

**TITULO DEL PROYECTO**

**Software inteligente para realizar recetas a partir de vegetales, carbohidratos y carnes**

Universitario (a): José Fernando Alfaro Ayzama

Carrera: Ing. Ciencias De La Computación

Docente: Ing. Carlos Walter Pacheco Lora

Sucre

# Antecedentes

Las personas enfrentan desafíos importantes al llevar una dieta saludable debido a la falta de información precisa sobre el contenido calórico y nutricional de sus comidas.

# Problema principal

La dificultad para identificar adecuadamente los alimentos que se consumen y calcular con precisión sus calorías puede afectar negativamente la salud de las personas. El desconocimiento o dificultad de generar recetas saludables basadas en los alimentos disponibles limita la adopción de hábitos alimenticios adecuados.

Gran parte de la población desconoce cómo combinar adecuadamente los alimentos para obtener recetas equilibradas y saludables. Esta situación es definitiva para quienes desean mejorar o adecuar su alimentación ante ciertas condiciones o exigencia que su estado de salud requiere

# Abordaje de la solución

Para abordar esta problemática, se propone el desarrollo de un software que identifique alimentos a partir de imágenes capturadas desde el celular, calcule sus calorías en base a su tamaño y genere recetas balanceadas. El sofware funcionará de la siguiente manera:

Se empleará un modelo de visión por computadora basado en Yolo-segmv8n para identificar alimentos específicos como vegetales, carbohidratos y carnes.

Para la estimación de tamaño, volumen aproximado y porciones de los alimentos se utilizara Depth Anything, además que será apoyado en una base estimada de calorías para poder calcular las calorías de los alimentos reconocidos.

Para la generación de la receta se consumirá la extensión de un modelo entrenado gpt-3, brindando una receta balanceada según los alientos reconocidos.

El conjunto de datos se construirá mediante la captura de imágenes y recopilación de datasets con sus respectivas segmentaciones, para el entrenamiento.

# Objetivo General

Gestionar y recomendar una combinación de alimentos que garanticen en al menos un 90% los requerimientos alimentarios suficientes de una persona en la comida principal.

# Objetivos Específicos

* Identificar los elementos teóricos del estado del arte en cuanto a **métodos para calcular las calorías de un alimento específico** a partir de sus características visuales, explorando datos nutricionales y técnicas existentes para estimar su valor calórico. Además, investigar fuentes y estrategias para generar recetas saludables basadas en un conjunto específico de alimentos.
* Identificarlos enfoques y metodologías clave del estado del arte en base a **proyectos similares** que hayan aplicado inteligencia artificial en la identificación de alimentos, cálculo de calorías y generación de recetas, para seleccionar las técnicas más efectivas.
* A**ptar y ajustar un modelo** que identifique alimentos a partir de imágenes, calcule su contenido calórico y genere recetas saludables, utilizando técnicas de visión por computadora y mapeo de imágenes para estimar tamaños
* Validar y evaluar **el modelo** mediante la comparación de sus resultados con datos reales y métodos tradicionales, asegurando su precisión en la identificación de alimentos, cálculo de calorías y generación de recetas.

# Fundamentos teóricos considerados en el trabajo

## Ámbito de la inteligencia artificial, técnicas, algoritmos, modelos base, entre otros

Este proyecto implementa un conjunto de técnicas de cálculo matemático para calorías, algoritmos de visión por computadora para la detección de alimentos, algoritmos de mapeo monocular en imágenes para calcular tamaños como Depth Anything. El modelo base seleccionado para la detección de imágenes es Yolov8n, elegido por su capacidad de detectar objetos con alta precisión y rapidez en dispositivos móviles.

**Técnicas utilizadas:**

* **Preprocesamiento de imágenes:** Se emplea Roboflow para segmentar y normalización de tamaño a 640×480 píxeles para estandarizar la resolución. Cada imagen es segmentada y etiquetada por su clase.
* **Reconocimiento de imágenes:** Se empela yolo-segmv8n para entrenar este modelo con las imágenes procesadas para el reconocimiento de alimentos que corresponden a carnes, carbohidratos y vegetales.
* **Calculo de tamaños y orientación de cámara:** Se emplea Depth Anything, para primero calibrar las la posición de la cámara móvil, asegurando la toma de fotos siempre desde un ángulo fijo, para no variar el tamaño de los alimentos reconocidos para su posterior cálculo de calorías, basadas en el volumen calculado y en un diccionario de datos fijos.
* **Generación de receta:** Se le utilizara el modelo ya entrenado gpt-3 para la generación de recetas a partir de la lista recibida de alimentos reconocidos.

**Algoritmos aplicados:**

* **Yolo-segv8n:** Aplicación del algoritmo de detección añadiendo una rama para predecir máscaras de segmentación para cada objeto detectado, permitiendo identificar la ubicación y la forma precisa de los objetos en la imagen de manera eficiente.
* **Depth Anything:** Un modelo fundacional para la estimación de profundidad monocular. Aprende representaciones de profundidad robustas a partir de una sola imagen.

## Ámbito al que se aplicó la inteligencia artificial

La Inteligencia Artificial en alimentos surgió a mediados de 2010 con visión para detección (ej., Im2Calories de Google). A finales de esa década, aplicaciones usaban **CNNs** para reconocer alimentos en el seguimiento dietético. Paralelamente, la estimación calórica se integró con **bases de datos** en aplicaciones como MyNetDiary. A principios de 2020, **CNNs** más sofisticadas analizaron porciones en aplicaciones calóricas. Modelos **basados en Transformer** como versiones tempranas de GPT iniciaron la generación básica de recetas. Desde mediados de 2020, **LLMs** avanzados (ej., GPT-3, Gemini) permiten análisis nutricional detallado y creación compleja de recetas, como en ChefGPT

En este proyecto se aplica en el ámbito de alimentación mediante la generación de recetas a partir de alimentos comunes clasificados en carnes, carbohidratos y vegetales, del entorno local, utilizando yolosegm-v8n para la detección de alimentos, depth anything para el buen posicionamiento de la cámara móvil y el cálculo de tamaño de los alimentos y gpt-3 para la generación balanceada de recetas con ingredientes detectados.

# Metodología

## Sustento metodológico

Las metodologías empleadas en este proyecto son:

* **Recopilación de datos:**

Se realizará una recolección de imágenes de alimentos crudos con distintas perspectivas y ángulos, también se recolectará imágenes de datasets de alimentos.

* **Preprocesamiento de datos:**

Las imágenes recolectadas se procesaran en la aplicación de roboflow, donde se procederá a clasificar los alimentos según su clase más su respectivo segmentado, para posteriormente aplicar data augmentation y dividir en datos de validación, entrenamiento.

