

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú, Decana de América

Facultad de Ingeniería de Sistemas e Informática
E.P. de Ingeniería de Software



**INSTALACIÓN Y MANUAL DE USUARIO -
LABORATORIOS DE SOFTWARE INTELIGENTE**

DOCENTE:

CALDERÓN VILCA, DAVID

ALUMNOS:

CALLUPE HUAMAN, GERMÁN

DIAZ CHAMBI, LESLIE

LIMA – PERÚ

2025

1. Requisitos Previos	3
2. Instalación	3
2.1. Instalar Python	3
2.2. Instalar PyCharm	3
2.3. Entorno Virtual (.venv)	4
2.4. Cargar proyecto	4
2.5. Errores comunes	5
3. Estructura del proyecto	6
4. Descripción de módulos	7
4.1. Aplicación de 'Algoritmos Genéticos'	7
4.2. Aplicación de 'Naive Bayes'	8
4.3. Aplicación de 'Redes Neuronales'	8
4.4. Aplicación de 'Clustering'	8
4.5. Aplicación de 'Transfer Learning'	9
5. Librerías necesarias	9
6. Consideraciones técnicas	13
7. Manual de usuario	15
7.1. Menú Principal	15
7.2. Algoritmos Genéticos	16
7.3. Naive Bayes	17
7.4. Redes Neuronales	18
7.5. Clustering	19
7.6. Transfer Learning	19

1. Requisitos Previos

Sistema operativo	Windows 10 o superior (recomendado).
Python	Versión 3.11
Espacio en disco	Al menos 500 MB disponibles.
Memoria RAM necesaria	Mínimo 4 GB.
Conexión a internet	Requerida para instalación de librerías y descarga de modelos

2. Instalación

2.1. Instalar Python



El sistema necesita Python, un lenguaje de programación que permite que el software funcione.

Pasos:

- Visite <https://www.python.org/downloads/>.
- Descargue la versión recomendada (Python 3.11.x).
- Al instalar, marque la casilla **"Add Python to PATH"**.
- Haga clic en *Install Now*.

2.2. Instalar PyCharm



PyCharm es un programa que facilita la ejecución y edición del código si desea explorar o modificar el proyecto.

Pasos:

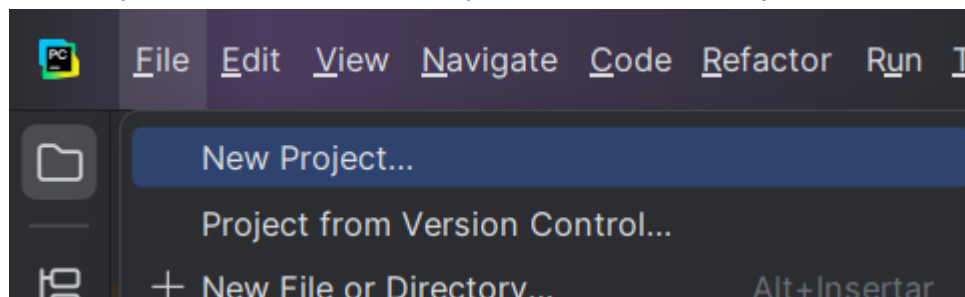
- Visite <https://www.jetbrains.com/pycharm/download/>.
- Descargue la versión **Community** (es gratuita).
- Instale PyCharm como cualquier otro programa de Windows.
- Una vez instalado, puede abrir la carpeta del proyecto y ejecutar `Main.py` con un solo clic.

2.3. Entorno Virtual (.venv)

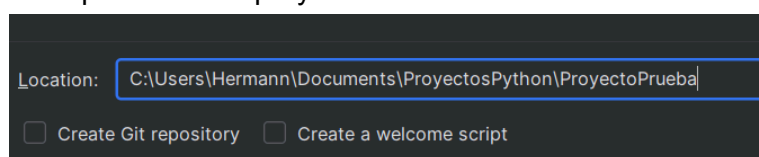
Un entorno virtual permite tener todas las herramientas necesarias sin afectar otros programas de la computadora.

Una forma sencilla de crear un proyecto python nuevo con un entorno virtual es el siguiente:

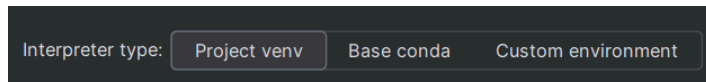
- En PyCharm ir a File o Archivo y seleccionar 'New Project'



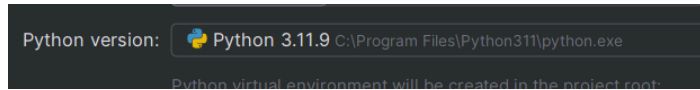
- Elegir un nombre para nuestro proyecto.



- En intérprete elegir 'Project venv'



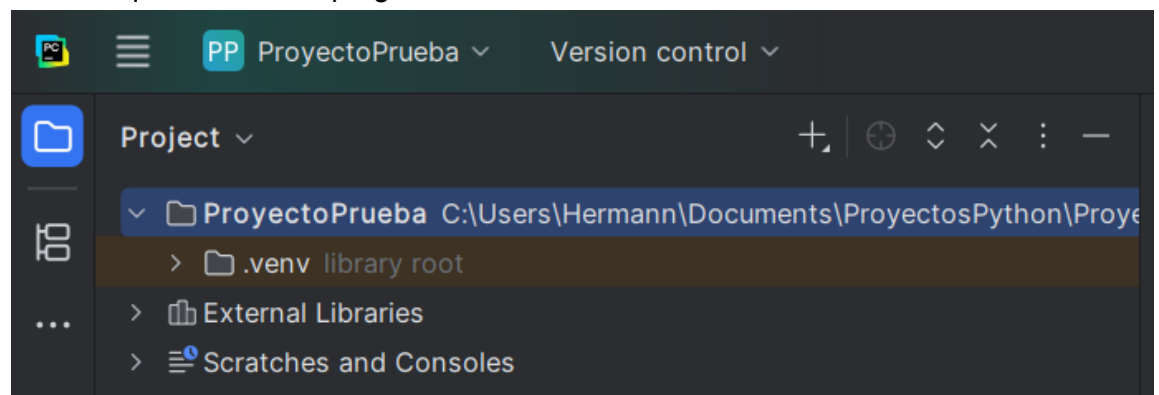
- Y en versión Python elegir 3.11.9 o 3.11.x. Luego dar crear.



