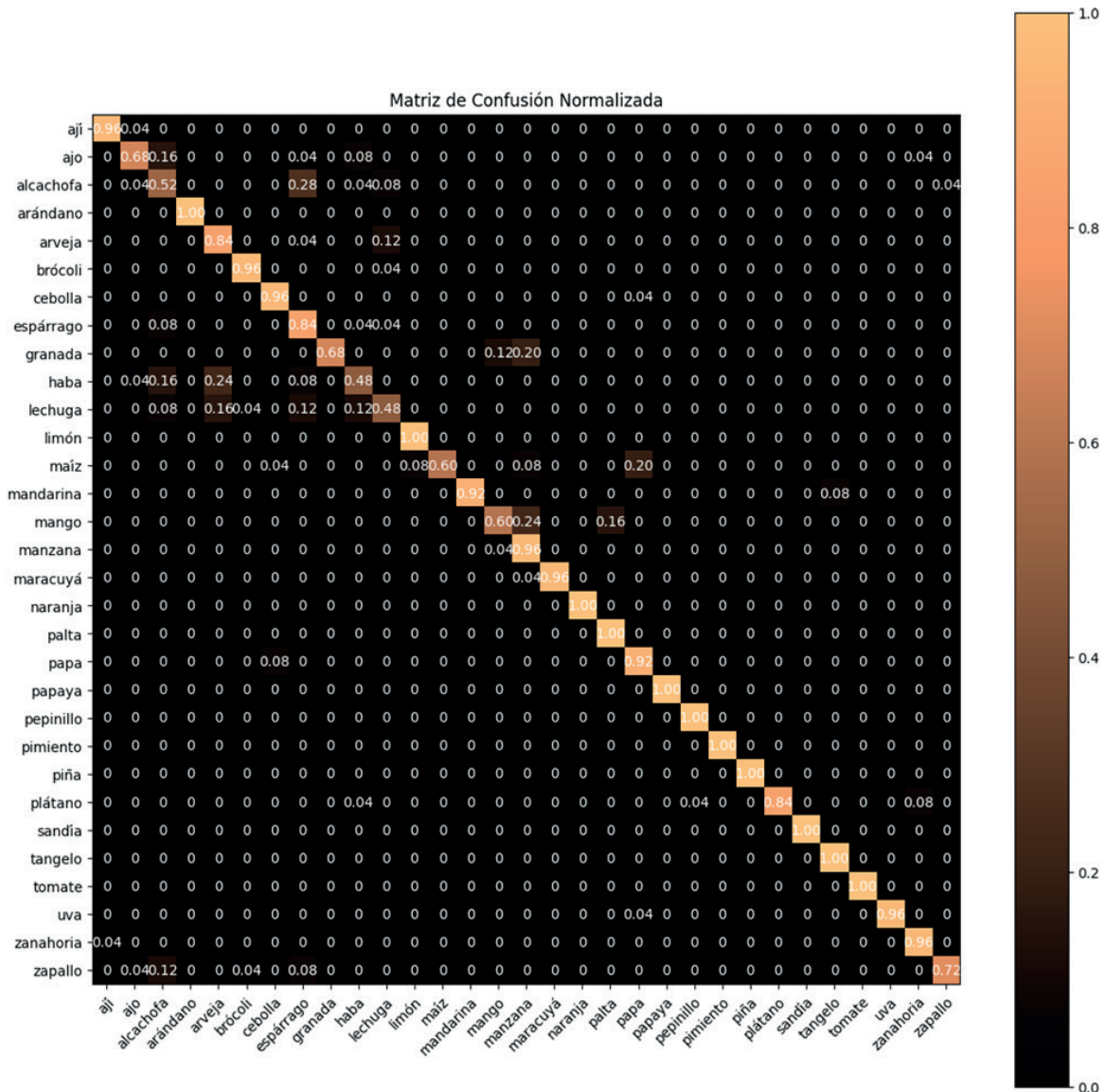
En la Figura 13, se muestra la matriz de confusión normalizada. Las clases “Haba” y “Lechuga” tuvieron el menor *recall*, ambas con 0.48, seguido de “Alcachofa” con 0.52.