* **Entrenamiento del modelo yolosegm-v8n:**

Se entrenara el modelo con los datos pre procesados, para que el modelo aprenda a reconocer y segmentar los alimentos crudos.

* **Calibración y cálculo de tamaño de alimentos con Depth Anything:**

Una vez finalizado el entrenamiento se integra con el modelo de anotación para el posterior cálculo de cantidad y tamaño de los alimentos detectados. Para luego calcular las calorías en base a datos estáticos y cálculos.

* **Generación de receta con gpt-3:**

Una vez se obtengan los datos obtenidos de los alimentos reconocidos más sus calorías y sus porciones, se pasaran a un modelo para que genere recetas balaceadas.

* **Desarrollo de la aplicación móvil:**

Se desarrollará una interfaz móvil que permita a los usuarios capturar fotos de los alimentos, para su posterior generación de receta con datos calóricos de cada ingrediente usado para le receta.

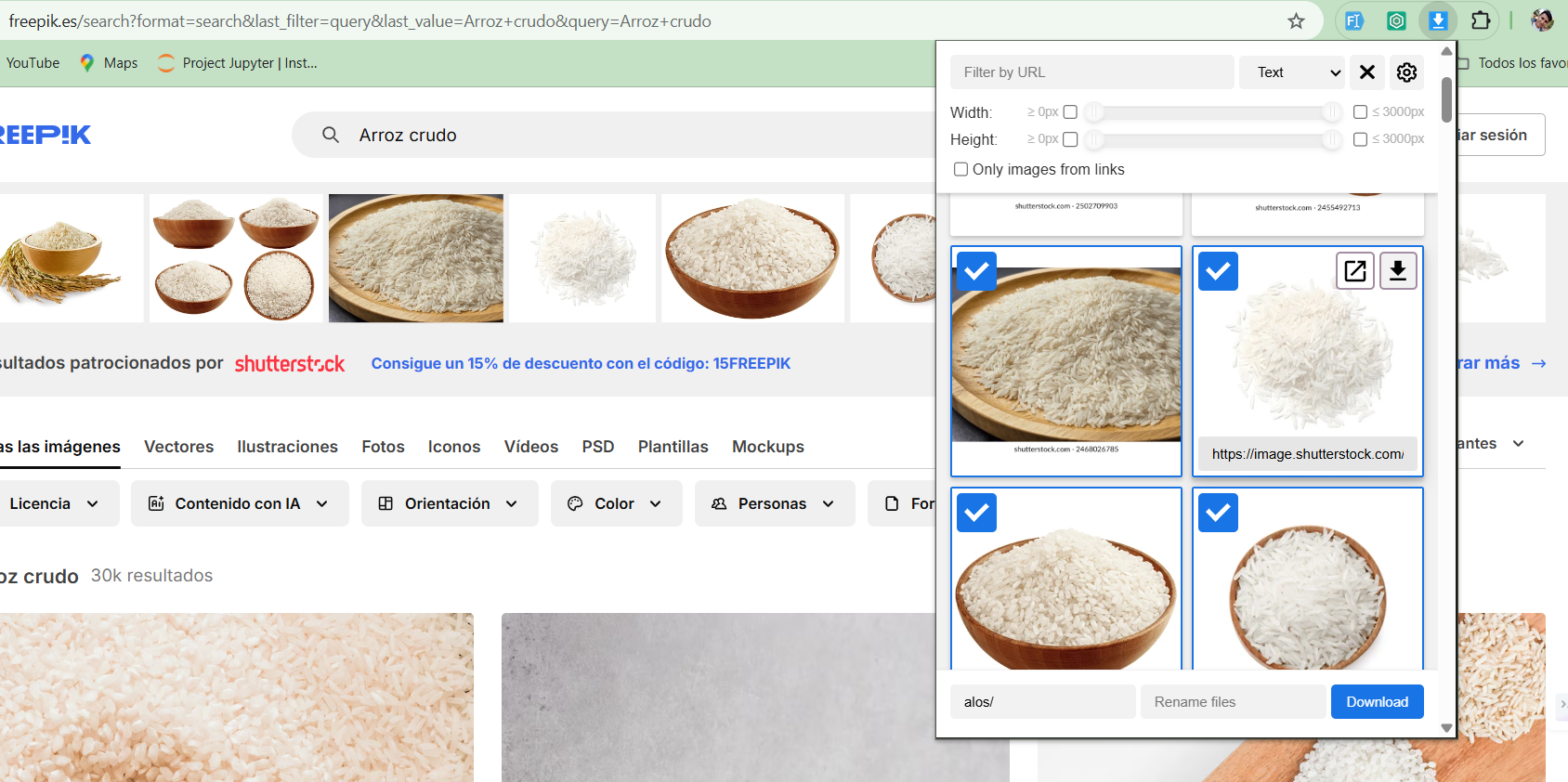
## Técnicas de recolección de datos

* **Recolección de fotos reales :**

Se recopilaron imágenes de los alimentos en mercados y entornos donde existían dichos alimentos con cámaras de teléfono.