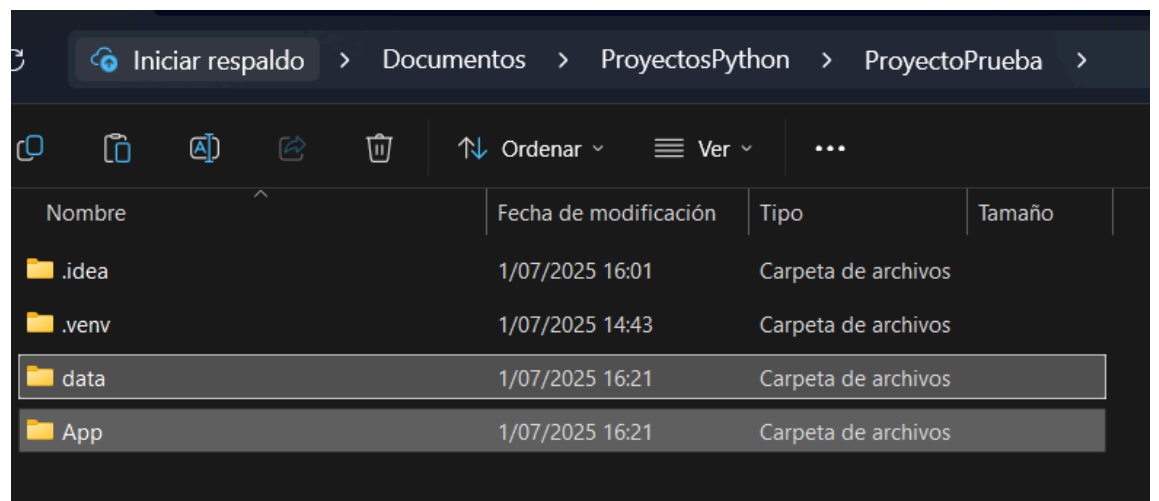
2.4. Cargar proyecto

Este proyecto hace uso de muchas librerías pesadas que complicó un poco su distribución. Es por eso que detallaremos cómo poder levantar el proyecto sin inconvenientes.

Una vez que tenemos el programa anterior creado



Procedemos a copiar las carpetas 'App' y 'data' que contienen los archivos .py y los 'datasets' necesarios a la carpeta de este nuevo proyecto. Estas pueden ser descargadas del repositorio GitHub.



El github puede ser accedido desde aqui:

<https://github.com/GermanCallupeH/ProyectoSI.git>

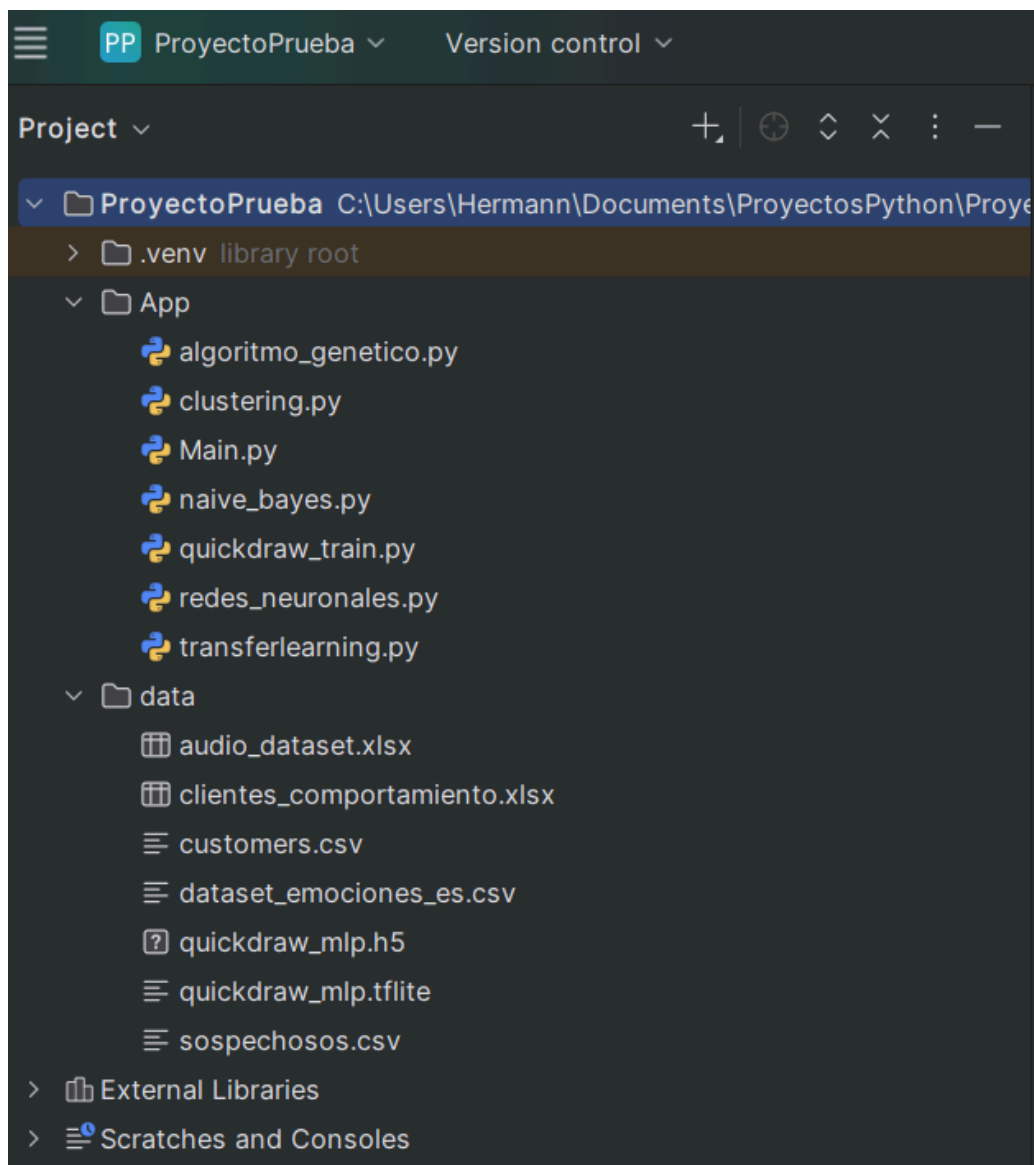
2.5. Errores comunes

Python no reconocido: asegúrese de haber marcado la opción *Add to PATH* durante la instalación.

Faltan librerías: asegúrese de haber ejecutado correctamente el archivo `Main.py`, que instalará automáticamente todo lo necesario.

3. Estructura del proyecto

Luego de copiar la carpeta `App` y `data`, nos debería quedar un proyecto así. Al ejecutar el `main` con un intérprete Python 3.11, se debe empezar a descargar las librerías necesarias, este proceso toma aproximadamente 4-5 minutos, dependiendo de la conexión a internet.



El proyecto está organizado de forma clara y sencilla dentro de una carpeta principal. Esta organización permite identificar fácilmente los archivos del programa,

los módulos de cada aplicación y los archivos de datos necesarios para que todo funcione correctamente.

El proyecto puede ser abierto desde un repositorio en línea o descargado directamente en una carpeta local. Una vez abierto con PyCharm (u otro editor), se puede visualizar una estructura como la siguiente:

La estructura general es la siguiente:

App/: Esta carpeta contiene todos los módulos del sistema, es decir, los archivos que ejecutan cada una de las aplicaciones (como el módulo de Naive Bayes, Clustering, etc.). Cada archivo `.py` representa una parte funcional independiente.

`Main.py`: Este es el archivo principal. Al ejecutarlo, se muestra un menú que permite acceder a cada uno de los módulos del sistema.

`algoritmo_genetico.py`: Lógica del módulo de Algoritmos Genéticos.

`clustering.py`: Contiene la interfaz para el análisis de agrupamiento de datos (clustering).

`naive_bayes.py`: Lógica para clasificar sospechosos con el algoritmo Naive Bayes.

`redes_neuronales.py`: Entrenamiento de dibujos (QuickDraw) con redes neuronales simples.

`transferlearning.py`: Clasificación de emociones usando aprendizaje por transferencia (TF-IDF).

`quickdraw_train.py`: Entrenamiento personalizado con TensorFlow para redes neuronales.

data/: Carpeta donde se almacenan todos los archivos de entrada y salida necesarios para que los módulos funcionen correctamente.

Archivos `.csv` como `customers.csv` o `sospechosos.csv` contienen datos estructurados (como frases, emociones, comportamientos de clientes, etc.).

Archivos `.xlsx` como `clientes_comportamiento.xlsx` contienen datos en formato de Excel.

Archivos `.h5` o `.tflite` son modelos previamente entrenados por el sistema, usados por el módulo de redes neuronales.

.venv/: Esta carpeta se crea automáticamente cuando se configura el entorno virtual del proyecto. No debe modificarse manualmente, ya que contiene las librerías necesarias para que el sistema funcione.

4. Descripción de módulos

4.1. Aplicación de 'Algoritmos Genéticos'

Este módulo simula una técnica inspirada en la evolución natural. En lugar de resolver el problema directamente, prueba con diferentes "candidatos" (por ejemplo, posibles respuestas) y va mejorando generación tras generación, como si fueran especies que se adaptan. El usuario puede observar cómo se optimiza una solución a través de ciclos (generaciones), cruzando y mutando "individuos", hasta encontrar el mejor.

Objetivo: Mostrar cómo la lógica de la evolución biológica puede utilizarse para resolver problemas en informática.

Qué ve el usuario: Una ventana que muestra cómo una población mejora su rendimiento en cada generación hasta llegar al óptimo.

4.2. Aplicación de 'Naive Bayes'

Este módulo es un pequeño "juego detectivesco" en el que se deben adivinar características del culpable basándose en pistas. Utiliza un modelo llamado *Naive Bayes*, que predice la clase más probable dada una serie de condiciones. Aunque es una técnica muy utilizada en clasificación de texto y spam, aquí se presenta en un formato lúdico.

Objetivo: Demostrar cómo funciona una de las técnicas más usadas en inteligencia artificial de forma sencilla y divertida.

Qué ve el usuario: Botones para elegir pistas, y luego un botón para "adivinar" quién es el culpable. El modelo responde con su predicción.

4.3. Aplicación de 'Redes Neuronales'

Este módulo permite al usuario dibujar en una pizarra virtual. Luego, una inteligencia artificial entrenada intenta adivinar qué fue lo que dibujó: ¿un gato? ¿una casa? ¿un pez? El modelo utilizado es una red neuronal parecida a las que se usan en reconocimiento de imágenes reales.

Objetivo: Mostrar cómo una red neuronal puede aprender a identificar dibujos, incluso de manera muy simplificada.

Qué ve el usuario: Una pantalla donde puede dibujar con el mouse y presionar un botón para que el modelo adivine. Si acierta, gana el usuario; si falla, gana la IA.

4.4. Aplicación de 'Clustering'

Este módulo agrupa automáticamente audios en función de características como el ritmo, el tono o la duración. Sin que el modelo sepa qué emoción representa cada audio, intenta crear grupos de sonidos similares. Esto se conoce como *clustering*, y es útil cuando no se tiene una clasificación previa de los datos.

Objetivo: Mostrar cómo un sistema puede descubrir patrones ocultos y hacer agrupaciones por su cuenta.

