

Antonio de Jesús Covarrubias Sánchez

Registro: 22110347

T-7E

Ingeniería Mecatrónica

Sistemas Expertos

Profesor: Cabrera Arellano Mauricio Alejandro

Proyecto 3er. Parcial

Sistema Pokexperto.



ceti CENTRO DE ENSEÑANZA
TÉCNICA INDUSTRIAL



Enlace perfil de GitHub:

<https://github.com/tonycovasan22110347>

Enlace repositorio de GitHub:

<https://github.com/tonycovasan22110347/Sistema-experto-Pokesistema-experto->

Sistema Pokexperto:

Versión: 2.0

Fecha: 04 de junio de 2025

Introducción

Este documento presenta el "Sistema de Información y Efectividad de Pokémon", una aplicación de escritorio integral desarrollada en Python con la librería Tkinter. Diseñado para brindar una experiencia interactiva y enriquecedora, este sistema ofrece a los usuarios las siguientes capacidades clave:

- **Búsqueda Detallada de Pokémon:** Accede a información exhaustiva como tipos, estadísticas base, mejor estadística y un análisis profundo de su efectividad defensiva.
- **Calculadora de Efectividad de Tipos:** Simula combates para entender las interacciones de tipos entre ataques y defensores.
- **Gestión de Datos de Pokémon:** Añade y modifica Pokémon personalizados en la base de datos local del sistema.
- **Búsqueda de Movimientos:** Consulta detalles de movimientos Pokémon (tipo, categoría, poder, etc.).
- **Sugerencias de Counters:** Obtén recomendaciones estratégicas de Pokémon y movimientos para contrarrestar a un Pokémon específico.

El sistema se fundamenta en una base de conocimiento flexible y un motor de inferencia robusto que aplica reglas de efectividad de tipos para generar análisis estratégicos y una visualización intuitiva.

Arquitectura del Sistema

El proyecto está diseñado con una arquitectura modular, donde cada componente tiene una responsabilidad bien definida, lo que facilita el desarrollo, el mantenimiento y la futura escalabilidad:

- **interfaz.py:** El componente principal de la Interfaz Gráfica de Usuario (GUI). Gestiona la interacción con el usuario, recibe entradas y presenta los resultados de forma visual.
- **motor_inferencia.py:** El "cerebro" del sistema. Actúa como el puente entre la interfaz y la lógica de datos/reglas. Procesa solicitudes de la interfaz, coordina las operaciones con la base de conocimiento y las reglas, y prepara los datos para su visualización.
- **base_conocimiento.py:** Encargado de la gestión de la base de datos. Se ocupa de cargar los datos de Pokémon y movimientos desde archivos CSV (Pokemon.csv y movimientos_pokemon.csv) en memoria, y de persistir cualquier cambio de vuelta a los archivos.
- **reglas.py:** Contiene el conjunto de definiciones de tipos Pokémon y las complejas reglas de efectividad entre ellos. Es el corazón del sistema de inferencia de efectividad.
- **Pokemon.csv:** El archivo CSV principal que almacena la información de los Pokémon (nombre, tipos, estadísticas, etc.).
- **movimientos_pokemon.csv:** El archivo CSV que contiene la base de datos de movimientos Pokémon (nombre, tipo, poder, etc.).
- **Pokemon.PNG:** La imagen del logo utilizada en la interfaz de usuario.
- **dataset/:** Una carpeta dedicada a contener los archivos CSV para una mejor organización.

Requisitos del Sistema y Preparación

Para ejecutar esta aplicación, necesitarás:

- **Python:** Versión 3.x (recomendado 3.8 o superior).
- **Librerías de Python:**
 - pandas: Para la gestión y manipulación de datos (pip install pandas)
 - Pillow: Para la carga y visualización de imágenes PNG (pip install Pillow)
 - tkinter (Tkinter): Librería estándar de Python para interfaces gráficas (normalmente incluida con Python).

