

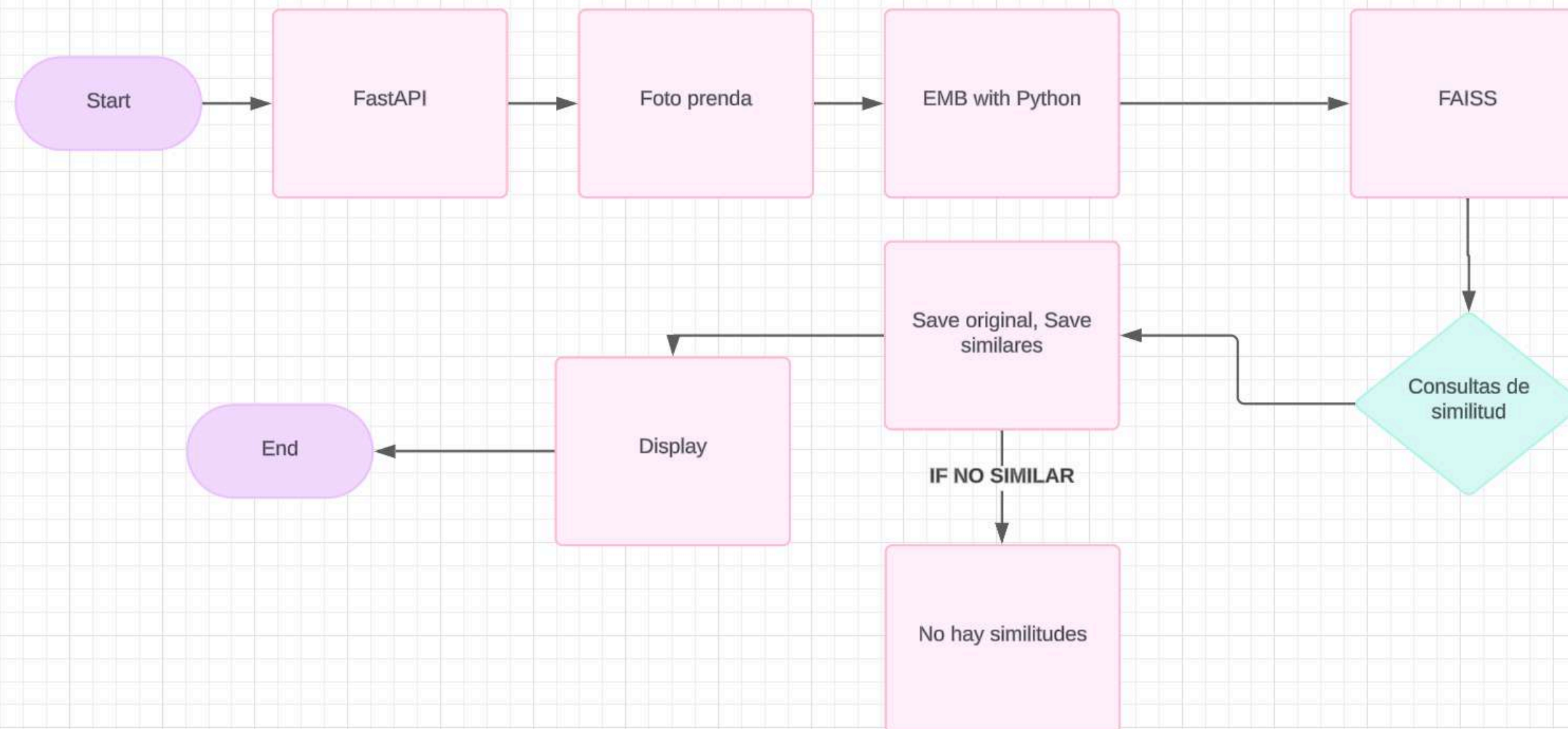
PROPUESTA PROYECTO

APLICACIÓN MÓVIL

SARAH CÁRDENAS MONTIEL
SEBASTIÁN MÁRQUEZ
IVÁN HERNÁNDEZ ROSALES
CAROLINA PÉREZ VALENCIA
DAVID DÍAZ AVILÉS



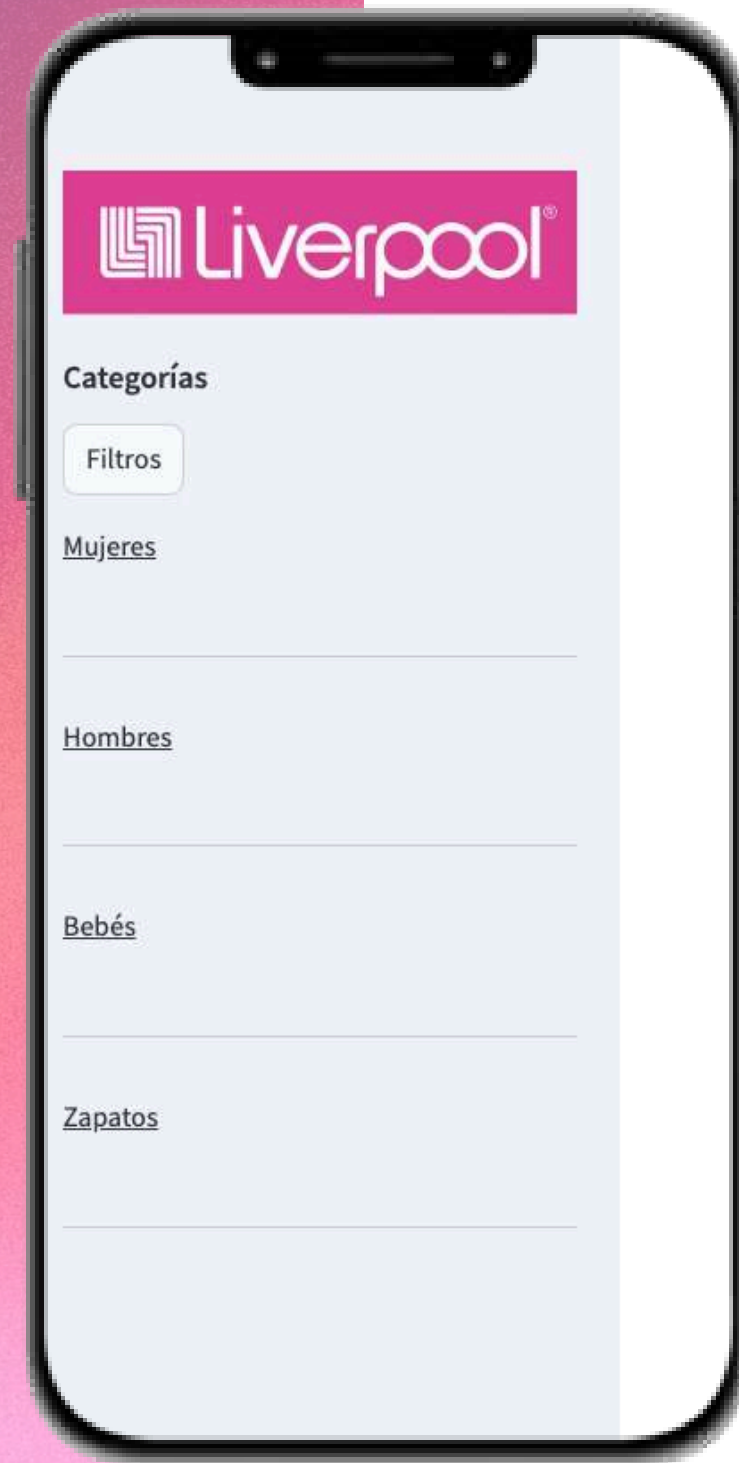
DIAGRAMA INICIAL



INTRODUCCIÓN

APP MÓVIL

Desarrollo de una app en la que podemos usar la cámara del dispositivo, con el objetivo de explorar prendas similares a la fotografiada.



OBJETIVO

Entrenar a la IA para la identificación de imágenes en las que podamos encontrar similitudes con las fotos presentadas, para convertir las imágenes en embeddings.



PROCEDIMIENTO

Configuración del Modelo de Embeddings

Entrenar a la IA para la identificación de imágenes en las que podamos encontrar similitudes con las fotos presentadas, para convertir las imágenes en embeddings. Esto con el propósito de crear una función para generar el embeddings a partir de una imagen.

Creación y Gestión del Índice FAISS

Configurar un índice FAISS que almacene los embeddings del catálogo y permita realizar búsquedas rápidas de similitud.

Con la intención de implementar una función de búsqueda de similitud para encontrar los embeddings más cercanos a un vector dado.

Desarrollo de la App:

Desarrollar una Appl que permita:

- Recibir imágenes desde la app para generar embeddings.
- Consultar el índice FAISS.
- Comparar con las imágenes filtradas la más similar

Mostrar casos similares al referenciado.

Lo hacemos mediante la entrega al sistema de la foto, comparar con la referencia y nos muestre las similitudes.

PROCEDIMIENTO

Integración con la App Móvil

Permitir que la app móvil capture una foto y la envíe al servidor para su procesamiento.

Con las tareas de recibir y desplegar los productos similares en la interfaz, implementar validaciones para manejar errores.

Pruebas y Optimización

Verificar la precisión y eficiencia del sistema.

Tareas:

- Probar el sistema con imágenes de prueba y evaluar la relevancia de los resultados.
- Optimizar el tiempo de respuesta y el procesamiento de imágenes.
- Implementar optimizaciones como índices aproximados, para mejorar la velocidad de búsqueda.

Desarrollo de la API con FastAPI

Proveer documentación clara para uso, despliegue y mantenimiento del sistema.

Tareas:

Documentar la estructura del proyecto y el flujo de trabajo.

Resultados preliminares

Proyectar mejoras a largo plazo para optimizar el sistema.