Qué ve el usuario: Una visualización con puntos de diferentes colores que representan las agrupaciones encontradas. También se muestra un análisis del resultado y sus conclusiones.

4.5. Aplicación de 'Transfer Learning'

Este módulo aprovecha un modelo de lenguaje preentrenado en español (spaCy) para detectar emociones en frases escritas. El usuario puede cargar un archivo con frases etiquetadas (por ejemplo, frases tristes, alegres, etc.), entrenar rápidamente un modelo, y luego probar con nuevas frases.

Objetivo: Mostrar cómo se puede reutilizar conocimiento ya entrenado para resolver nuevos problemas con pocos datos.

Qué ve el usuario: Una ventana donde puede cargar un archivo, entrenar un modelo con un solo clic, y escribir frases para que la IA detecte su emoción, junto con un nivel de confianza.

5. Librerías necesarias

tkinter

¿Para qué sirve?: Es la librería que permite crear interfaces gráficas (ventanas, botones, campos de texto, etc.).

Usada en: Todos los módulos.

Ejemplo de uso: La ventana principal del menú o el botón de "Cargar CSV" en Transfer Learning.

pandas

¿Para qué sirve?: Permite manejar datos estructurados (como hojas de Excel o archivos CSV) de forma eficiente.

Usada en: Transfer Learning y Clustering.

Ejemplo de uso: Leer archivos CSV que contienen frases y emociones para entrenar el modelo.

numpy

¿Para qué sirve?: Facilita el manejo de arreglos numéricos y cálculos matemáticos rápidos.

Usada en: Redes Neuronales, Clustering, y Algoritmos Genéticos.

Ejemplo de uso: Normalizar imágenes o procesar características de audios.

matplotlib

¿Para qué sirve?: Permite crear gráficos y visualizaciones.

Usada en: Clustering.

Ejemplo de uso: Mostrar gráficamente los grupos de audios detectados por los algoritmos.

scikit-learn (sklearn)

¿Para qué sirve?: Es una de las librerías más usadas en inteligencia artificial. Contiene modelos de aprendizaje automático listos para usar.

Usada en: Transfer Learning, Naive Bayes, Clustering.

Ejemplo de uso:

Clasificador de emociones (Transfer Learning).

Clasificador bayesiano del culpable (Naive Bayes).

Algoritmos de agrupamiento (K-Means, DBSCAN) en Clustering.

tensorflow

¿Para qué sirve?: Sirve para crear y entrenar redes neuronales. Es potente pero algo pesado.

Usada en: Redes Neuronales (QuickDraw).

Ejemplo de uso: Entrenar un modelo que reconoce dibujos simples hechos a mano.

pillow (PIL)

¿Para qué sirve?: Manipular imágenes (abrir, modificar, guardar).

Usada en: Redes Neuronales.

Ejemplo de uso: Convertir el dibujo del usuario a una imagen en blanco y negro para que lo entienda la IA.

tqdm

¿Para qué sirve?: Mostrar una barra de progreso durante operaciones largas.

Usada en: Redes Neuronales (entrenamiento).

Ejemplo de uso: Mientras el modelo se entrena, aparece una barra indicando el avance.

spacy

¿Para qué sirve?: Analizar texto en lenguaje natural (gramática, significado, etc.).

Usada en: Transfer Learning.

Ejemplo de uso: Utiliza un modelo preentrenado en español para entender mejor el contexto de las frases.

openpyxl

¿Para qué sirve?: Leer y escribir archivos de Excel (formato .xlsx).

Usada en: Transfer Learning (opcional, si se desea trabajar con Excel).

Ejemplo de uso: Cargar datasets desde archivos de Excel.

os

¿Para qué sirve?: Interactuar con el sistema operativo (como buscar archivos o crear carpetas).

Usada en: Todos los módulos.

Ejemplo de uso: Acceder a la carpeta data/ o guardar un modelo entrenado.

sys

¿Para qué sirve?: Gestionar comandos del sistema, cerrar la aplicación, o instalar paquetes desde código.

Usada en: Menú principal, gestión de dependencias.

Ejemplo de uso: Cerrar la app desde el botón “Salir” o lanzar otro módulo.

subprocess

¿Para qué sirve?: Ejecutar comandos del sistema, como instalar automáticamente paquetes si faltan.

Usada en: Menú principal (verificación de dependencias).

Ejemplo de uso: Instalar automáticamente spacy si no está instalado.

importlib

¿Para qué sirve?: Verifica si una librería ya está instalada.

Usada en: Menú principal (para evitar instalar lo que ya está).

Ejemplo de uso: Revisión rápida antes de instalar paquetes.

6. Consideraciones técnicas

Antes de ejecutar correctamente esta aplicación, es importante tener en cuenta algunos aspectos técnicos que aseguran su correcto funcionamiento. Esta sección está pensada para orientar al usuario sobre los requisitos básicos, el entorno en el que corre el programa y las decisiones que se tomaron para facilitar su instalación.

Entorno de Desarrollo

El proyecto ha sido desarrollado y probado con:

- **Python 3.11**
- **Editor:** PyCharm Community Edition
- **Gestión de librerías:** Uso del propio código para instalar automáticamente dependencias.

Se utilizó PyCharm por su facilidad para trabajar con interfaces gráficas y por su compatibilidad con proyectos estructurados por módulos.

Instalación automática de dependencias

Para facilitar la vida del usuario, el sistema **verifica e instala automáticamente** las librerías necesarias la primera vez que se ejecuta. Esto incluye herramientas para el manejo de texto, entrenamiento de modelos de inteligencia artificial, y generación de gráficos, entre otras.

El usuario no necesita instalar manualmente nada, ya que el propio sistema se encarga de hacerlo si detecta que falta algo. Este proceso puede tardar unos minutos la primera vez, dependiendo de la velocidad de conexión y del equipo.

Estructura Modular

Cada módulo de la aplicación (como el de Naive Bayes, Clustering o Transfer Learning) está diseñado para ejecutarse de forma independiente, pero todos comparten una misma interfaz desde el **menú principal**. Esto hace que el sistema sea flexible, organizado y fácil de mantener.