Con el análisis de las filas de la matriz, se supo que la clase “Lechuga” fue identificada hasta con 5 etiquetas: “Alcachofa”, “Arveja”, “Brócoli”, “Espárrago” y “Haba”, siendo todas estas verduras verdes. Tras el análisis de las columnas, se supo que la etiqueta “Espárrago” fue otorgada a imágenes de otras 6 clases: “Ajo”, “Alcachofa”, “Arveja”, “Haba”, “Lechuga” y “Zapallo”.

Los valores con 0 entre las intersecciones de las filas y columnas (celdas), indican que una clase no fue clasificada en ningún caso con otra clase del modelo.

En la Figura 14, se muestra el *accuracy* del modelo de clasificación en torno a la métrica tras 20 épocas, comparando el desempeño del modelo para el conjunto de entrenamiento y de validación. A partir de la época 13, el conjunto de entrenamiento obtiene un valor de 1.0000. Desde la época 10, el conjunto de validación obtuvo valores entre 0.85 y 0.90, siendo 0.8619 el *accuracy* en la última época. Clasificó correctamente la clase a la que pertenecía una imagen en el 86.19% de los casos.

Figura 13
Matriz de confusión normalizada



B. Usabilidad

A continuación, se mostrarán unos gráficos de barras con las respuestas de los 17 encuestados para la Escala de Usabilidad del Sistema (SUS) para la aplicación "FITBITE"

Con respecto al uso del sistema, registramos las respuestas en la Figura 15.

Sobre la complejidad y facilidad del sistema, los encuestados respondieron: ver Figuras 16-17.

Respecto a la complejidad técnica del sistema, se obtuvo lo siguiente: ver Figuras 18-20.

Con respecto al tiempo de aprendizaje de las funcionalidades del sistema: ver Figura 21-24.

Para lograr obtener y calcular el puntaje de calificación de acuerdo a la escala de clasificación de la usabilidad SUS, se resolvió la siguiente operación: $\sum_{\#i=1}^n (i - 5)$, para lo cual, se obtuvo el valor 285 y para $\sum_{\#i=1}^n (25 - i)$, el resultado arrojó 266.

Entonces, la usabilidad en promedio es:

$$i = \frac{[(285) + (266)] * 2.5}{17} = 81.029$$

De acuerdo con la escala de clasificación de usabilidad del System Usability Scale (SUS), el aplicativo móvil obtuvo una calificación excepcional, alcanzando un puntaje de 81.029, lo cual corresponde a una calificación A.