Ideas:

- Integrar la base de datos proporcionada por la misma empresa.
- Implementar un sistema de recomendación que facilite la similitud de imágenes.

PROGRAMA

PROGRAMA

```
import streamlit as st
from PIL import Image
import base64

# Configuración de la página
st.set_page_config(page_title="Mi Aplicación", layout="wide")

# Ruta de la imagen de encabezado
img_path_header = "/Users/sarahcardenas/Desktop/Liverpool1.jpg"

# Cargar imagen de encabezado y codificarla en base64
with open(img_path_header, "rb") as img_file:
    img_base64_header = base64.b64encode(img_file.read()).decode()

# Estilos de encabezado con fondo rosa y logotipo a la izquierda
st.markdown(f"""
<style>
    .top-header {{
        background-color: #ff007f;
        padding: 15px;
        display: flex;
        align-items: center;
        justify-content: space-between;
    }}
    .top-header img {{
        width: 100px;
        height: 50px;
    }}
</style>
<div class="top-header">
    <img alt="Liverpool logo" data-bbox="100 100 200 200"/>
    <h1>PROGRAMA</h1>
</div>
""")
```


PROGRAMA

```
# Espaciado superior
st.markdown("<br><br>", unsafe_allow_html=True)

# Botones
left, middle, right = st.columns(3)

# Cámara
if right.button("Búsqueda con cámara", key="camera_search", use_container_width=True):
    right.markdown("Toma la fotografía")

# Checkbox para habilitar la cámara
enable = st.checkbox("Enable camera", value=True) # Se habilita por defecto
```

```
# Mostrar la cámara solo si se habilita
if enable:
    # Capturar la imagen
    picture = st.camera_input("Take a picture")

    # Guardar la imagen en el estado de sesión
    if picture:
        st.session_state['picture'] = picture

# Mostrar la imagen si existe en el estado de sesión
if 'picture' in st.session_state:
    st.image(st.session_state['picture'])
```


PROGRAMA

```
if left.button("Búsqueda manual", key="manual_search", use_container_width=True):  
    left.markdown("Describe el producto que deseas buscar ")  
  
if middle.button("Búsqueda por selección de imagen", key="image_selection", use_container_width=True):  
    middle.markdown("Sube una imagen.")  
  
# Cargador de archivos  
uploaded_file = st.file_uploader("", type=["csv", "txt"])
```


PROGRAMA

```
# Espaciado inferior
st.markdown("<br><br>", unsafe_allow_html=True)

# Ruta de la imagen de publicidad
img_path_pub = "/Users/sarahcardenas/Desktop/publicidad.png"

# Cargar imagen de publicidad y mostrarla ocupando ancho
try:
    with open(img_path_pub, "rb") as img_file_pub:
        img_pub = Image.open(img_file_pub)
        st.image(img_pub, use_column_width=True, caption="Publicidad")
except Exception as e:
    st.error(f"No se pudo cargar la imagen de publicidad: {e}")
```


PROGRAMA

```
# Estilo para el título
st.markdown(
    """
    <style>
        .center-text {
            text-align: center;
            font-size: 24px; /* Cambia el tamaño según lo desees */
            font-weight: bold; /* Puedes hacer el texto en negrita si lo prefieres */
        }
    </style>
    """,
    unsafe_allow_html=True
)

# Lista lateral
img_path = "/Users/sarahcardenas/Desktop/Liverpool11.jpg"
width = 400

# Cargar y codificar la imagen en base64
with open(img_path, "rb") as img_file:
    img_base64 = base64.b64encode(img_file.read()).decode()
```

```
# Estilos CSS para la barra lateral
st.markdown("""
    <style>
        .sidebar-content {
            color: black;
        }
    </style>
    """, unsafe_allow_html=True)

# Contenido de la barra lateral
st.sidebar.image(f"data:image/jpeg;base64,{img_base64}", use_column_width=True)

# Mostrar las categorías como una lista con divisores
st.sidebar.subheader("Categorías")

# Botón en la barra lateral
if st.sidebar.button("Filtros"):
    st.sidebar.write("Filtros activados.") # Acción que se realiza al presionar el botón

categorias = ["Mujeres", "Hombres", "Bebés", "Zapatos"]

for categoria in categorias:
    st.sidebar.markdown(f"<u>{categoria}</u>", unsafe_allow_html=True)
    st.sidebar.markdown("---") # Línea divisoria entre categorías
```


ENTREGA FINAL



Categorías

Filtros

Mujeres

Hombres

Bebés

Zapatos



Mis Compras Iniciar sesión

Búsqueda manual

Búsqueda por selección de imagen

Búsqueda con cámara





Categorías

Filtros

Mujeres

Hombres

Bebés

Zapatos



Mis Compras Iniciar sesión

Búsqueda manual

Búsqueda por selección de imagen

Búsqueda con cámara

Toma la fotografía

☒ Enable camera

Take a picture



This app would like to use your camera.

[Learn how to allow access.](#)