El sistema puede expandirse en el futuro agregando nuevos módulos de inteligencia artificial o mejoras visuales, sin necesidad de cambiar toda la aplicación.

Peso del Proyecto y Tamaño de Librerías

Se ha hecho un esfuerzo por mantener el proyecto **ligero y funcional**. No se han utilizado modelos de gran tamaño como BERT o distilBERT para evitar que el peso del ejecutable final supere los 300 MB. En su lugar, se han usado alternativas más livianas como TF-IDF con `scikit-learn` o redes neuronales simples con TensorFlow.

Compatibilidad del Ejecutable

La aplicación puede ser empaquetada en un archivo `.exe` (ejecutable de Windows), para facilitar su distribución sin requerir la instalación de Python

en cada equipo. Esto permite que cualquier usuario pueda abrir el sistema con doble clic, como cualquier otro programa de escritorio.

Archivos esperados

Algunos módulos requieren que el usuario cargue archivos propios, como:

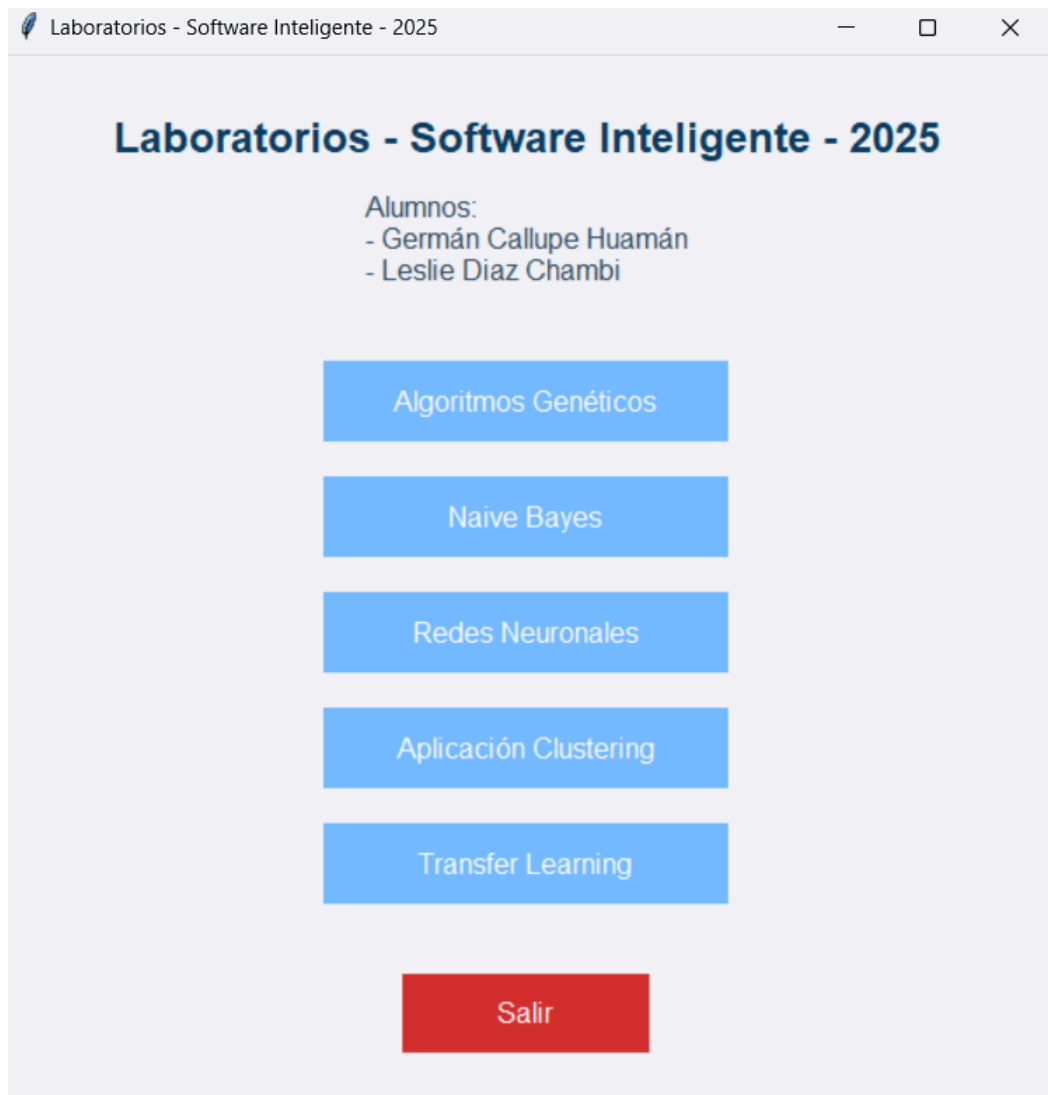
- **Transfer Learning:** necesita un archivo CSV con frases y emociones.
- **Clustering:** necesita un archivo con características de audios.
- **Redes Neuronales (QuickDraw):** requiere archivos .npy con dibujos.

Estos archivos deben colocarse en la carpeta data/ que ya viene incluida en el proyecto.

7. Manual de usuario

Esta sección está pensada para ayudarte a **utilizar cada uno de los módulos** de la aplicación, explicando **paso a paso** lo que debes hacer desde que inicias el programa hasta obtener resultados. Se ha diseñado con una interfaz amigable e intuitiva para facilitar su uso, incluso si no tienes experiencia previa con programas de inteligencia artificial.