Figura 14
 Accuracy del modelo de clasificación de imágenes

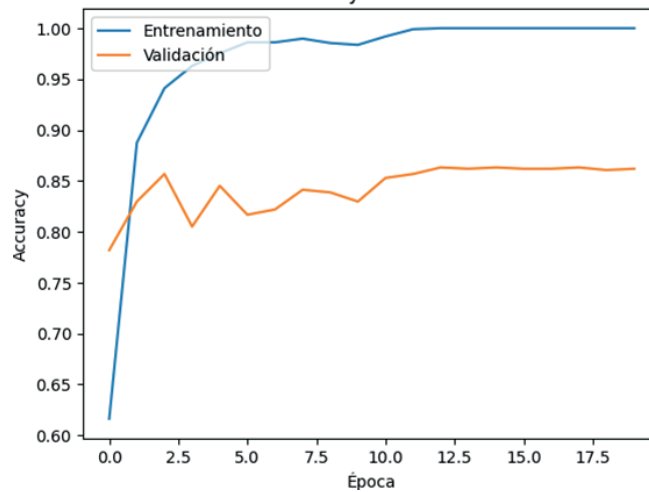


Figura 15
 Respuestas registradas de la pregunta 1 de la encuesta

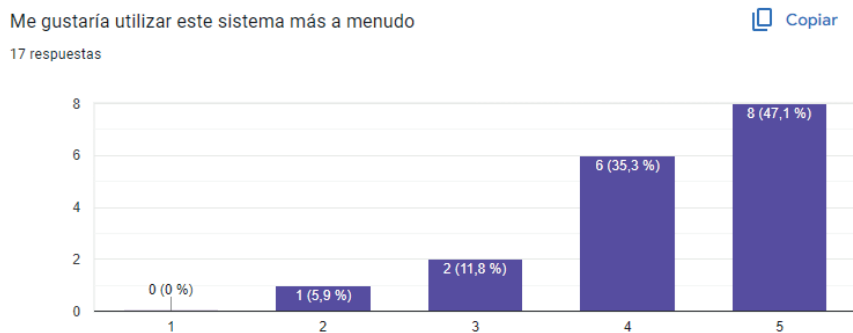


Figura 16
 Respuestas registradas de la pregunta 2 de la encuesta

Me parece que este sistema es más complicado de lo que debería ser
 17 respuestas