Pasos para la Configuración y Ejecución:

1. **Instalar Librerías:** Abre tu terminal (Símbolo del Sistema en Windows, Terminal en macOS/Linux) y ejecuta los siguientes comandos para instalar las dependencias:
 - Bash
 - pip install pandas
 - pip install Pillow
2. **Organizar Archivos:**
 - Descarga todos los archivos .py (base_conocimiento.py, interfaz.py, motor_inferencia.py, reglas.py).
 - Descarga los archivos CSV (Pokemon.csv, movimientos_pokemon.csv).
 - Descarga la imagen Pokemon.PNG.
 - Crea una **nueva carpeta llamada dataset** en el mismo directorio donde guardaste los archivos .py.
 - **Mueve Pokemon.csv y movimientos_pokemon.csv dentro de la carpeta dataset.**
 - Asegúrate de que Pokemon.PNG esté directamente en el directorio principal del proyecto, junto a los archivos .py.

3. Ejecutar la Aplicación:

- Abre tu terminal y navega al directorio principal de tu proyecto (donde se encuentra interfaz.py).
- Ejecuta el siguiente comando:
 - Bash
 - python interfaz.py
- La ventana principal del "Buscador y Efectividad de Pokémon" debería aparecer.

Funcionamiento Detallado de los Módulos

La robustez del sistema reside en la interacción bien definida de sus módulos:

base_conocimiento.py - Gestión de la Base de Datos

Este módulo es el custodio de los datos del sistema.

- **Carga y Persistencia:** Se encarga de cargar los DataFrame de Pandas para Pokémon (Pokemon.csv) y Movimientos (movimientos_pokemon.csv) al inicio de la aplicación. Cualquier adición o modificación de datos es gestionada por este módulo, asegurando que los cambios se guarden de forma persistente en los archivos CSV.
- **Columna UserAdded:** Introduce y gestiona la columna booleana UserAdded en Pokemon.csv. Esta columna es vital para diferenciar entre Pokémon originales (valor False) y aquellos agregados por el usuario (valor True), permitiendo un control preciso sobre la edición.
- **Manejo de Rutas:** Utiliza os.path para construir rutas de archivo relativas, haciendo que el sistema sea portátil y funcione correctamente independientemente de dónde se encuentre el directorio del proyecto.

reglas.py - Lógica de Efectividad de Tipos

El corazón estratégico del sistema, este módulo define cómo interactúan los tipos Pokémon.

- **efectividad_ataque:** Un diccionario crucial que mapea cada tipo atacante a una lista de tipos defensores a los que es "eficaz" (2x daño), "no_efectivo" (0.5x daño) o "inmune" (0x daño).
- **calcular_efectividad(tipos_ataque):** Simula un ataque. Recibe los tipos de un Pokémon atacante y calcula la efectividad combinada contra todos los demás tipos disponibles, aplicando las reglas de multiplicación de daño para múltiples tipos y resolviendo conflictos (ej., un Pokémon tipo Agua/Volador recibe 4x daño de ataques Eléctricos, no 2x).
- **obtener_efectividad_defensiva(tipos_defensor):** Simula una defensa. Recibe los tipos de un Pokémon defensor y determina, para cada tipo de ataque posible en el juego, cuánto daño recibiría el Pokémon defensor. Esta función es clave para identificar las debilidades, resistencias e inmunidades del Pokémon.

motor_inferencia.py - El Procesador Central

Actúa como el cerebro intermedio, procesando las solicitudes de la interfaz y orquestando la lógica con la base de datos y las reglas.

- **analizar_pokemon_por_nombre(nombre):**
 - Consulta base_conocimiento.py para obtener los datos del Pokémon.
 - Traduce los tipos de Pokémon de inglés (en CSV) a español para la interfaz.
 - Calcula la "Mejor Stat" del Pokémon.
 - Invoca obtener_efectividad_defensiva de reglas.py para el análisis de debilidades/fortalezas.
 - **NUEVO: Prepara datos para Sugerencias de Counters:** Si el Pokémon es encontrado, también precalcula qué tipos son *muy* eficaces contra él para luego sugerir Pokémon y movimientos counter.
 - Retorna un diccionario completo de datos del Pokémon, incluyendo su ID y el valor de UserAdded.

agregar_nuevo_pokemon_a_bd(datos_pokemon_dict):

