

# Universidad Veracruzana



Académico: López Hernández Jesús Leonardo

Experiencia educativa: Inteligencia Artificial

Licenciatura en Tecnologías de la Información en las  
Organizaciones

Presenta:

**Rafael Espíritu Arias**



# Índice

## Contenido

Visión general del sistema.....	3
Librerías necesarias.....	3
1. entrenar_prefijos.py – Modelo de predicción por prefijos .....	4
2. entrenar_trigramas.py – Modelo de predicción por contexto (trigramas) .....	4
3. teclado_mano_avanzado.py – Interfaz visual, gestos y sugerencias.....	5
A. Carga de recursos y modelos .....	5
B. Inicializa MediaPipe Hands .....	5
C. Dibujar el teclado virtual.....	6
D. Detección de punta del dedo .....	6
E. Detección de gesto para borrar (mano abierta) .....	6
F. Detección de tecla tocada.....	7
G. Predicción y sugerencias .....	7
H. Pantalla completa y resolución.....	7
Resumen general del flujo .....	8



## Visión general del sistema

Este proyecto consta de tres scripts de Python que trabajan en conjunto:

1. **entrenar\_prefijos.py** – Entrena un modelo para completar palabras usando prefijos.
2. **entrenar\_trigramas.py** – Entrena un modelo para predecir palabras usando trigramas (contexto de dos palabras).
3. **teclado\_mano\_avanzado.py** – Crea el teclado virtual que:
  - Detecta la punta del dedo para navegación.
  - Selecciona una tecla con un gesto de "pellizco".
  - Borra el texto cuando detecta una mano abierta.
  - Reproduce un sonido al escribir.
  - Sugiere palabras según lo escrito.

## Librerías necesarias

Para ejecutar el proyecto correctamente, se requieren las siguientes librerías:

```
pip install opencv-python mediapipe pygame
```

Descripción de cada librería:

LIBRERÍA	USO PRINCIPAL
<b>CV2</b>	Mostrar video en vivo, dibujar el teclado y detectar la posición de los dedos.
<b>MEDIPIPE</b>	Detección precisa de manos y puntos de referencia (landmarks).
<b>PYGAME</b>	Reproducción de sonidos cuando se presiona una tecla.
<b>PICKLE</b>	Guardar y cargar modelos de predicción en archivos .pkl.
<b>COLLECTIONS.DEFAULTDICT</b>	Estructura eficiente para almacenar modelos de predicción.



## 1. **entrenar\_prefijos.py** – Modelo de predicción por prefijos

### Función:

Este script genera un modelo que **asocia prefijos con palabras completas** para sugerir posibles terminaciones.

### Cómo funciona:

1. **Lee el archivo de texto** (corpus.txt).
2. Convierte todas las palabras a mayúsculas.
3. Por cada palabra, **genera todos sus prefijos posibles**.
4. Cada prefijo se asocia a todas las palabras que lo comienzan.
5. Guarda el modelo en modelo\_prefijos.pkl usando pickle.

### Ejemplo:

Si tienes "GATO", se generan:

- G → GATO
- GA → GATO
- GAT → GATO
- GATO → GATO

## 2. **entrenar\_trigramas.py** – Modelo de predicción por contexto (trigramas)

### Función:

Este script entrena un modelo que **predice la próxima palabra** en función de las dos anteriores (como en predicción inteligente de texto).

### Cómo funciona:

1. **Lee línea por línea** del corpus.txt.
2. Divide en palabras y recorre el texto formando tríos de palabras consecutivas.
3. Por cada par de palabras (clave), guarda cuántas veces aparece una tercera palabra.
4. Guarda el modelo como modelo\_trigramas.pkl.
- 5.



**Ejemplo:**

Texto: "ME GUSTA EL CAFÉ"

- (ME, GUSTA) → EL
- (GUSTA, EL) → CAFÉ

### 3. teclado\_mano\_avanzado.py – Interfaz visual, gestos y sugerencias

**Función:**

Este es el script **principal** del sistema. Usa los modelos entrenados y controla todo el teclado virtual.