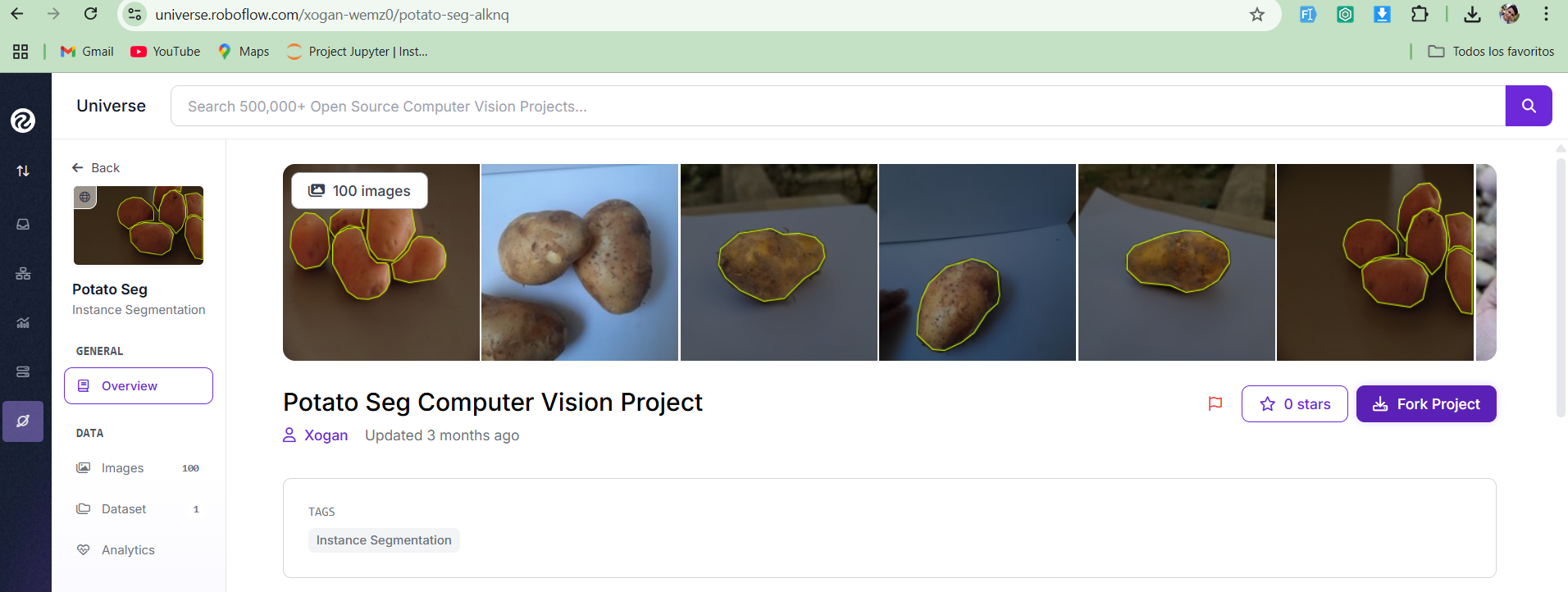
* **Recolección de imágenes de internet :**

También se procedió a descargar imágenes de alimentos de internet para aumentar los datos.



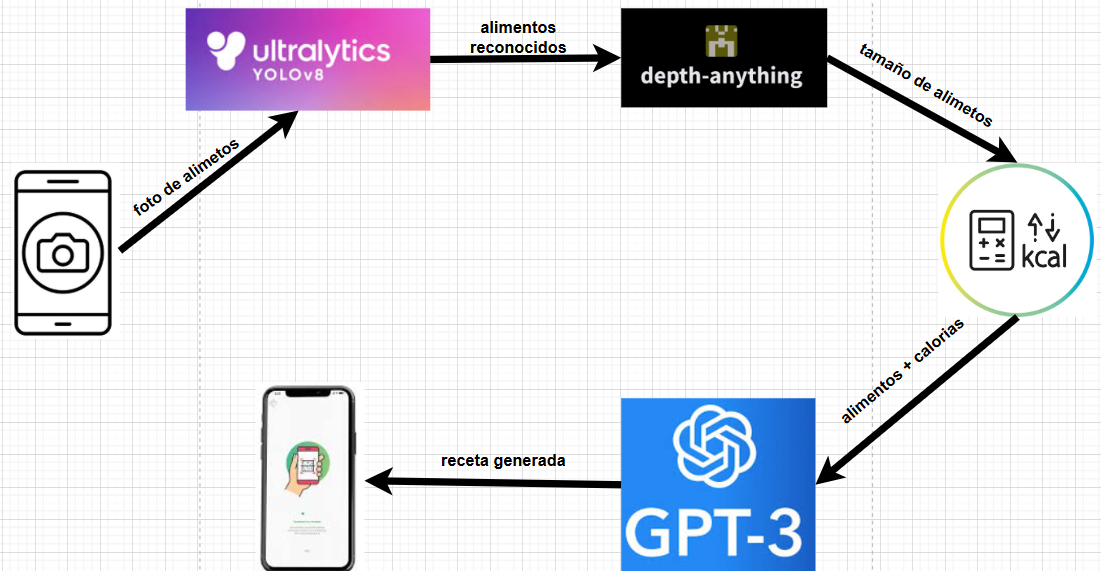
* **Recolección de imágenes de Datasets existentes :**

Se recolectaron imágenes de algunos datasets que tenían alimentos que coincidían con nuestro proyecto



## Materiales y herramientas

### Arquitectura Software Desarrollado



* La aplicación móvil desarrollada con Flutter, captura la imagen y la envía al modelo Yolov8 para el reconocimiento de los alimentos detectados.
* El modelo Yolov8 recibe la foto, la procesa, reconoce y segmenta los alimentos reconocidos en la foto y manda la imagen con los alimentos reconocidos a Depth Anything este calcula el tamaño y las porciones de cada alimento reconocido para posteriormente realizar el cálculo de las calorías.
* Se calculan las calorías en base a el tamaño de cada alimento con información de datos fijos de los alimentos, generando una lista de alimentos con su caloría y su porción que servirán como ingredientes base de nuestra receta.
* El modelo de gpt-3 recibe la lista de ingredientes y genera una receta para mostrar al usuario.

### Esquema y descripción de componentes de hardware.

### 



PC:

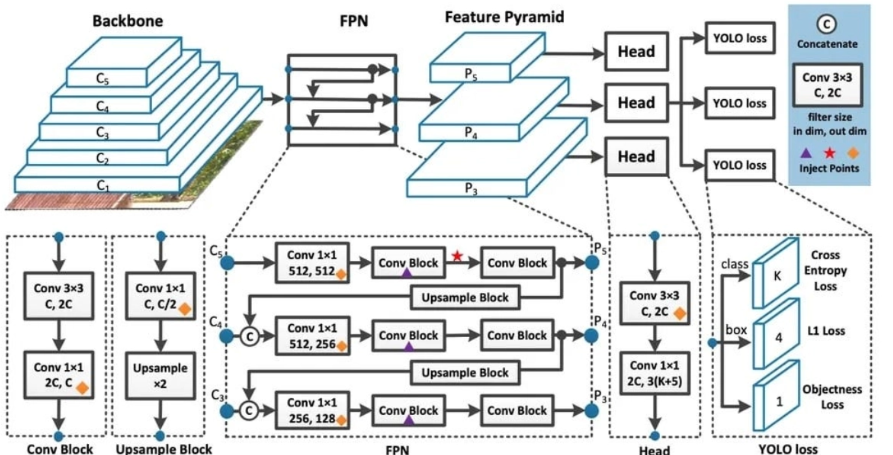
* + **CPU:** Unidad central de procesamiento, encargado de ejecutar las instrucciones y realizar los cálculos necesarios para el reconocimiento de los alimentos, cálculo de calorías y generación de recetas.
  + **SSD:** Unidad de Estado Sólido, utilizada para el almacenamiento rápido del sistema operativo, los modelos y los datos procesados.
  + **RAM:** Memoria de Acceso Aleatorio, donde se cargan los datos y las instrucciones que la CPU necesita para realizar sus tareas, permitiendo un acceso rápido.
  + **GPU-INTEL:** Unidad de Procesamiento Gráfico integrada de Intel, que optimiza el rendimiento.
  + **RED-WIFI:** Indica la conexión de red inalámbrica que permite la comunicación con el celular.