- Valida la entrada del usuario (nombre único, tipos válidos, stats numéricos y positivos).
 - Genera un nuevo ID para el Pokémon.
 - Traduce los tipos de español a inglés para guardar en el CSV.
 - Establece UserAdded a True.
 - Añade el nuevo Pokémon al DataFrame y lo guarda mediante base_conocimiento.py.
- **modificar_pokemon_en_bd(pokemon_id, datos_pokemon_dict):**
 - Localiza el Pokémon por su ID.
 - Valida los datos de entrada (nombre único entre otros Pokémon, stats).
 - Actualiza los datos del Pokémon en el DataFrame y los guarda.
 - **buscar_movimiento_por_nombre(nombre):**
 - Consulta movimientos_pokemon.csv a través de base_conocimiento.py para obtener los detalles de un movimiento específico (Tipo, Categoría, Poder, Precisión, PP, Efecto).
 - Retorna un diccionario con la información del movimiento.
 - **buscar_counters_para_pokemon(pokemon_tipos_debiles):**
 - Recibe la lista de tipos a los que el Pokémon objetivo es débil.
 - **Sugerencias de Pokémon:** Busca en la base de datos Pokémon que tengan al menos uno de esos tipos fuertes como su Tipo 1, filtrando por legendarios para dar opciones más comunes y ordenando por poder.
 - **Sugerencias de Movimientos:** Para cada tipo al que el Pokémon objetivo es débil, busca los 3 movimientos más potentes de ese tipo, consultando movimientos_pokemon.csv.
 - Devuelve listas de Pokémon y un diccionario de movimientos sugeridos.

interfaz.py - La Ventana del Usuario

La interfaz es el punto de interacción principal, presentando la información de forma clara y permitiendo la entrada de datos.

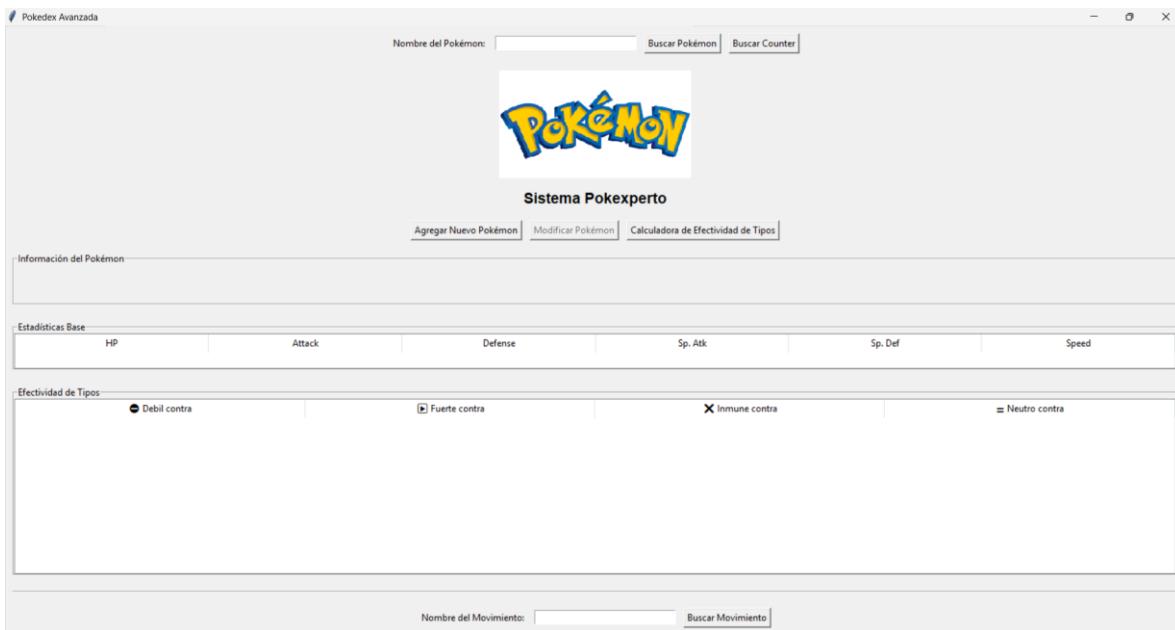
- **Diseño con Tkinter y ttk:** Utiliza `ttk.Treeview` para tablas estilizadas y `tk.Toplevel` para ventanas secundarias, mejorando la experiencia de usuario.
- **Gestión de Estados:** Habilita/deshabilita botones (Modificar Pokémon) basándose en el estado de los datos (ej., si un Pokémon es `UserAdded`).
- **Refrescado de Datos:** Llama a las funciones del `motor_inferencia.py` y refresca las tablas y campos de texto para mostrar la información actualizada.
- **Manejo de Errores:** Emplea `messagebox` para comunicar advertencias, errores o éxitos al usuario.

