

The Fridge Survival Guide Project

Documento de ruta provisional:

Paso 1: Configuración Inicial del Entorno en GCP

1.1. Habilitar APIs y Acceder a Vertex AI

1. **Habilita las APIs necesarias:**

- Ve a [Google Cloud Console](#).
- En el menú de navegación, selecciona **APIs & Services > Library**.
- Busca y habilita:
 - **Vertex AI API**.
 - **Cloud Storage API**.

2. **Abre Vertex AI:**

- En el menú de navegación, selecciona **Vertex AI > Dashboard**.

3. **Crea un bucket de Cloud Storage:**

```
gsutil mb -p $GOOGLE_CLOUD_PROJECT -c standard -l us-central1  
gs://$GOOGLE_CLOUD_PROJECT-fridge/
```

- Reemplaza **\$GOOGLE_CLOUD_PROJECT** con tu ID de proyecto.
-

Paso 2: Preparar y Subir Datasets

2.1. Dataset de Imágenes (Ingredientes)

1. **Descarga imágenes de ejemplo** (usaremos 3 ingredientes: tomate, huevo, cebolla):

```
gsutil -m cp -r gs://splis/gsp223/images/*  
gs://$GOOGLE_CLOUD_PROJECT-fridge/
```

- *Nota:* Usamos el dataset de nubes como ejemplo, pero en tu caso, reemplázalo con imágenes de ingredientes organizadas en carpetas por clase (ej: **tomato/**, **egg/**, **onion/**).

2. **Crea un archivo CSV** para el dataset:

- Descarga el template:

```
gsutil cp gs://spl/spls/gsp223/data.csv .
```

Edita el CSV para que apunte a tus imágenes (ejemplo para ingredientes):

```
image_path,label  
gs://tu-bucket/fridge/tomato/tomato1.jpg,tomato  
gs://tu-bucket/fridge/egg/egg1.jpg,egg
```

- `gs://tu-bucket/fridge/onion/onion1.jpg,onion`
- Sube el CSV a tu bucket:
`gsutil cp data.csv gs://$GOOGLE_CLOUD_PROJECT-fridge/`

2.2. Dataset de Recetas (Tabla para AutoML Tables)

Crea un CSV de recetas (`recipes.csv`):

```
name,ingredients  
Tortilla de patatas,"egg, potato, onion"  
Ensalada,"lettuce, tomato"
```

1. `Pancakes,"flour, milk, egg"`

2. Sube el CSV a GCS:

```
gsutil cp recipes.csv gs://$GOOGLE_CLOUD_PROJECT-fridge/
```

3. Importa el CSV a BigQuery:

```
bq load --autodetect --source_format=CSV recipes.recipes  
gs://$GOOGLE_CLOUD_PROJECT-fridge/recipes.csv
```

Paso 3: Crear Datasets en Vertex AI

3.1. Dataset de Imágenes (AutoML Vision)

1. Ve a Vertex AI > Datasets.
2. Crea un nuevo dataset:
 - Nombre: `fridge_images`.
 - Tipo de datos: Image.
 - Objetivo: Single-label classification.

- **Región:** `us-central1`.
 - 3. **Importa datos:**
 - Selecciona el archivo CSV desde tu bucket (`data.csv`).
 - Espera a que se complete la importación (2-5 min).
-

3.2. Dataset de Recetas (AutoML Tables)

1. **Ve a Vertex AI > Datasets.**
 2. **Crea un nuevo dataset:**
 - **Nombre:** `fridge_recipes`.
 - **Tipo de datos:** Tabular.
 - **Fuente:** BigQuery (`recipes.recipes`).
 - **Objetivo:** Predecir la columna `name` (receta).
-

Paso 4: Entrenar Modelos

4.1. Modelo de Visión (AutoML Vision)

1. **Entrena el modelo:**
 - Ve a tu dataset `fridge_images`.
 - Haz clic en **Train new model**.
 - Configura:
 - **Budget:** 1 node hour (para pruebas rápidas).
 - **Modelo:** MobileNetV2 (optimizado para CPU).
 - Inicia el entrenamiento (tarda ~1 hora).
 2. **Despliega el modelo:**
 - Una vez entrenado, desplégalo como un **endpoint** en Vertex AI.
-

4.2. Modelo de Recomendación (AutoML Tables)

1. **Entrena el modelo:**
 - Ve a tu dataset `fridge_recipes`.
 - Haz clic en **Train new model**.
 - Configura:
 - **Budget:** 1 node hour.
 - **Target column:** `name`.
 - Inicia el entrenamiento (tarda ~30 min).