CELULAR

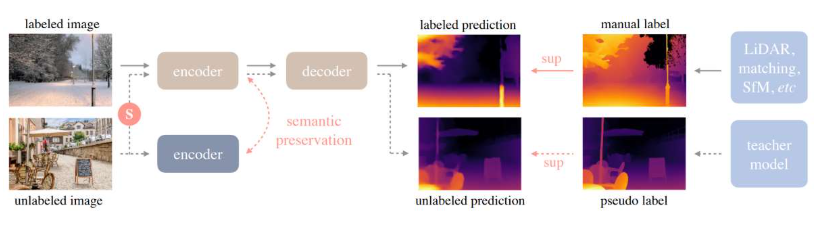
* + **CAMARA:** Utilizada para capturar la imagen del alimento que se enviará al servidor para su análisis.
  + **CPU:** Unidad Central de Procesamiento del celular, encargada de ejecutar la aplicación cliente, capturar la imagen y mostrar los resultados recibidos del servidor.
  + **RAM:** Memoria de Acceso Aleatorio del celular, necesaria para el funcionamiento de la aplicación cliente.
  + **RED-WIFI:** Indica la conexión de red inalámbrica que permite la comunicación con el servidor para enviar la foto y recibir las calorías y la receta.

### Esquema y descripción de modelos o componentes inteligentes (esquemas y/o graficas).

**Yolo v8:**

****

**Depth Anything:**

****

### Valores de parámetros e hiper parámetros aplicados.

**Parámetros entrenamiento de yolo-segmv8n:**

* **Parámetros de Capas Convolucionales (Conv):**
* **Capa 0 (Conv): 464 parámetros. (Capa convolucional inicial)**
* **Capa 1 (Conv): 4,672 parámetros. (Capa convolucional con más filtros)**
* **Capa 3 (Conv): 18,560 parámetros. (Capa convolucional con reducción de tamaño y más filtros)**
* **Capa 5 (Conv): 73,984 parámetros. (Capa convolucional más profunda con aún más filtros)**
* **Capa 7 (Conv): 295,424 parámetros. (Capa convolucional profunda con gran capacidad de aprendizaje)**
* **Parámetros de Bloques C2f (varían en número de parámetros):**
* **Capa 2 (C2f): 7,360 parámetros. (Bloque C2f inicial)**
* **Capa 4 (C2f): 49,664 parámetros. (Bloque C2f más profundo con mayor complejidad)**
* **Capa 6 (C2f): 197,632 parámetros. (Bloque C2f con una capacidad de aprendizaje significativa)**
* **Otras Capas Relevantes:**
* **Capa 9 (SPPF): 164,608 parámetros. (Capa de Spatial Pyramid Pooling - Fast, agrega información a diferentes escalas)**
* **Capa 22 (Segment - Capa Final de Segmentación): 15 parámetros. (La cabeza de segmentación, responsable de predecir las máscaras de los objetos)**

**Híper parámetros:**

* **epochs: 10 (Número total de pasadas por el dataset de entrenamiento)**
* **lr0 (tasa de aprendizaje inicial): 0.01 (aunque el optimizador automático ajustó este valor)**
* **batch: 16 (Tamaño del lote de imágenes por iteración)**
* **imgsz: 640 (Tamaño de las imágenes de entrada)**
* **optimizer: AdamW (Algoritmo de optimización utilizado)**
* **weight\_decay: 0.0005 (Factor de regularización para evitar el sobreajuste)**
* **iou (umbral de IoU para NMS): 0.7 (Umbral para la supresión no máxima durante la evaluación)**
* **pretrained: True (Indica que se utilizaron pesos pre-entrenados del modelo)**con activación Softmax.

### Especificaciones técnicas.

El desarrollo del proyecto se sustentó en librerías especializadas en visión por computadora y mapeo monocular de imágenes:

**Librerías de preprocesamiento y visualización:**

* **OpenC** **PIL (Pillow):** Biblioteca fundamental para la manipulación básica de imágenes en Python, como cargar, guardar y realizar conversiones de formato.
* **OpenCV (cv2):** Biblioteca integral para visión por computadora, utilizada para leer imágenes, realizar transformaciones de color y posiblemente implementar algunas operaciones de aumento.
* **Albumentations:** Librería rápida y flexible para el aumento de datos de imágenes, aplicada para generar variaciones de las imágenes de entrenamiento y mejorar la robustez del modelo.
* **TorchVision:** Parte de PyTorch, proporciona herramientas para la carga de datasets y transformaciones de imágenes que se integran con los tensores de PyTorch, facilitando elpreprocesamiento final para el entrenamiento.
* **NumPy:** Biblioteca para computación numérica en Python, fundamental para la manipulación de arrays multidimensionales que representan las imágenes y los tensores.
* **Roboflow:** Plataforma para la gestión, aumento y pre procesamiento de datasets de visión por computadora.
* **Ultralytics: La librería principal de YOLOv8; typing (Sequence):** Se usa para anotar colecciones de datos de entrada y garantizar tipado estático en funciones de preprocesamiento.
* **TensorFlow/Keras (Sequential**
* **Flask:** Microframework web para construir la API.
* **Flask-CORS:** Extensión para Flask que habilita el acceso cross-origin.
* **torch**: Framework de aprendizaje automático (PyTorch).
* **torchvision.transforms:** Módulo de PyTorch para transformaciones de imágenes.
* **io**: Módulo para trabajar con flujos de datos.
* **transformers**: Biblioteca de Hugging Face con modelos pre-entrenados.

### Lenguajes de programación, entre otros.

**Lenguaje de Programación:**

* Python es el lenguaje principal utilizado en el desarrollo del proyecto.
* Flutter es un marco de desarrollo de código abierto de Google que utiliza el lenguaje de programación Dart para crear aplicaciones móviles con interfaces de usuario atractivas y fluidas.

**Entorno de Desarrollo y ejecución:**

* **IDE:** Visual Studio Code.
* **Entorno virtual:** Se creo uno especifico para trabajar con librerías y dependencias de visión por computadora **IA\_env (Python 3.12.9)**
* **Hardware de entrenamiento:** colab con gpu t4.
* **Dispositivo de inferencia:** Android de gama media inifnix hot-30.