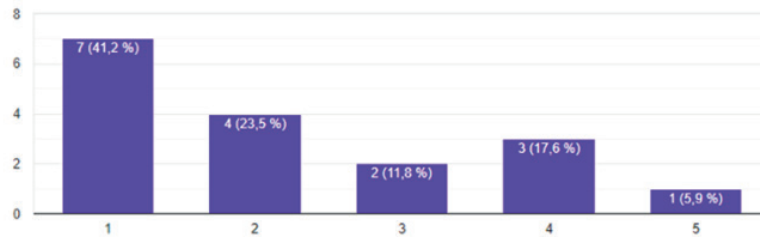


Figura 17
 Respuestas registradas de la pregunta 3 de la encuesta

Creo que el sistema es sencillo y fácil de usar
 17 respuestas

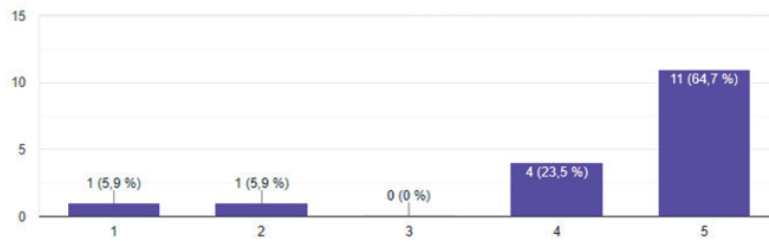


Figura 18
 Respuestas registradas de la pregunta 4 de la encuesta

Necesito apoyo técnico para utilizar este sistema
 17 respuestas

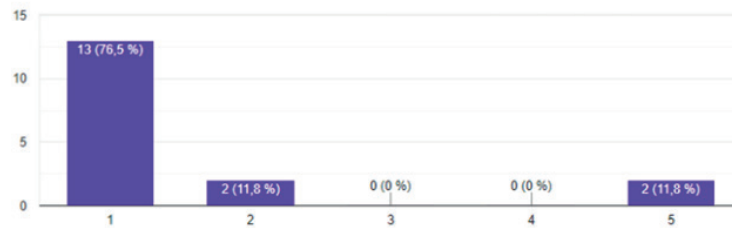


Figura 19
 Respuestas registradas de la pregunta 5 de la encuesta

Creo que el sistema funciona bien y está bien integrado
 17 respuestas

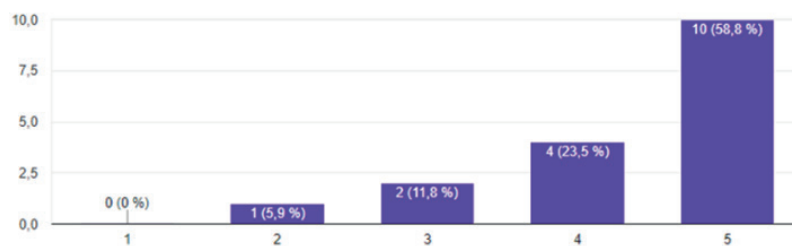


Figura 20
 Respuestas registradas de la pregunta 6 de la encuesta

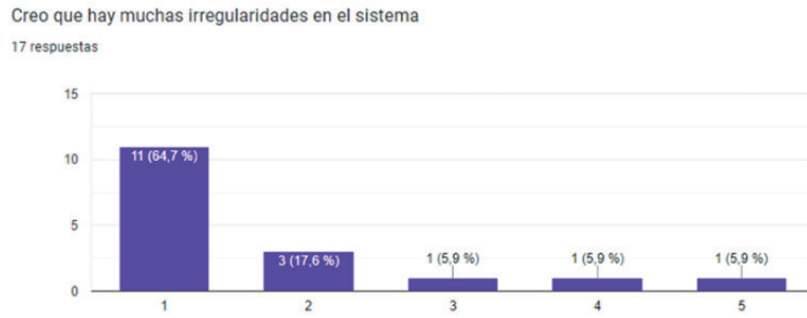


Figura 21
 Respuestas registradas de la pregunta 7 de la encuesta



Figura 22
 Respuestas registradas de la pregunta 8 de la encuesta

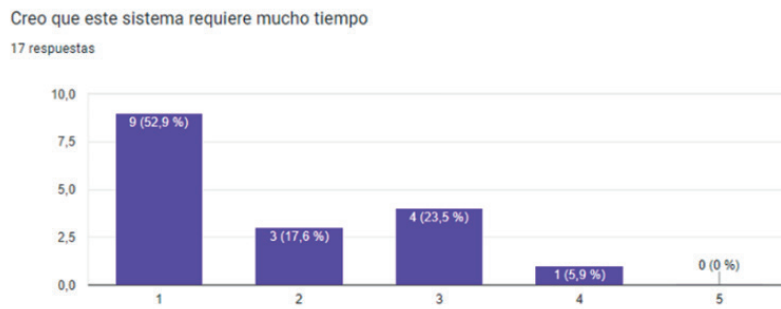


Figura 23
 Respuestas registradas de la pregunta 9 de la encuesta

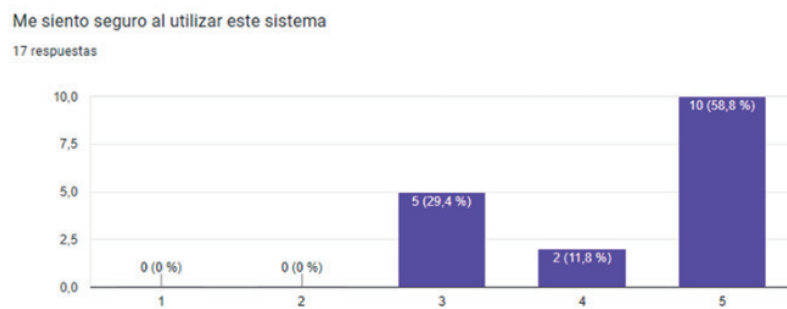
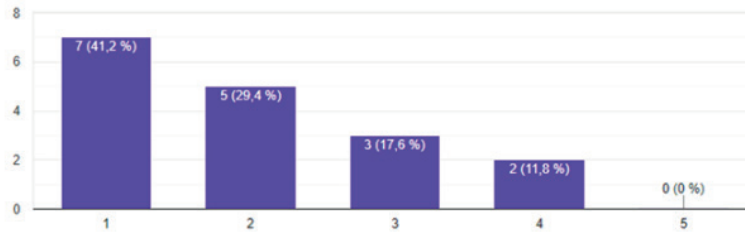


Figura 24

Respuestas registradas de la pregunta 10 de la encuesta

Creo que hay muchas cosas que aprender antes de poder empezar a utilizar este sistema

17 respuestas



IV. CONCLUSIONES

Concluimos que el desarrollo de nuestra aplicación móvil que utiliza un modelo de clasificación de imágenes con TensorFlow Lite ha logrado cumplir el objetivo de permitir a los usuarios identificar alimentos a través de la cámara de sus dispositivos móviles y obtener recomendaciones de comidas saludables basadas en estos alimentos.

La capacidad de identificar alimentos en tiempo real proporciona una experiencia práctica y accesible para los usuarios, permitiéndoles tomar decisiones informadas sobre su alimentación y mejorar sus hábitos nutricionales. Además, al ofrecer recomendaciones de comidas saludables basadas en los alimentos identificados, nuestra aplicación promueve una dieta equilibrada y contribuye al bienestar general de los usuarios.