Guía de Uso del Sistema

Esta sección te guiará paso a paso por las funcionalidades principales de la aplicación.

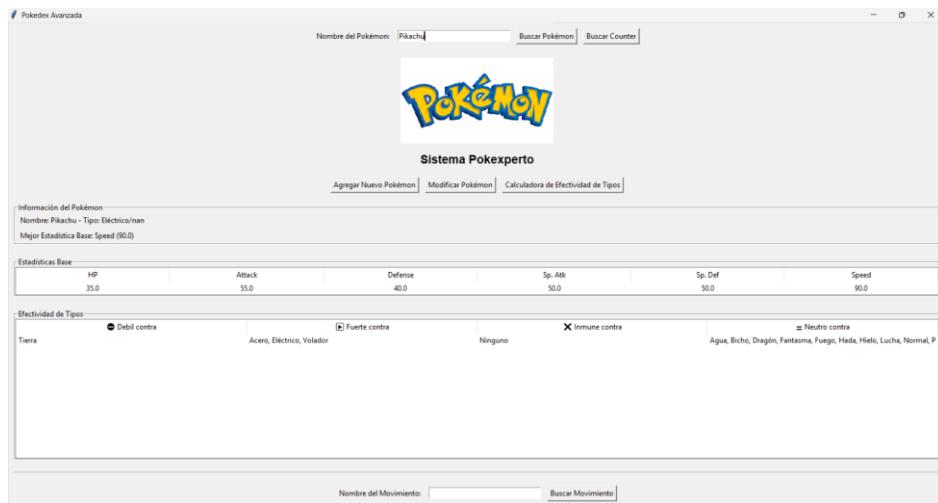
Interfaz Principal y Búsqueda de Pokémon

La ventana principal es tu punto de partida.



1. Buscar un Pokémon:

- En el campo "Nombre del Pokémon:", ingresa el nombre (ej., "Pikachu", "Charizard").
- Haz clic en "**Buscar Pokémon**".



Resultados de la Búsqueda de Pokémon:

- **Información del Pokémon:** Nombre, Tipos y la Mejor Stat.
- **Estadísticas Base:** Una tabla compacta con HP, Attack, Defense, Sp. Atk, Sp. Def y Speed.
- **Efectividad de Tipos:** Una tabla detallada de cómo otros tipos afectan al Pokémon (Fuerte, Débil, Inmune, Neutro).
- **Sugerencias de Counters:**
 - **Pokémon Sugeridos:** Listado de Pokémon de tu base de datos que tienen tipos fuertes contra las debilidades del Pokémon buscado.
 - **Movimientos Sugeridos:** Los 3 movimientos más potentes para cada tipo al que el Pokémon buscado es débil.