Secciones clave:

#### A. Carga de recursos y modelos

```
python

click_sound = pygame.mixer.Sound("click.wav")
with open("modelo_trigramas.pkl", "rb") as f:
    modelo_trigramas = pickle.load(f)
```

- Se carga el sonido del clic.
- Se carga el modelo entrenado de predicción por trigramas.

#### B. Inicializa MediaPipe Hands

```
python

hands = mp.solutions.hands.Hands(...)
```

Esto permite detectar y rastrear la mano en tiempo real. Especialmente útil para saber **dónde está el dedo índice** y qué gestos se realizan.



### C. Dibujar el teclado virtual

```
python

# Filas del teclado
filas = [
    list("QWERTYUIOP"),
    list("ASDFGHJKL"),
    list("ZXCVBNM"),
    ["ESPACIO", "Borrar"] + [str(i) for i in range(10)]
]
```

1. Crea un diseño completo de teclado.
2. Calcula la posición de cada tecla para que estén **centradas** y distribuidas uniformemente.
3. El teclado es **negro** con letras blancas, lo cual mejora la visibilidad.

### D. Detección de punta del dedo

```
python

def detectar_punta_index(...):
    x = int(hand_landmarks.landmark[8].x * width)
    y = int(hand_landmarks.landmark[8].y * height)
```

Se usa el landmark **número 8** que representa la punta del dedo índice.

### E. Detección de gesto para borrar (mano abierta)

```
python

def detectar_gesto_palma_abierta(landmarks):
    dedos_abiertos = [...]
    pulgar_izq = ...
    pulgar_der = ...
    return all(dedos_abiertos) and (pulgar_izq or pulgar_der)
```



Si todos los dedos están estirados y el pulgar también, se detecta **mano abierta** y se borra todo el texto.

## F. Detección de tecla tocada

El código comprueba si la punta del dedo está sobre una tecla por más de 1 segundo:

```
python

if tx < x < tx + tecla_size and ty < y < ty + tecla_size:
    if ultima_tecla != tecla:
        ultima_tecla = tecla
        tiempo_inicio = time.time()
    elif time.time() - tiempo_inicio > 1:
        ...
...
```

1. Si el usuario se queda apuntando a una tecla por más de 1 segundo, se activa la tecla.
2. Se reproduce el sonido.
3. Si es "Borrar", elimina una letra.
4. Si es "Espacio", inserta un espacio.
5. Si es una letra o número, se añade al texto.

## G. Predicción y sugerencias

```
python

sugerencia = predecir_palabra(texto_escrito, modelo_trigramas)
```

1. Cada vez que se actualiza el texto, se consulta el modelo de trigramas para obtener hasta 3 sugerencias.
2. Estas aparecen en recuadros seleccionables debajo del texto.
3. Si el usuario selecciona una, reemplaza la última palabra con la sugerida.

## H. Pantalla completa y resolución

```
python

frame = cv2.resize(frame, (1920, 1080))
cv2.setWindowProperty("Teclado Mano Predictivo", cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
```

Esto asegura que el teclado se vea bien en pantallas grandes y quede **centrado** visualmente.



¿Cómo se comunican los archivos?

ARCHIVO	PRODUCE	CONSUMIDO POR
<b>ENTRENAR_PREFIJOS.PY</b>	modelo_prefijos.pkl	(no usado directamente en teclado_mano_avanzado.py)
<b>ENTRENAR_TRIGRAMAS.PY</b>	modelo_trigramas.pkl	<input checked="" type="checkbox"/> usado en teclado_mano_avanzado.py
<b>TECLADO_MANO_AVANZADO.PY</b>	—	Usa los modelos .pkl para sugerir palabras

Nota: Aunque el modelo de prefijos no está siendo usado activamente, podrías integrarlo para dar sugerencias mientras se escribe, incluso sin contexto.

## Resumen general del flujo

1. Entrenas modelos con corpus.txt.
2. Generas dos archivos .pkl (prefijos y trigramas).
3. Ejecutas teclado\_mano\_avanzado.py, el cual:
  - Usa la cámara para detectar tu mano.
  - Dibuja el teclado virtual en la pantalla.
  - Detecta tu dedo para seleccionar letras.
  - Borra texto si abres la mano.
  - Sugiere palabras a medida que escribes.
  - Emite un sonido cuando presionas una tecla.