



Sistemas Expertos

“Practica 3”

Manual de Usuario

Ernesto Misael Silva Muñoz

Registro: 22310168

Grupo: 7F

Ingeniería Mecatrónica

Tema: Stranger Things

1. Introducción

El presente documento describe el diseño, funcionamiento y arquitectura de un Sistema Experto de adivinación, comúnmente conocido como "Akinator". El objetivo principal de este proyecto es desarrollar una Inteligencia Artificial capaz de adivinar un personaje (en este caso, del universo de "Stranger Things") en el que el usuario esté pensando, a través de una serie de preguntas de "sí" o "no".

El sistema no solo adivina, sino que cumple con un requisito clave de los sistemas inteligentes: **el aprendizaje**. Cuando el sistema falla en su adivinanza, es capaz de interrogar al usuario para adquirir nuevo conocimiento, modificar su base de reglas y expandir su árbol de decisión, asegurando que no cometerá el mismo error en el futuro.

Este reporte está dividido en dos secciones principales: un **Manual de Usuario** que explica cómo interactuar con la aplicación, y un **Análisis de la Arquitectura** que detalla la teoría de Sistemas Expertos implementada, incluyendo reglas, casos y encadenamiento hacia adelante.

2. Manual de Usuario

Esta sección describe cómo un usuario final debe interactuar con la aplicación "Adivinador de Stranger Things".

2.1. Inicio del Juego

Al ejecutar el programa, el usuario es recibido con la ventana principal. Esta ventana contiene:

1. **Un Título:** "Adivinador de Stranger Things".
2. **Un Área de Texto:** Inicialmente, invita al usuario a pensar en un personaje.
3. **Un Área de Imagen:** Muestra una imagen genérica de un "adivino" o bola de cristal.
4. **Botones de Interacción:** Dos botones grandes: "Sí" y "No".

El juego comienza en cuanto el usuario responde a la primera pregunta mostrada en el área de texto.

2.2. Proceso de Adivinación

La interacción es simple e intuitiva:

1. El sistema mostrará una pregunta en el área de texto (p.ej., "¿Tu personaje tiene poderes psíquicos?").
2. El usuario debe hacer clic en "Sí" o "No" según el personaje en el que esté pensando.
3. Con cada respuesta, el sistema refinará su búsqueda y presentará una nueva pregunta. La imagen del "adivino" permanecerá visible.

2.3. Adivinanza Final

Eventualmente, el sistema creará saber la respuesta. En este punto:

- El **texto** cambiará a una pregunta de confirmación (p.ej., "¿Estás pensando en 'Once'?").
- La **imagen** cambiará del "adivino" a la foto del personaje que está intentando adivinar (p.ej., [Once.png](#)).

El usuario debe responder "Sí" o "No" a esta suposición final.

2.4. Módulo de Aprendizaje (Si la IA Falla)

Este es el núcleo interactivo del sistema. Si el usuario responde "**No**" a la adivinanza final:

1. **Diálogo 1 (Pedir Respuesta):** El sistema mostrará una ventana emergente admitiendo su derrota y preguntará: "¿En qué personaje estabas pensando?".
2. **Diálogo 2 (Pedir Pregunta):** Tras recibir la respuesta (p.ej., "Max Mayfield"), el sistema pedirá una pregunta de diferenciación: "Dame una pregunta (Sí/No) que sea 'Sí' para 'Max Mayfield' y 'No' para '[personaje que adivinó mal]'".
3. **Confirmación:** Una vez introducida la pregunta, el sistema mostrará un mensaje de "¡Gracias, he aprendido!" y el juego se reiniciará. El nuevo conocimiento queda guardado permanentemente.

3. Arquitectura del Sistema (Teoría Aplicada)

El "cerebro" de esta aplicación no es una simple lista de personajes, sino un **Sistema Experto** dinámico basado en un árbol de decisión. A continuación, se detalla cómo se integran los conceptos teóricos clave.

3.1. Base de Conocimiento: Un Árbol de Reglas y Casos

El conocimiento de la IA se almacena en una estructura de datos de árbol binario, donde cada nodo representa un fragmento de lógica.

- **Reglas (Nodos Intermedios):** Cada nodo que representa una **pregunta** (p.ej., "¿Es un monstruo?") funciona como una **Regla** en la base de conocimiento. Una regla **SI-ENTONCES** se aplica implícitamente con cada respuesta del usuario.
 - *Ejemplo de Regla:* **SI** la respuesta a "¿Es un monstruo?" es **SÍ**, **ENTONCES** seguir la rama "Sí" (que lleva a la regla/caso "Demogorgon"). **SI** es **NO**, **ENTONCES** seguir la rama "No" (que lleva a la regla "¿Tiene poderes?").
- **Casos (Nodos Hoja):** Cada nodo final, o "hoja", del árbol representa un **Caso** conocido. Un caso es una conclusión o un hecho final (un personaje). La ruta completa desde la raíz hasta esa hoja define inequívocamente ese caso.
 - *Ejemplo de Caso:* El caso "**Once**" se define por la ruta: (¿Es un monstruo? = No) \rightarrow (¿Tiene poderes psíquicos? = Sí) \rightarrow (¿Es el villano...? = No).