Búsqueda y Visualización de Movimientos

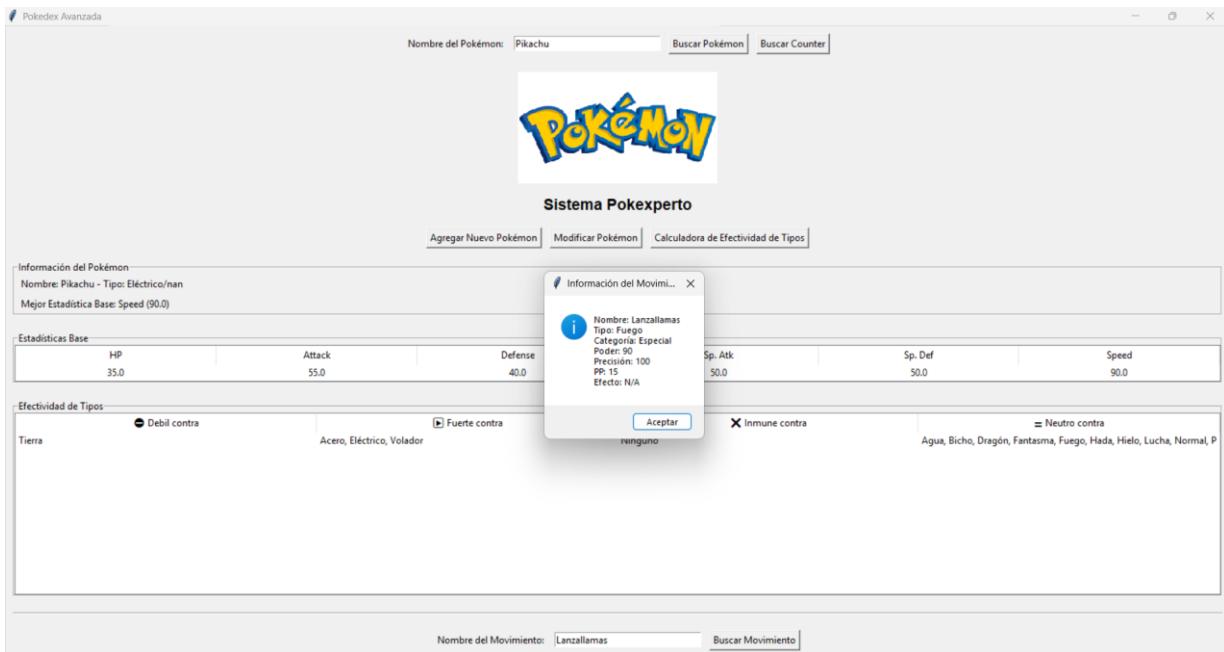
Puedes consultar los detalles de cualquier movimiento Pokémon.

- 1. Campo de Búsqueda de Movimiento:** Ubica la sección "Búsqueda de Movimientos" y el campo "Nombre del Movimiento:".
- 2. Ingresar Nombre:** Escribe el nombre de un movimiento (ej., "Lanzallamas", "Terremoto").
- 3. Botón "Buscar Movimiento":** Haz clic para mostrar su información.

[Inserta una captura de pantalla de la interfaz principal con un movimiento buscado, mostrando su información (Nombre, Tipo, Categoría, Poder, etc.).]

Resultados de la Búsqueda de Movimiento:

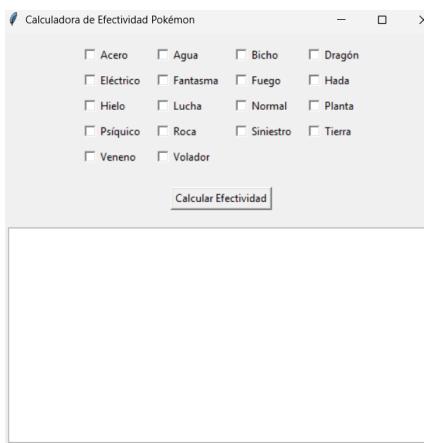
- Se mostrará el Nombre del Movimiento, su Tipo, Categoría (Físico, Especial, Estado), Poder, Precisión, Puntos de Poder (PP) y el Efecto que tiene en combate.



Calculadora de Efectividad de Tipos

Entiende las interacciones de tipos en combate.

1. **Botón "Abrir Calculadora de Tipos":** Haz clic en este botón en la ventana principal. Se abrirá una nueva ventana.



2. **Seleccionar Tipos de Ataque:** Marca las casillas de los tipos de ataque que deseas calcular (uno o varios).
3. **Botón "Calcular Efectividad":** Haz clic para ver los resultados.