7.1. Menú Principal

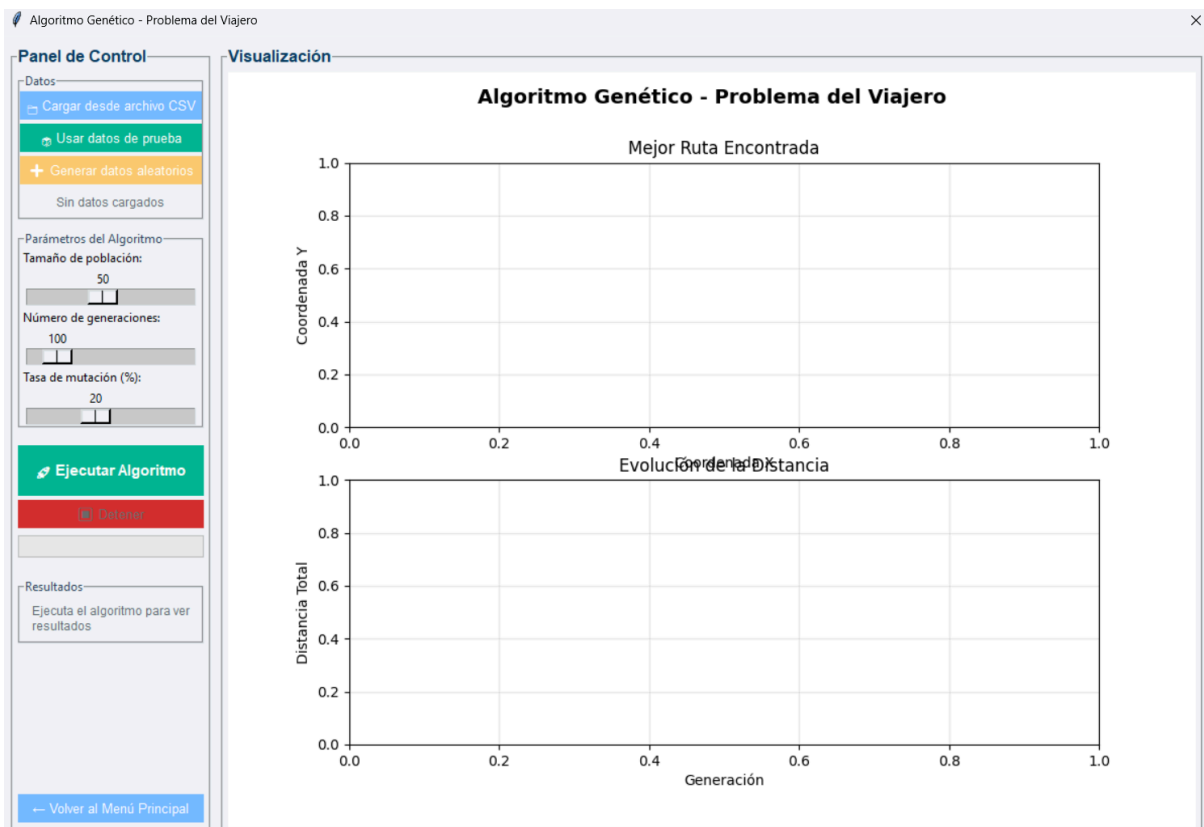


Al iniciar la aplicación, verás una **ventana central** con botones para cada uno de los módulos:

- Algoritmos Genéticos
- Naive Bayes
- Redes Neuronales
- Clustering
- Transfer Learning
- Salir

Cada botón abre una ventana dedicada a una funcionalidad específica. A continuación, se explica cómo usar cada una.

7.2. Algoritmos Genéticos



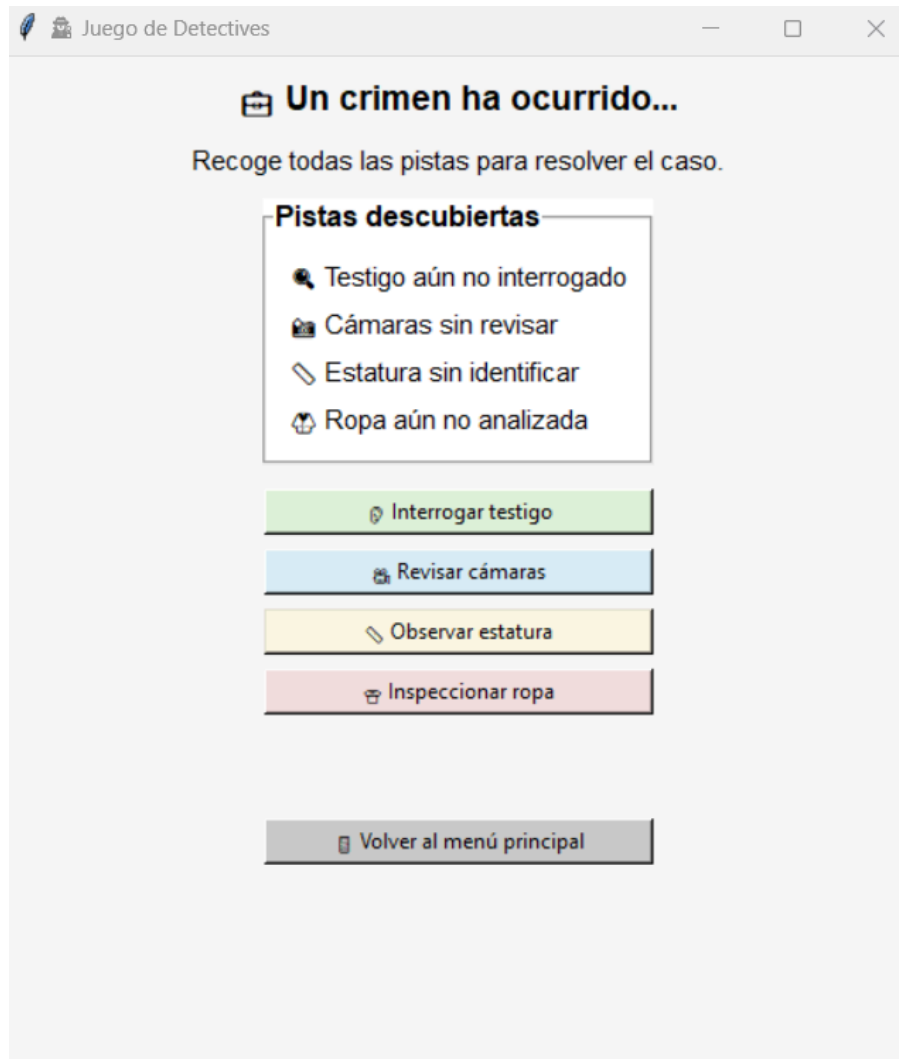
Propósito: Resolver un minijuego interactivo donde se debe adivinar una palabra secreta utilizando un algoritmo evolutivo.