## Plan de trabajo

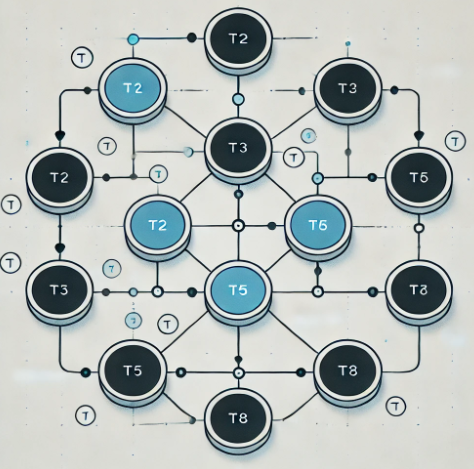
* **Recolección y armado de dataset de alimentos: Recolectar** la cantidad necesaria de imágenes, para procesarlas en roboflow, clasificando y segmentado cada imagen según su clase, exportando así un dataset segmentado, clasificado, con aumento de datos y datos de entrenamiento, validación y prueba.
* **Entrenamiento del modelo yolo:** Configurar el modelo de yolo para aprender solo alimentos que necesitamos que aprenda, con el dataset pre procesado.
* **Realización de pruebas al modelo entrenado:** Evaluar el modelo entrenado mediante pruebas exhaustivas, verificando su capacidad para identificar con precisión los alimentos.
* **Realización de estimación de calorías, tamaño y porciones:** mediante el modelo Depth Anything, información almacenada de calorías y el modelo entrenado de yolo realizamos el cálculo de tamaño de los alimentos reconocidos en porciones para el cálculo de sus calorías.
* **Generación de receta:** Mediante los alimentos reconocidos mas su porción y su estado calórico se generara una receta con gpt-3.
* **Desarrollo de la aplicación móvil para Android:** Desarrollar una interfaz de usuario en el entorno de Visual Studio Code (VSCode) que sea compatible con dispositivos Android, permitiendo a los usuarios interactuar con la aplicación.
* **Realización de pruebas en la APP desarrollada**: Ejecutar pruebas en la aplicación móvil completa, asegurándose de que funcione correctamente al identificar la posición correcta de la cámara para tomar la foto, encontrar alimentos más sus calorías para generar una receta.

## Cronograma

### Definición de tareas

|  |  |  |
| --- | --- | --- |
| **Fase** | **Duración** | **Fechas** |
| **T-1: Análisis e investigación** | **1 semanas** | **10 - 17 de marzo** |
| **T-2: Selección de herramientas y recopilación de datos(dataset)** | **1 semana** | **18 - 24 de marzo** |
| **T-3: Entrenamiento del modelo** | **3 semanas** | **25 – 14 de abril** |
| **T-4: Desarrollo de la aplicación Móvil** | **3 semanas** | **15 de abril – 5 de mayo** |
| **T-5: Integración de bastón, componentes(modelos y base de datos )** | **2 semanas** | **6 de abril – 19 de mayo** |
| **T-6: Pruebas y Validación** | **2 semanas** | **20 de mayo – 2 de junio** |
| **T-7: Presentación y Documentación** | **2 semanas** | **3 de junio - 16 de junio** |