Resultados del Cálculo: La tabla de resultados mostrará cómo los tipos seleccionados se desempeñan contra otros tipos:

- **► Efectivos contra:** Tipos a los que hacen el doble de daño.
- **● Poco efectivos contra:** Tipos a los que hacen la mitad de daño.
- **✗ Sin efecto contra:** Tipos a los que no hacen daño.
- **≡ Neutros contra:** Tipos a los que hacen daño normal.

Agregar Nuevo Pokémon

Extiende tu base de datos con Pokémon personalizados.

1. **Botón "Agregar Nuevo Pokémon"**: Haz clic en este botón en la ventana principal. Se abrirá una nueva ventana de formulario.

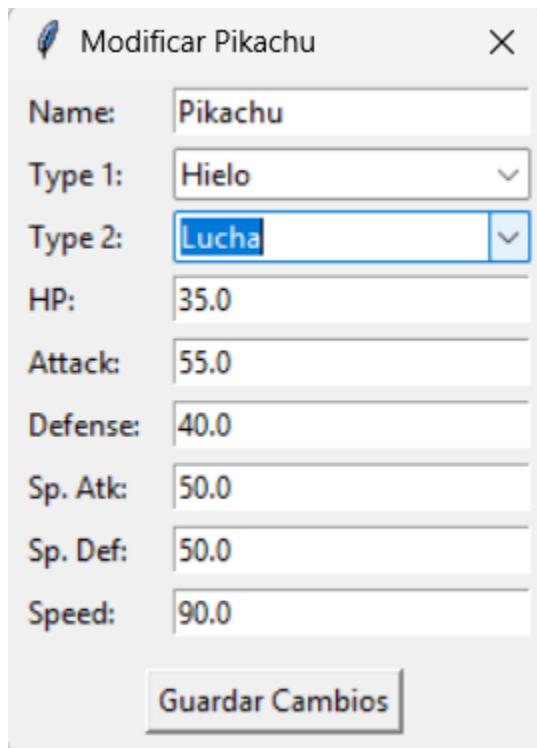
| Name: | Rowlet |
|----------|---------|
| Type 1: | Planta |
| Type 2: | Volador |
| HP: | 68 |
| Attack: | 55 |
| Defense: | 55 |
| Sp. Atk: | 50 |
| Sp. Def: | 50 |
| Speed: | 42 |

2. **Rellenar el Formulario**: Ingresa el nombre (único), selecciona el Tipo 1 (obligatorio) y Tipo 2 (opcional) de las listas desplegables, e ingresa los valores numéricos de las seis estadísticas base (HP, Attack, Defense, Sp. Atk, Sp. Def, Speed).
3. **Botón "Guardar Pokémon"**: Haz clic para añadir el Pokémon.
 - Recibirás un mensaje de éxito o de error (ej., si el nombre ya existe o los stats no son números válidos).
 - La ventana se cerrará automáticamente al guardar con éxito.
4. **Verificar Adición**: Puedes buscar el Pokémon recién añadido en la interfaz principal.

Modificar Pokémon Personalizados

Edita la información de los Pokémon que tú mismo has agregado. Los Pokémon originales de la base de datos no son modificables para proteger la integridad de los datos base.

1. **Buscar el Pokémon a Modificar:** En la interfaz principal, ingresa y busca el nombre del Pokémon que deseas editar.



The screenshot shows a dialog box titled "Modificar Pikachu". It contains the following fields:

- Name: Pikachu
- Type 1: Hielo
- Type 2: Lucha
- HP: 35.0
- Attack: 55.0
- Defense: 40.0
- Sp. Atk: 50.0
- Sp. Def: 50.0
- Speed: 90.0

A "Guardar Cambios" button is at the bottom.

2. **Botón "Modificar Pokémon":** Este botón solo se habilitará si el Pokémon actualmente buscado fue agregado previamente por el usuario. Haz clic en él.

[Inserta una captura de pantalla de la ventana "Modificar Pokémon" precargada con los datos del Pokémon editable.]