La utilización de la cámara del dispositivo móvil como una herramienta interactiva y visual proporciona una experiencia de usuario intuitiva y enriquecedora. Los usuarios pueden explorar una variedad de comidas saludables que pueden prepararse a partir de los alimentos identificados, lo que fomenta la creatividad culinaria y la diversidad en la elección de comidas saludables.

Una primera parte del trabajo, dado el aporte de clasificación de imágenes en la aplicación móvil FitBite, fue la obtención del modelo de clasificación de imágenes. Se obtuvo un accuracy de 0.8619 para el modelo, siendo un valor aceptable para clasificar una fruta o verdura, aunque dicha métrica puede mejorarse aún más mediante el aumento del tamaño del conjunto de datos utilizado, el ajuste de los hiperparámetros del modelo o el uso de una capa Dropout para una mayor generalización del modelo. También, se evidenció que las clases que tuvieron mayor número de posibles etiquetas son aquellas de colores similares.

Se concluye que el diseño e implementación de interfaces de usuario fáciles de usar para nuestra aplicación móvil ha sido un aspecto fundamental en el logro de nuestro objetivo. Al poner un fuerte enfoque en la experiencia del usuario, hemos creado una interfaz intuitiva y atractiva que permite a los usuarios interactuar de manera efectiva con la aplicación.

En cuanto a la usabilidad, con SUS se obtuvo un puntaje de 81.029, el cual otorga a la aplicación móvil la calificación A. Ello demuestra que la aplicación móvil cumple con las expectativas del usuario. Si bien dicha calificación es muy buena, es posible mejorar el atributo usabilidad para la aplicación FitBite para obtener una mayor satisfacción de los usuarios y tengan una mejor experiencia de usuario.

REFERENCIAS

- [1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2022). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.
- [2] Al-Ibrahim, A. & Jackson, R. (2019). *Healthy eating index versus alternate healthy index in relation to diabetes status and health markers in U.S. adults: NHANES 2007–2010*, *Nutrition Journal*, vol. 18, n.º 1, p. 26. <https://doi.org/10.1186/s12937-019-0450-6>.
- [3] Ahmed, M., Mamun, S. & Asif, A (2021). *DCNN-Based Vegetable Image Classification Using Transfer Learning: A Comparative Study*, 2021 5th International Conference on Computer, Communication and Signal Processing (ICCCSP), pp. 235-243. Chennai, India. <https://doi.org/10.1109/ICCCSP52374.2021.9465499>.