Pasos:

1. Presiona el botón **"Algoritmos Genéticos"**.
2. Se abrirá una ventana con el minijuego.
3. Haz clic en "Iniciar" para que el algoritmo comience a generar posibles soluciones.
4. Observa cómo las generaciones se acercan progresivamente a la solución.
5. Puedes detener o reiniciar el proceso en cualquier momento.

Resultado esperado: El sistema mostrará la cantidad de generaciones necesarias para descubrir la palabra objetivo.

7.3. Naive Bayes



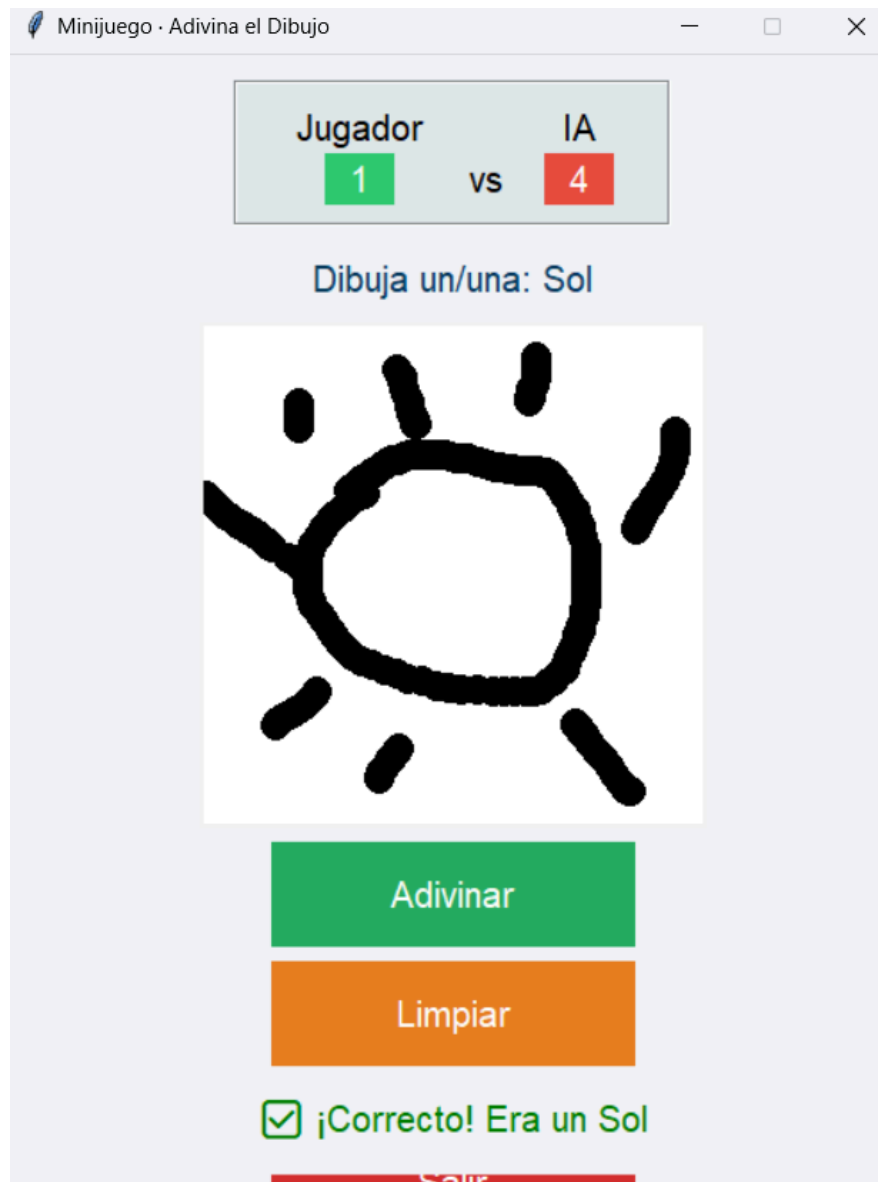
Propósito: Clasificar un crimen basándose en una serie de pistas, utilizando un algoritmo de probabilidad.

Pasos:

1. Presiona el botón **"Naive Bayes"**.
2. La interfaz mostrará pistas ocultas.
3. A medida que haces clic, se revelan pistas aleatorias.
4. Cuando tengas suficientes pistas, presiona "Predecir".
5. El sistema analizará y te dirá qué sospechoso es más probable.

Resultado esperado: Una predicción justificada del sospechoso según la lógica probabilística de Naive Bayes.