2. Despliega el modelo:

- Despliega el modelo como un endpoint.

Paso 5: Generar Predicciones

5.1. Predicción con el Modelo de Visión

1. Prepara una imagen de prueba:

```
gsutil cp gs://spl/gsp223/examples/* .
```

- *Nota:* Usa una imagen de un ingrediente (ej: `tomato.jpg`).

Haz una predicción:

```
curl -X POST \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
  -H "Content-Type: application/json" \
  "https://us-central1-aiplatform.googleapis.com/v1/projects/${GOOGLE_CLOUD_PROJECT}/locations/us-central1/endpoints/TU_ENDPOINT_ID:predict" \
  -d "@prediction_request.json" \
```

```
2. | jq > prediction.json
```

- *Nota:* `prediction_request.json` debe contener la imagen en formato base64.

5.2. Predicción con el Modelo de Recomendación

Envía los ingredientes detectados al endpoint de AutoML Tables:

```
from google.cloud import aiplatform

endpoint =
aiplatform.Endpoint("projects/TU_PROYECTO0/locations/us-central1/endpoints/TU_ENDPOINT_ID")

instance = {"ingredients": "egg, tomato, onion"}
result = endpoint.predict(instances=[instance])
```

```
1. print("Recetas recomendadas:", result.predictions)
```

Paso 6: Integración con Streamlit (Interfaz Web)

Crea un archivo **app.py**:

```
import streamlit as st
from google.cloud import aiplatform
from PIL import Image
import base64
import json
import requests

# Configuración de endpoints
VISION_ENDPOINT =
"projects/TU_PROYECTO/locations/us-central1/endpoints/TU_VISION_ENDP
OINT"
RECOMMENDER_ENDPOINT =
"projects/TU_PROYECTO/locations/us-central1/endpoints/TU_RECOMMENDER
_ENDPOINT"

st.title("Fridge Survival Guide")

uploaded_file = st.file_uploader("Sube una foto de tu frigorífico",
type=["jpg", "png"])

if uploaded_file:
    image = Image.open(uploaded_file)
    st.image(image, caption="Imagen subida", use_column_width=True)

    # Convierte la imagen a base64
    buffered = io.BytesIO()
    image.save(buffered, format="JPEG")
    img_str = base64.b64encode(buffered.getvalue()).decode("utf-8")

    # Predicción con AutoML Vision
    vision_client = aiplatform.Endpoint(VISION_ENDPOINT)
    vision_result = vision_client.predict(instances=[{"content":
img_str}])
```

```

    detected_ingredients = [pred["displayNames"] for pred in
vision_result.predictions[0]]
    st.write("Ingredientes detectados:", detected_ingredients)

# Predicción con AutoML Tables
    recommender_client = aiplatform.Endpoint(RECOMMENDER_ENDPOINT)
    instance = {"ingredients": ",".join(detected_ingredients)}
    recommender_result =
recommender_client.predict(instances=[instance])

1.     st.write("Recetas recomendadas:",
    recommender_result.predictions)

```

Despliega en Cloud Run:

```

gcloud run deploy fridge-app \
  --source . \
  --port 8501 \
  --memory 4Gi \
  --cpu 2 \
  --region us-central1 \

2.     --allow-unauthenticated

```

Paso 7: Alternativas si no Tienes Recursos en GCP

7.1. Usar Modelos Preentrenados (Local)

1. Modelo de visión:

Usa **MobileNetV2** con TensorFlow Hub:

```

import tensorflow as tf
import tensorflow_hub as hub

    model =
        hub.load("https://tfhub.dev/google/imagenet/mobilenet_v2_
100_224/classification/4")

```

2. Recomendador local:

Usa **scikit-learn** para similitud de coseno:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

recipes = ["Tortilla: egg, potato, onion", "Ensalada: lettuce,
tomato"]
vectorizer = TfidfVectorizer()

    ○ X = vectorizer.fit_transform(recipes)
```

Resumen de Tiempos

Día	Tarea
1-2	Configuración inicial y preparación de datasets.
2-4	Entrenamiento de modelos en Vertex AI.
5-6	Integración con Streamlit y despliegue en Cloud Run.