- [4] Akbaraly, T. et al. (2018). *Association of Long-Term Diet Quality with Hippocampal Volume: Longitudinal Cohort Study*, *The American Journal of Medicine*, vol. 131, <https://doi.org/10.1016/j.amjmed.2018.07.001>.
- [5] Ann, E., Hao, N., Wei, G., & Hee, K. (2021). "Feast In: A machine learning image recognition model of recipe and lifestyle applications". MA-TEC Web of Conferences, 335, 04006. <https://doi.org/10.1051/mateconf/202133504006>.
- [6] Belilovsky, E., Eksombatchai, P., Verma, A., Ramanan, D., & Mittal, P. (2020). Automated Hyperparameter Optimization for Recurrent Neural Networks with Repeated Training Validation. arXiv preprint arXiv:2004.12928.
- [7] Borja, R., Monleón, A., & Rodellar, J. (2020). *Estandarización de métricas de rendimiento para clasificadores Machine y Deep Learning*. Barcelona, España. *Revista Ibérica de Sistemas y Tecnologías de Información*.
- [8] Brewer, J., Santos, M., Lopez, M., Paz, V. & Chaparro, M. (2021). *Use of Formal and Informal Food Resources by Food Insecure Families in Lima, Peru: A Mixed-Methods Analysis*, *Journal of Community Health*, vol. 46, n.o 6, pp. 1069-1077. <https://doi.org/10.1007/s10900-021-00989-y>.
- [9] Brownlee, J. (2019). *Deep Learning for Computer Vision: Image Classification, Object Detection, and Face Recognition in Python*. Machine Learning Mastery [En línea]. <https://books.google.com.pe/books?id=DOamDwAAQBAJ>.
- [10] Campos, P. & Rioseco, B. (2020). Sistema de recomendación de recetas saludables, basado en la preferencia y necesidad nutricional del usuario. Universidad del Bío-Bío. <http://repobib.ubiobio.cl/jspui/handle/123456789/3724>.
- [11] Castellanos, J. y Niño, E. (2021). Prototipo de aplicación móvil para la sugerencia de recetas culinarias utilizando algoritmos de inteligencia artificial <http://hdl.handle.net/20.500.12749/16258>.
- [12] Chen, L., Zhang, Y., & Wei, L. (2021). Deep Learning for Image Classification: Methods, Challenges, and Opportunities. *Journal of Computational Science*, 54, 101262.
- [13] Cheng, W., Zang, C., Yu, S., & Liu, Y. (2021). A Study on Cross-platform Mobile App Development Frameworks: React Native and Flutter. *Proceedings of the 6th International Conference on Robotics, Control and Automation*, 651-656.
- [14] Chollet, F., Allaire, J. J., & Chollet, F. (2022). *Deep Learning with Python*. Manning Publications.
- [15] Cornejo, L. & R. Urbano (2021). Solución Tecnológica para el control de una dieta saludable empleando Visual Recognition en el Sector de Nutrición en el Perú. Tesis para optar el título profesional de Ingeniero de Sistemas de Información. Facultad de Ingeniería, Programa Académico de Ingeniería de Sistemas de Información, Universidad Peruana de Ciencias Aplicadas. <http://hdl.handle.net/10757/660323>.
- [16] Corral, L., Fernández-Alemán, J. L., Toval, A., & Escalona, M. J. (2020). Agile Practices in the Software Industry: A Systematic Mapping Study. *Journal of Systems and Software*, 169, 110690.
- [17] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2019). ImageNet: A Large-Scale Hierarchical Image Database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248-255.
- [18] Elnikety, S., Alomari, M., Kapadia, S., Elmore, A., & Stoica, I. (2018). The Case for Determinism in Database Systems. *Proceedings of the ACM Symposium on Cloud Computing*, 245-258.
- [19] FAO & OMS (2020). *Dietas saludables sostenibles - Principios Rectores*. Roma. <https://doi.org/10.4060/ca6640es>.
- [20] FAO, FIDA, OPS, WFP & UNICEF (2020). *Pa-norama de la seguridad alimentaria y nutrición en América Latina y el Caribe 2020*. Santiago de Chile. <https://doi.org/10.4060/cb2242es>.
- [21] FAO, IFAD, PAHO, UNICEF & WFP (2023), *Regional Overview of Food Security and Nutrition – Latin America and the Caribbean 2022: towards improving affordability of healthy diets*. Santiago. <https://doi.org/10.4060/cc3859en>.
- [22] García, L., Muñoz, L., & Sánchez, E. (2022). Evaluation of the MVC Architecture for Web Applications. In *International Conference on Information Systems and Technologies* (pp. 102-113). Springer.
- [23] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- [24] Gousios, G., Vasilescu, B., Serebrenik, A., & Zaidman, A. (2020). The GHTorrent Dataset and Tool Suite. *Proceedings of the 17th International Conference on Mining Software Repositories*, 233-236.