### Diagrama de Pert

****

### Diagrama de Gantt

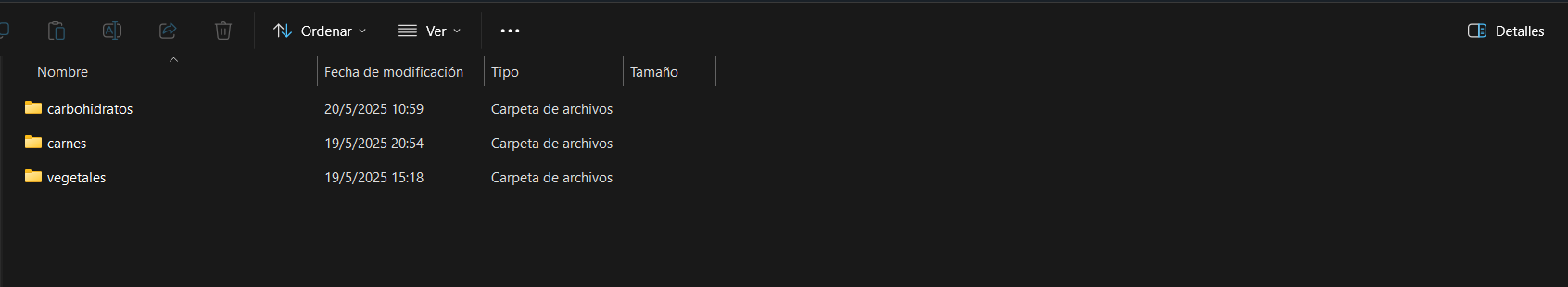
****

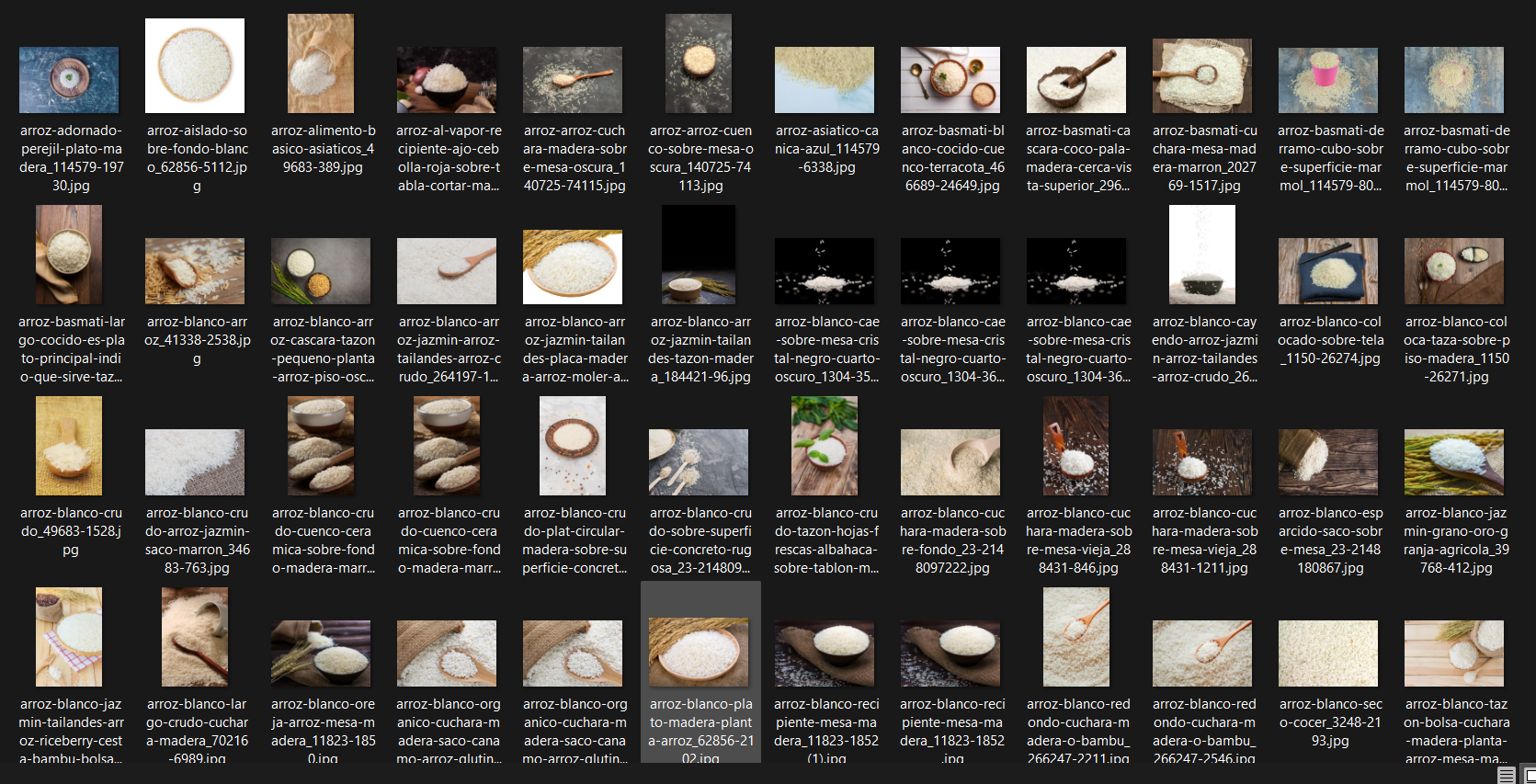
# Resultados

## Dataset

### Descripción y preprocesamiento realizado (Evidencia del antes y después)

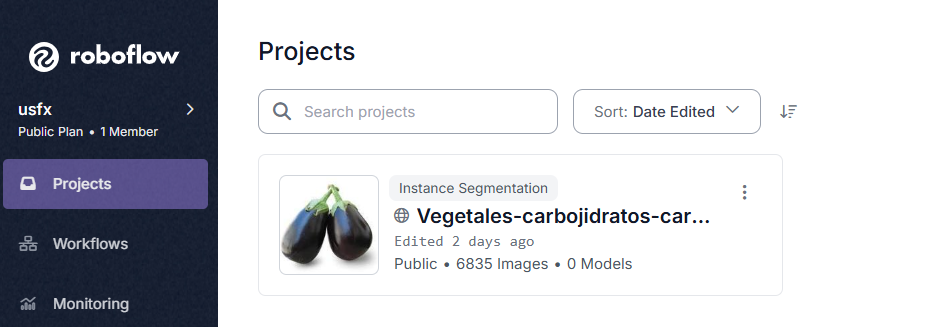
El preprocesamiento de los datos se llevó a cabo en varias etapas. Se recolectaron imágenes de alimentos de datasets, internet y en entornos reales y se almacenaron en carpetas.