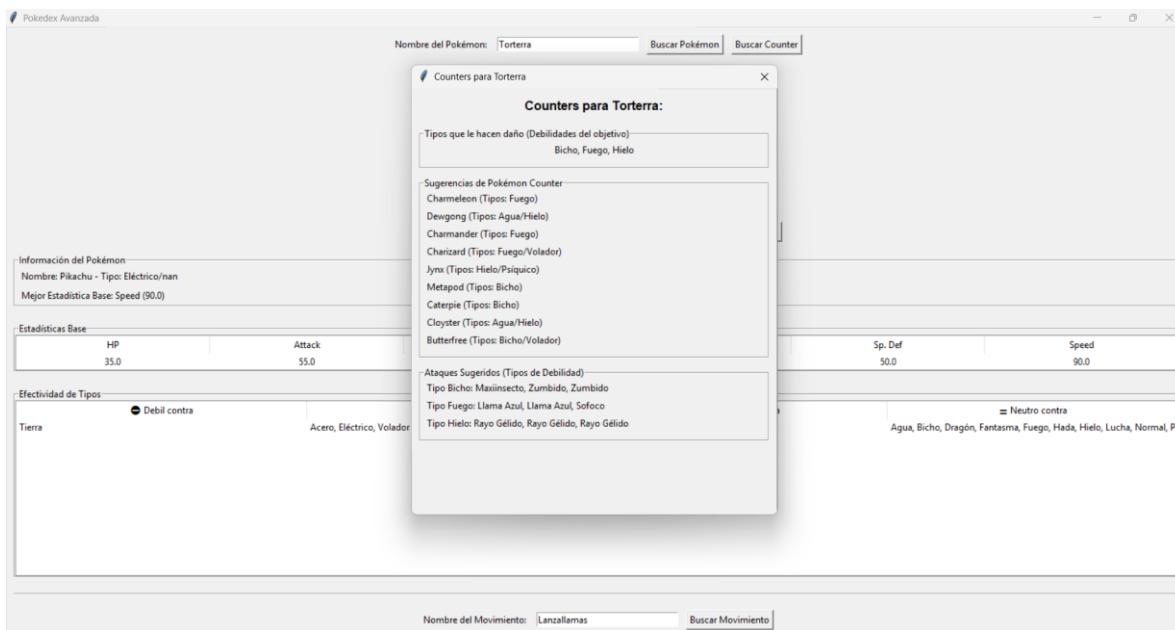
3. **Editar el Formulario:** La ventana se abrirá con todos los campos precargados. Realiza los cambios necesarios.
4. **Botón "Guardar Cambios":** Haz clic para aplicar las modificaciones.
 - Recibirás un mensaje de éxito o de error.
 - La ventana se cerrará y la interfaz principal se actualizará automáticamente con los datos modificados.

Sugerencias de Counters:

- Esta es una funcionalidad estratégica que te ayuda a identificar posibles respuestas contra el Pokémon buscado. El sistema analiza las debilidades defensivas del Pokémon buscado y sugiere opciones tanto de Pokémon como de Movimientos que son efectivos contra esas debilidades.

○ Pokémon Sugeridos:

- **Funcionamiento:** El sistema busca en su base de datos de Pokémon aquellos que tienen un tipo principal (Tipo 1) que es directamente "Fuerte contra" (es decir, el Pokémon buscado es Débil contra ese tipo) las debilidades del Pokémon objetivo. Prioriza Pokémon que no son Legendarios para ofrecer opciones más accesibles.
- **Presentación:** Muestra una lista de nombres de Pokémon, indicando sus tipos para facilitar la identificación.



○ Movimientos Sugeridos:

- **Funcionamiento:** Para cada tipo al que el Pokémon buscado es "Débil contra", el sistema busca en la base de datos de movimientos los tres movimientos más potentes de ese tipo específico.
- **Presentación:** Los movimientos se agrupan por el tipo al que pertenecen facilitando la elección de ataques clave.

Conclusión

El "Sistema de Información y Efectividad de Pokémon" es una herramienta robusta y amigable que combina la gestión de datos, la inferencia estratégica y una interfaz intuitiva. Permite a los usuarios explorar el vasto universo Pokémon, comprender sus mecánicas de combate y personalizar su experiencia al gestionar sus propios registros. Su diseño modular garantiza la facilidad de mantenimiento y sienta las bases para futuras expansiones.