- [25] Heydarian, M., Doyle, T. & Samavi, R. (2022). *MLCM: Multi-Label Confusion Matrix*, *IEEE Access*, vol. 10, pp. 19083-19095. <https://doi.org/10.1109/ACCESS.2022.3151048>.
- [26] Hossain, M. S., Muhammad, G., & Khan, A. (2020). Image Classification Using Machine Learning: A Review. *Proceedings of the 6th International Conference on Computer Science, Engineering and Applications*, 1-6.
- [27] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2020). Densely Connected Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2), 470-483.
- [28] Instituto Nacional de Estadística e Informática (2021). "Perú: Enfermedades No Transmisibles y Transmisibles, 2020" [En línea]. https://proyectos.inei.gob.pe/endes/2020/SALUD/ENFERMEDADES_ENDES_2020.pdf.
- [29] Ipsos (2019). *Global Views on Food: What are the world's food habits* [En línea]. https://www.ipsos.com/sites/default/files/ct/publication/documents/2019-08/ipsos_global_advisor_views_on_food_120418_draft2_0.pdf.
- [30] Jiang, H., Wang, W., Liu, M., Nie, L., Duan, L., & Xu, C. (2019). *Market2Dish: A Health-aware Food Recommendation System. Proceedings of the 27th ACM International Conference on Multimedia - MM '19*. <https://doi.org/10.1145/3343031.3350594>.
- [31] Kaiming, H., Xiangyu, Z., Shaoqing, R., & Jian, S. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- [32] Kong, N., Moy, F., Ong, S., Tahir, G. & Loo, C. (2023). *MyDietCam: Development and usability study of a food recognition integrated dietary monitoring smartphone application. DIGITAL HEALTH*, vol. 9, p. 205520762211493. <https://doi.org/10.1177/20552076221149320>.
- [33] Kul, S., Sayar, A. (2022). *A Smart Recipe Recommendation System Based on Image Processing and Deep Learning*. In: Ben Ahmed, M., Boudhir, A.A., Karaş, İ.R., Jain, V., Mellouli, S. (eds) *Innovations in Smart Cities Applications Volume 5*. SCA 2021. *Lecture Notes in Networks and Systems*, vol 393. Springer, Cham. https://doi.org/10.1007/978-3-030-94191-8_83.
- [34] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
- [35] Lewis, J. & Sauro, J. (2018). *Item benchmarks for the system usability scale*. *Journal of Usability Studies*, Vol. 13, Issue 3, pp. 158-167.
- [36] Li, Z., Liu, F., Yang, W., Peng, S. & Zhou, J. (2022). *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects, IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, n.º 12, pp. 6999-7019. <https://doi.org/10.1109/TNNLS.2021.3084827>.
- [37] Li, J., Qin, L., & Wang, L. (2020). Deep Learning for Image Classification: A Review. *Neurocomputing*, 396, 386-403.
- [38] Liu, Z., Zhang, C., Zhang, C., & Xu, C. (2020). Image Classification Using Convolutional Neural Networks: A Comprehensive Review. *Neural Computing and Applications*, 32(11), 7655-7669.
- [39] Loshchilov, I., & Hutter, F. (2020). Decoupled Weight Decay Regularization. *International Conference on Learning Representations (ICLR)*.
- [40] Mehta, D., Patil, P., Phalke, A., Patel, R., & Choudhary, A. (2022). TensorFlow Lite: Machine Learning Inference on Edge Devices. *Proceedings of the International Conference on Computer and Communication Technology*, 145-150.
- [41] Mejía, J. & Insuasty, J. (2022). Plan de negocio e innovación para el emprendimiento y diseño de un aplicativo móvil destinado al e-commerce de productos saludables. *Universidad Militar Nueva Granada*. <http://hdl.handle.net/10654/41201>.
- [42] Méndez, I. (2021). Desarrollo y validación de un índice de calidad de dieta basado en las recomendaciones de las Guías Alimentarias para la Población Argentina. Tesis de Maestría en Nutrición Humana. Facultad de Ciencias Médicas, Departamento de Postgrado, Universidad Nacional de La Plata. <https://doi.org/10.35537/10915/119372>.
- [43] Ministerio de Desarrollo Agrario y Riego (2022). Situación actual de las frutas y verduras en el Perú: Producción, Exportación e Importación. <http://hdl.handle.net/20.500.13036/1227>.
- [44] Miller, V. et al. (2022). *Global dietary quality in 185 countries from 1990 to 2018 show wide differences by nation, age, education, and urbanicity*, *Nature Food*, vol. 3, n.º 9, pp. 694-702, <https://doi.org/10.1038/s43016-022-00594-9>.