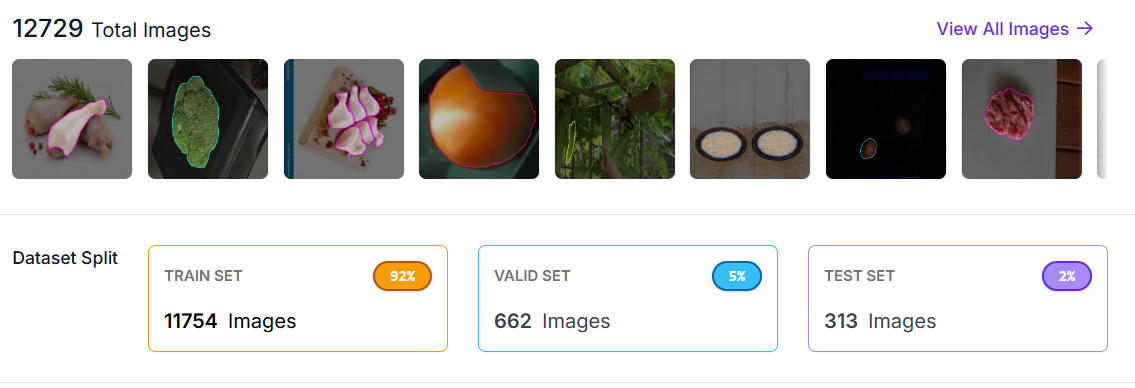




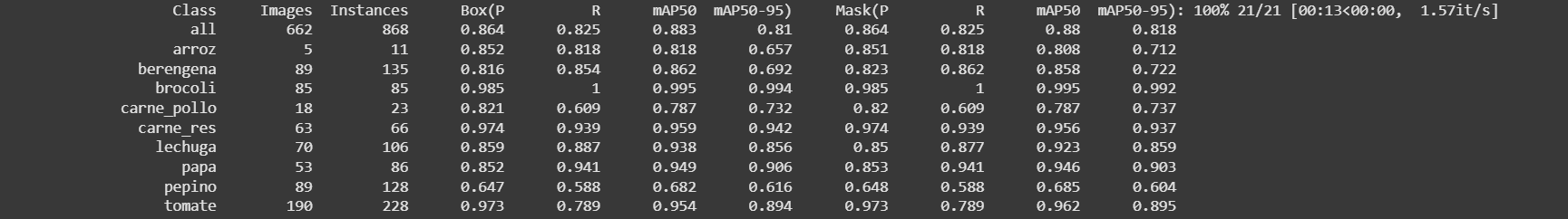


Posteriormente se subieron a roboflow para marcación y clasificación





Luego se entreno el modelo yolo para reconocimiento de alimentos

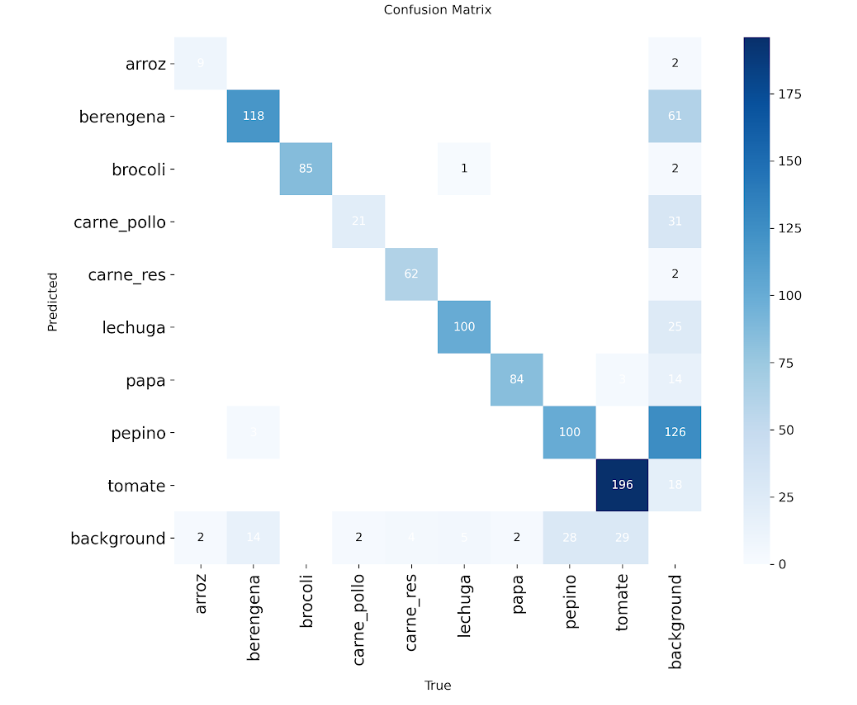


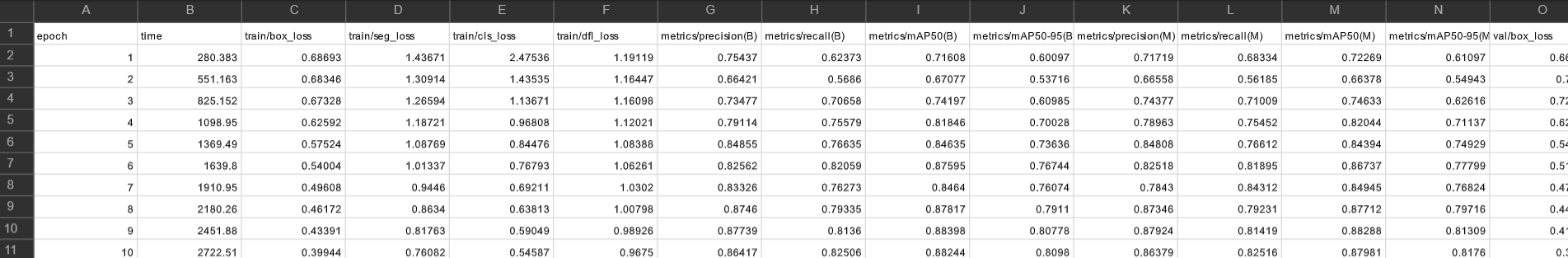
### Conjunto de entrenamiento, evaluación y validación

### 

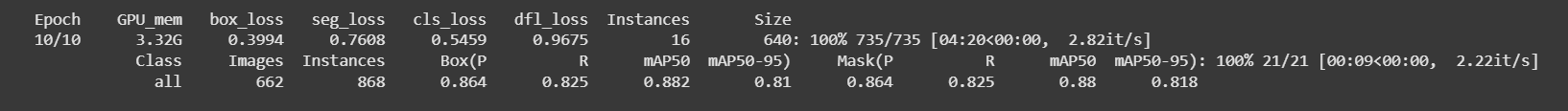
## Resultados de entrenamiento y prueba (Métricas de rendimiento)

### Matriz de Confusión

****

****

### Precisión



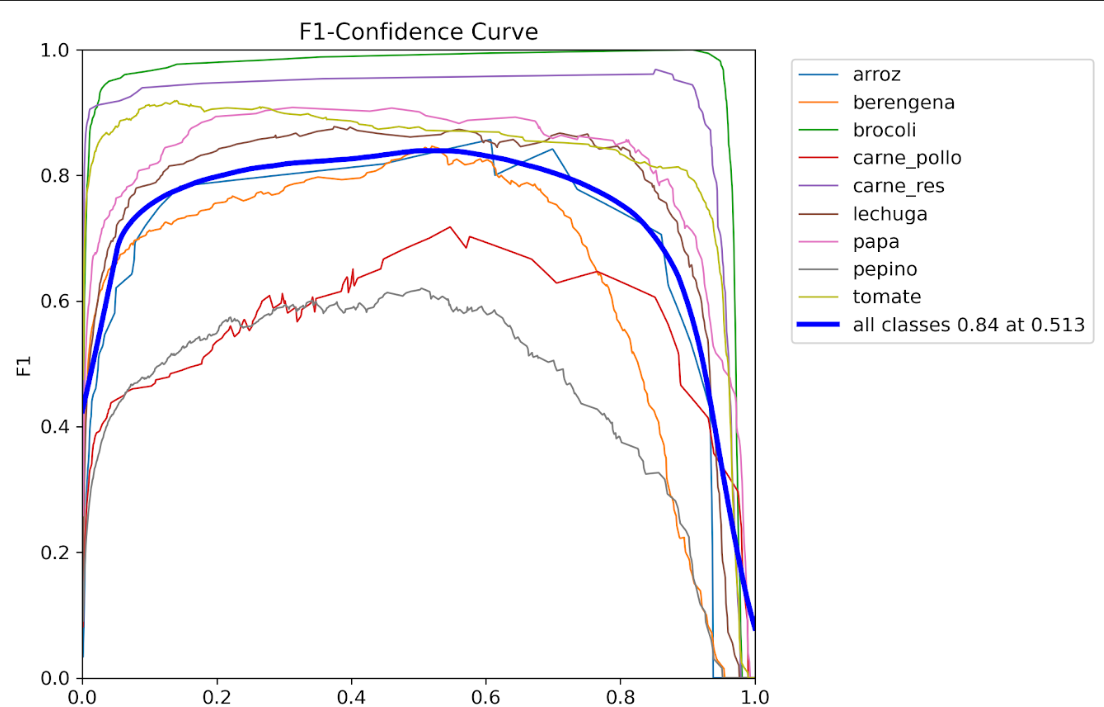
**La presicion del modelo alcanzo un 86 % de precision**

### Sensibilidad (Recall)

### 

### F-Score

**El F1-score** es la media armónica de precisión y recall. Es útil cuando se desea tener en cuenta tanto la precisión como el recall en una sola métrica. Cuanto más alto sea el F1-score, mejor será el equilibrio entre precisión y recall.