3.2. Motor de Inferencia: Encadenamiento Hacia Adelante

El sistema utiliza una estrategia de **Encadenamiento Hacia Adelante** (*Forward Chaining*) para navegar por la base de conocimiento. Este motor de inferencia funciona de la siguiente manera:

1. **Estado Inicial:** La "memoria de trabajo" de la IA está vacía. No tiene hechos sobre el personaje del usuario.
2. **Adquisición de Hechos:** El sistema presenta la primera regla (la pregunta raíz). La respuesta del usuario (**Sí** o **No**) se convierte en el **primer hecho** conocido.
3. **Disparo de Reglas:** Este nuevo hecho "dispara" la siguiente regla aplicable. Si el usuario dice "No" a "¿Es un monstruo?", este hecho se añade a la memoria. El motor busca la siguiente regla en esa rama del árbol (p.ej., "¿Tiene poderes psíquicos?") y la presenta.
4. **Ciclo de Inferencia:** Este proceso se repite. Cada respuesta del usuario se añade a la pila de hechos, lo que permite al motor "encadenar" hacia la siguiente regla relevante.
5. **Conclusión:** El encadenamiento continúa hasta que no se pueden disparar más reglas, lo que ocurre cuando se llega a un **Caso** (un nodo hoja). En este punto, el motor presenta el caso como su conclusión final.

3.3. Módulo de Adquisición de Conocimiento (Aprendizaje)

El requisito de que el sistema "debe aprender" se cumple mediante un módulo de adquisición de conocimiento que se activa cuando el motor de inferencia llega a una conclusión errónea.

El proceso es una "cirugía" en el árbol de conocimiento:

1. **Identificación del Fallo:** La IA identifica el **Caso** (la hoja, p.ej., "Demogorgon") al que llegó incorrectamente.
2. **Adquisición de Datos:** Mediante los diálogos emergentes, el sistema adquiere dos nuevos datos del "experto" (el usuario):
 - El nuevo **Caso** correcto (p.ej., "Vecna").
 - Una nueva **Regla** para diferenciarlos (p.ej., "¿Es el villano de la T4?").
3. **Modificación del Árbol:** El sistema realiza la siguiente transformación:
 - El nodo hoja "Demogorgon" **deja de ser un Caso**.
 - Se **convierte en una Regla**, reemplazando su texto con la nueva pregunta ("¿Es el villano de la T4?").
 - Crea **dos nuevos nodos Caso** (hojas) que cuelgan de esta nueva regla:
 - El hijo "Sí" se convierte en el nuevo caso "Vecna".
 - El hijo "No" se convierte en el caso original "Demogorgon".

Este proceso expande dinámicamente la Base de Conocimiento, volviéndose más preciso con cada partida fallida.

4. Implementación Técnica

- **Lenguaje:** Python 3.
- **Interfaz Gráfica (GUI):** Se utilizó la biblioteca **Tkinter**, que viene incluida en la librería estándar de Python. Se compone de una ventana principal (**root**), **Labels** para texto, **Buttons** para interacción y un **Canvas** para la gestión de imágenes.
- **Gestión de Imágenes:** Se utiliza la biblioteca **Pillow (PIL)** para abrir, redimensionar y mostrar formatos de imagen no nativos de Tkinter, como **.png**, **.jpg** y **.jpeg**.

- **Persistencia de Datos (Memoria):** Para asegurar que el aprendizaje sea permanente, el estado completo del árbol de conocimiento se guarda en un archivo **conocimiento.json**.
 - Se eligió **JSON** sobre otros métodos (como **pickle**) por ser un formato de texto legible por humanos, seguro y altamente portátil, permitiendo que la base de conocimiento pueda ser inspeccionada o incluso editada manualmente si fuera necesario.

5. Conclusión

Este proyecto implementa exitosamente un Sistema Experto funcional. A través de una interfaz gráfica amigable, el sistema demuestra los principios fundamentales de la IA: una **base de conocimiento** estructurada (reglas y casos), un **motor de inferencia** lógico (encadenamiento hacia adelante) y un **módulo de aprendizaje** dinámico que le permite adquirir conocimiento y mejorar su desempeño con el tiempo.