- [45] Minut, M. (2021). Fruits-262. <https://www.kaggle.com/datasets/aelchiminut/fruits262/versions/1>.
- [46] Oltean, M. (2021). Fruits 360. <https://www.kaggle.com/datasets/moltean/fruits>.
- [47] Organización Panamericana de la Salud (2019). Alimentos y bebidas ultraprocesados en América Latina: ventas, fuentes, perfiles de nutrientes e implicaciones normativas. Washington, D. C. <https://doi.org/10.37774/9789275320327>.
- [48] Rabanal, J. (2019). *Programa de Educación Nutricional, con abordaje multidisciplinario, para desarrollar la cultura alimentaria en estudiantes de Educación Secundaria-Huamachuco, 2017, Rev. Cienc. Tecnol*, 15(4): 133 - 142, ISSN 1810-6781. Universidad Nacional de Trujillo. Escuela de Postgrado. <https://revistas.unitru.edu.pe/index.php/PGM/article/view/2674>.
- [49] Rahman, S., Islam, M. Z., Paul, S., & Rashid, A. (2018). Google Colaboratory: A Research Tool for Data Science and Machine Learning. arXiv preprint arXiv:1808.10635.
- [50] Rao, M. (2023). *A Comparative Analysis of Deep Learning Frameworks and Libraries, Int J Intell Syst Appl Eng*, vol. 11, no. 2s, pp. 337–342. ISSN: 2147-6799. <https://ijisae.org/index.php/IJISAE/article/view/2707/1287>.
- [51] Rodríguez, J. M., Pérez, A., & Gutiérrez, G. (2022). An Overview of Model-View-Controller Architectures for Web Applications. In *International Conference on Advanced Information Systems Engineering* (pp. 179-194). Springer.
- [52] Schwaber, K. & Sutherland, J. (2020). La Guía de Scrum - La Guía Definitiva de Scrum: Las Reglas del Juego [En línea]. <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf>.
- [53] Serrano, M., Vara, J. M., & Marcos, E. (2021). Static Analysis for Improving Software Quality in Android Development: A Systematic Mapping Study. *Information and Software Technology*, 130, 106445.
- [54] Seth, K. (2022). Fruits and Vegetables Image Dataset. <https://www.kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition>.
- [55] Singh A, Bhadani R. (2020). *Mobile Deep Learning with TensorFlow Lite, ML Kit and Flutter: Build Scalable Real-World Projects to Implement End-to-End Neural Networks on Android and IOS / Anubhav Singh, Rimjhim Bhadani*. 1st edition. Packt. https://www.google.com.pe/books/edition/Mobile_Deep_Learning_with_TensorFlow_Lit/sbTbDwAAQBAJ?hl=es-419&gbpv=0.
- [56] Win & Datum Internacional (2021). *Salud WWS 2021* [En línea]. [https://www.datum.com.pe/new_web_files/files/pdf/WIN%20y%20DATUM%20-%20Health%20\(Peru\)%20V5_220407041045.pdf](https://www.datum.com.pe/new_web_files/files/pdf/WIN%20y%20DATUM%20-%20Health%20(Peru)%20V5_220407041045.pdf).
- [57] Wirth, F., Schmieders, E., & Ritter, T. (2020). Practical Server-Side Web Development with Node.js and PHP. *Proceedings of the International Conference on Information and Communication Technology*, 45-51.
- [58] Ye, H., Ouyang, J., Yang, H., Wang, Z., & Wang, X. (2021). Deep Learning Based Interaction Design in UI. *Proceedings of the 14th International Conference on Computer Science and Education*, 1211-1216.
- [59] Zenun, F., Fallaize, R., Weech, M., Hwang, F. & Lovergrove, J. (2022). *Effectiveness of Web-Based Personalized Nutrition Advice for Adults Using the eNutri Web App: Evidence From the EatWellUK Randomized Controlled Trial, Journal of Medical Internet Research*, vol. 24, pp. 1-17. <https://doi.org/10.2196/29088>.
- [60] Zhang, J. (2023). Southeast Asia Common Fruit and Vegetables. <https://www.kaggle.com/datasets/jingyazhang97/safv-30>.

Financiamiento:

Propio

Conflictos de interés:

Los autores declaran no tener conflictos de interés.

Contribuciones de autoría:

Juan Martín Domínguez Matos: Desarrollo la parte del Backend de la solución y redactó el artículo científico.

Jeanpiere Julian Palacios Barrutia: Desarrollo la parte del Frontend de la solución y redactó el artículo científico.

Flavia Francesa Abanto Salas : Redacción del artículo científico.

Ivan Carlo Petrik Azabache: Asesor y revisó el artículo científico y la solución móvil y web.