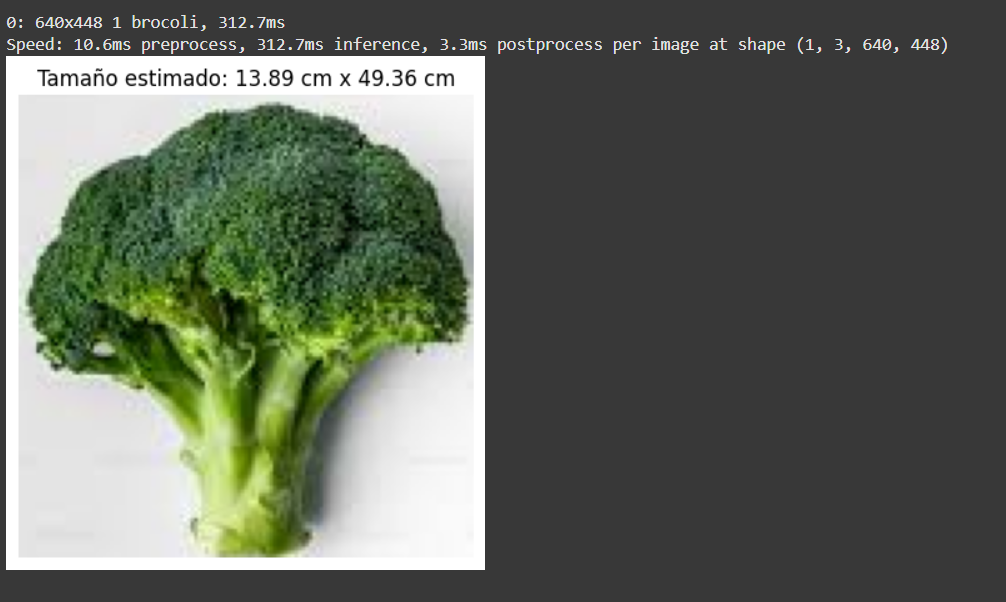


## Resultados de aplicación y utilización

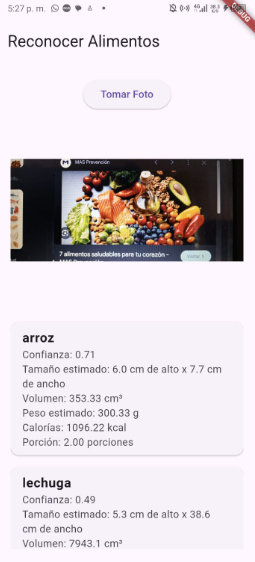
El modelo entrenado de yolov8 llego a un 84 % de precisión



El modelo depth Anything tiene un variación del 20 % el estimar los tamaños exactos de los alimentos reconocidos

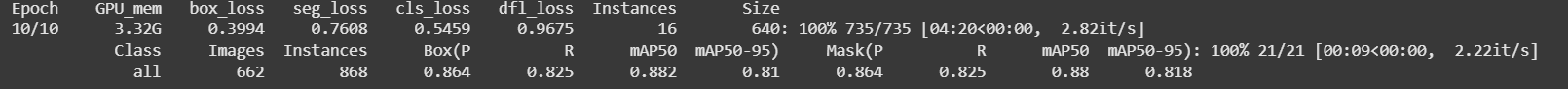


Datos con todo integrado en la aplicación móvil:

## Coincidencia de los resultados con los objetivos planteados.

El objetivo planteado era llegar a un 90 % en el reconocimiento de alimentos, pero el modelo entrenado solo alcanzo un 86% en el reconocimiento de alimentos



## Conclusiones

Viabilidad técnica y funcional del sistema: El desarrollo de un software que permite identificar alimentos mediante visión por computadora, estimar sus calorías utilizando modelos de profundidad, y generar recetas con modelos de lenguaje como GPT-3, ha demostrado ser técnicamente viable. Los resultados obtenidos indican una buena precisión en la detección de alimentos y en la generación de recetas relevantes.

Precisión del modelo de detección: El modelo YOLO-segmv8n entrenado alcanzó una precisión del 86%, lo cual es satisfactorio para una primera versión y demuestra un desempeño robusto en la identificación de los tres grupos alimenticios (carnes, vegetales y carbohidratos).

Utilidad en contextos reales: La aplicación móvil desarrollada con Flutter permite al usuario interactuar de forma intuitiva, tomar una fotografía de los alimentos, y recibir una receta personalizada, lo que puede contribuir significativamente a mejorar los hábitos alimenticios de la población.

Impacto social y potencial de escalabilidad: Este tipo de soluciones puede ser especialmente útil en contextos urbanos donde las personas tienen poco conocimiento nutricional. Además, su escalabilidad permite que, con ajustes y ampliaciones, pueda adaptarse a dietas específicas (diabéticos, veganos, etc.).

Integración exitosa de múltiples modelos de IA: La combinación de modelos de visión por computadora, estimación de profundidad y generación de texto permitió crear un sistema inteligente integral que demuestra la potencia del enfoque interdisciplinario en la inteligencia artificial.

## Recomendaciones

Ampliar el conjunto de datos: Se recomienda aumentar y diversificar el dataset de entrenamiento con más imágenes de alimentos locales en distintas condiciones de iluminación y ángulos, para mejorar la generalización del modelo.

Implementar mejoras en la interfaz móvil: Incorporar sugerencias nutricionales adicionales, filtros por necesidades alimenticias y un historial de recetas generadas, puede aumentar la utilidad de la aplicación para el usuario.

Optimización del modelo para dispositivos móviles: Aunque se obtuvo buen rendimiento, es recomendable explorar técnicas de cuantización o poda de modelos para mejorar aún más el tiempo de inferencia en dispositivos de gama baja.

## Bibliografía

* [**https://www.tuasaude.com/es/calorias-de-los-alimentos/**](https://www.tuasaude.com/es/calorias-de-los-alimentos/)
* [**https://www.runnersworld.com/es/nutricion-deportiva/a30238564/foodvisor-aplicacion-contar-calorias-perder-peso/**](https://www.runnersworld.com/es/nutricion-deportiva/a30238564/foodvisor-aplicacion-contar-calorias-perder-peso/)
* [**https://as.com/deporteyvida/2017/08/21/portada/1503314306\_822170.html**](https://as.com/deporteyvida/2017/08/21/portada/1503314306_822170.html)