7.4. Redes Neuronales



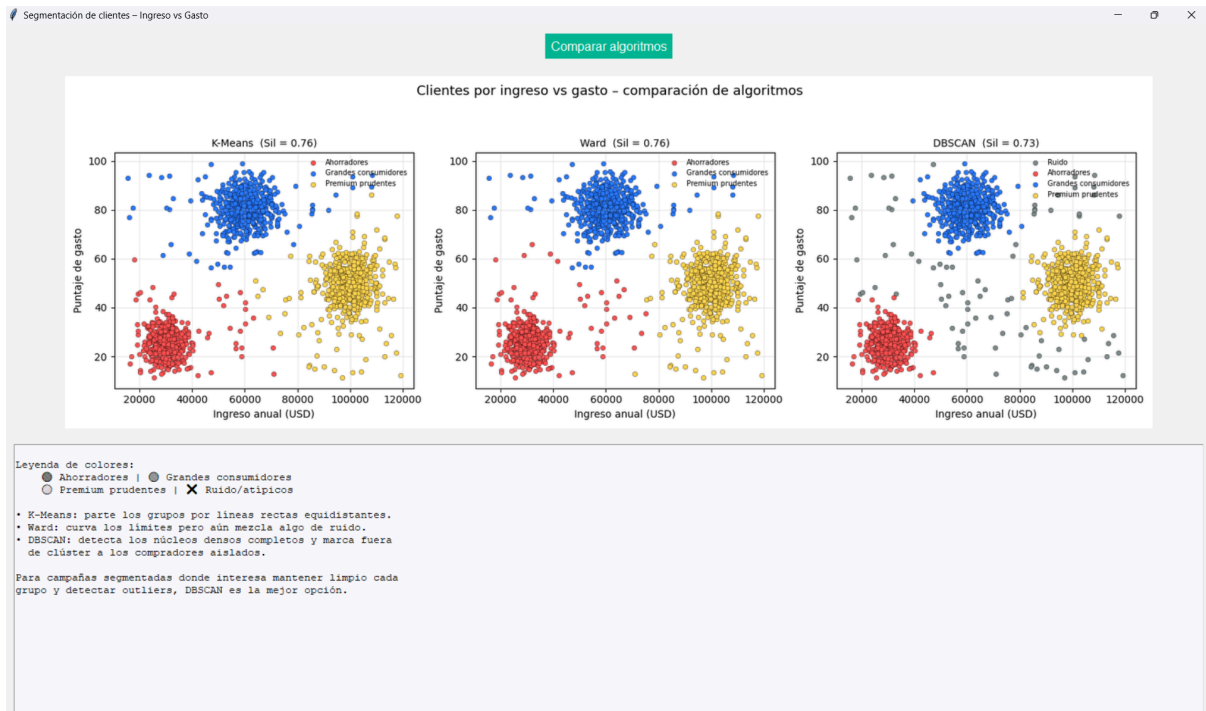
Propósito: Clasificar dibujos simples (como casa, sol, gato, etc.) entrenando una red neuronal con datos de QuickDraw.

Pasos:

1. Presiona el botón **"Redes Neuronales"**.
2. El sistema comenzará a entrenar automáticamente un modelo.
3. Verás el progreso del entrenamiento y la precisión final.
4. El modelo se guarda automáticamente para futuras predicciones.

Resultado esperado: Precisión del modelo en clasificar imágenes simples y mensaje de confirmación al finalizar el entrenamiento.

7.5. Clustering



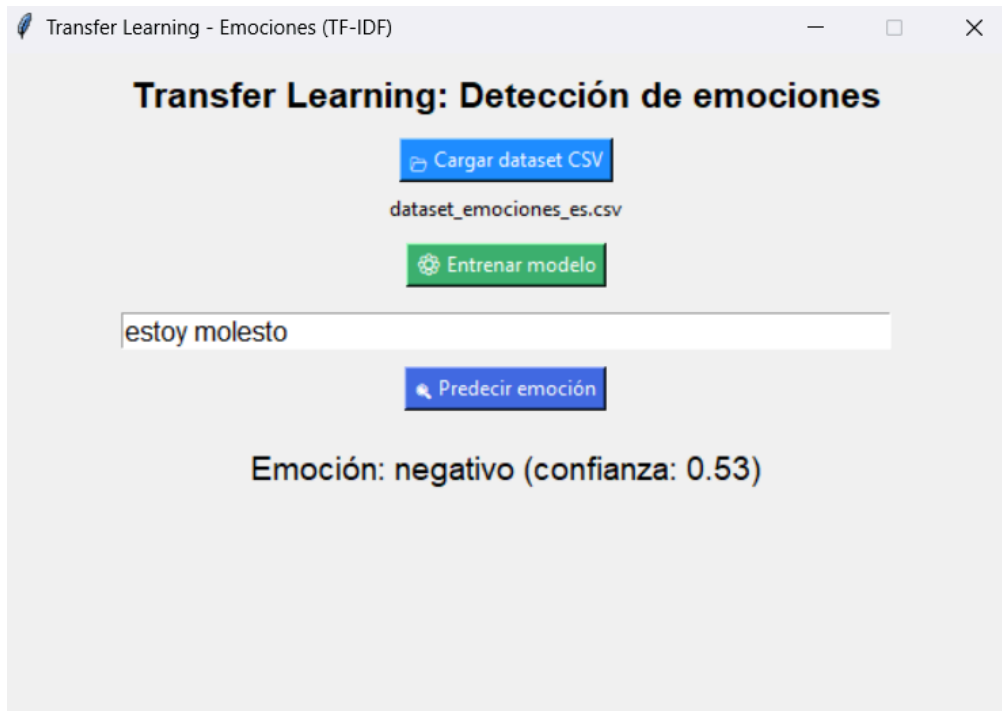
Propósito: Agrupar grabaciones de audio según sus características (como velocidad o tono emocional) usando diferentes algoritmos de agrupamiento.

Pasos:

1. Presiona el botón **"Aplicación Clustering"**.
2. Elige el archivo Excel que contiene las características de los audios.
3. Observa los gráficos de agrupamiento generados (K-Means, Ward, DBSCAN).
4. Se mostrarán también conclusiones automáticas sobre los resultados.

Resultado esperado: Visualización de los grupos detectados y un pequeño informe interpretativo.

7.6. Transfer Learning



Propósito: Detectar emociones en frases de texto utilizando un modelo entrenado sobre frases etiquetadas (positiva, negativa, etc.).

Pasos:

1. Presiona el botón **"Transfer Learning"**.
2. Carga un archivo CSV con frases y emociones.
3. Presiona "Entrenar" para que el sistema prepare el modelo.
4. Escribe una frase y presiona "Predecir emoción".
5. El sistema mostrará la emoción detectada y su nivel de confianza.

Resultado esperado: Una etiqueta emocional asociada a la frase y una puntuación de confianza (por ejemplo, "Emoción: negativa (confianza: 0.92)").