



MATERIA
SISTEMAS EXPERTOS

“Práctica 3”

Stephanie Vianney Aguila Rivera 22310212

PROF. Mauricio Alejandro Cabrera Arellano

Ingeniería en Mecatrónica

Grupo 7F

26 de Octubre de 2025

Introducción

Este documento presenta el diseño, desarrollo y funcionamiento del sistema experto "Adivina Quién: Edición Bridgerton". El objetivo principal de este proyecto es aplicar los conceptos teóricos de la Inteligencia Artificial, específicamente en el campo de los Sistemas Expertos, para crear una aplicación funcional y didáctica.

El sistema desarrollado simula el clásico juego "Adivina Quién", pero enfocado en el universo de la popular serie "Bridgerton". El programa es capaz de identificar a un personaje en el que el usuario está pensando, basándose en una serie de preguntas de "sí" o "no".

Las características clave de este proyecto son:

- **Interfaz Gráfica de Usuario (GUI):** El sistema cuenta con una interfaz amigable e intuitiva desarrollada en Python con la librería Tkinter.
- **Base de Conocimientos Externa:** El conocimiento del sistema (personajes y atributos) se almacena en un archivo JSON, permitiendo que sea fácil de actualizar y expandir.
- **Motor de Inferencia Dinámico:** El sistema utiliza un motor de inferencia basado en reglas para filtrar a los posibles candidatos.
- **Capacidad de Aprendizaje:** Siguiendo el requisito de la práctica, si el sistema falla en adivinar, es capaz de "aprender" el nuevo personaje y la característica que lo distingue, actualizando su base de conocimientos para futuras partidas.

El Cerebro del Sistema

Arquitectura del Sistema

El sistema se compone de tres partes fundamentales:

1. **Base de Conocimientos (BC):** Es el "libro de texto" del experto. En nuestro caso, es el archivo `basedatos_bridgerton.json`. Este archivo no almacena reglas, sino hechos sobre 16 personajes (Daphne, Anthony, Penelope, etc.). Cada personaje es un conjunto de atributos (hechos), por ejemplo: `{"nombre": "Eloise", "es_intelectual": "si", "desafia_normas": "si"}`.
2. **Motor de Inferencia (MI):** Es el "cerebro" o el "experto" que razona. Es nuestro script `bridgerton_gui.py`. Su trabajo es tomar los hechos de la BC y, usando las respuestas del usuario, inferir una conclusión.
3. **Interfaz de Usuario (IU):** Es la ventana gráfica (GUI) con la que el usuario interactúa, mostrando la bienvenida, las preguntas y los botones.

Representación del Conocimiento: Reglas

Aunque nuestra base de datos almacena *hechos*, el Motor de Inferencia opera usando Reglas de Producción implícitas en el código. Estas reglas siguen la estructura lógica **SI ... ENTONCES**

El motor no tiene las reglas escritas como texto, sino como lógica de programación. La regla principal de filtrado en nuestro sistema es:

- REGLA DE FILTRADO (Ejemplo):
 - **SI** el sistema pregunta "¿Tu personaje es un Bridgerton?"
 - **Y** el usuario responde "Sí"
 - **ENTONCES** el Motor de Inferencia debe **ELIMINAR** de la lista de "posibles" a todos los personajes cuyo atributo **es_bridgerton** sea "no".

Motor de Inferencia: Encadenamiento Hacia Adelante

El requisito de la práctica fue implementar un Encadenamiento Hacia Adelante (Forward Chaining). Este método de inferencia parte de los hechos iniciales (las respuestas del usuario) y avanza aplicando reglas para deducir nuevos hechos, hasta llegar a una conclusión (la adivinanza).

Así funciona en nuestro juego:

1. Estado Inicial: La lista de **posibles** contiene a los 16 personajes de la Base de Conocimientos.
2. Añadir un Hecho: El sistema hace una pregunta (ej. ¿Es hombre?). El usuario responde **Sí**. Este "Sí" es un hecho nuevo.
3. Disparar Regla: El Motor de Inferencia toma este hecho (**es_hombre: "sí"**) y dispara la "Regla de Filtrado".
4. Generar Nuevo Estado: La regla se ejecuta. Todos los personajes femeninos son eliminados de la lista **posibles**. La lista ahora solo contiene a Anthony, Benedict, Colin y Simon.
5. Repetir Ciclo: El sistema busca otra pregunta (ej. ¿Es protagonista?) para diferenciar a los restantes. El ciclo se repite.
6. Meta: El ciclo termina cuando se cumple una de dos condiciones:
 - **len(posibles) == 1** (Solo queda un personaje).
 - **numero_de_preguntas == 5** (Se alcanzó el límite de preguntas).

El sistema siempre avanza "hacia adelante", desde las respuestas iniciales hasta la conclusión final.

Adquisición de Conocimiento: Razonamiento Basado en Casos (CBR)

La función de "aprendizaje" es una aplicación de Razonamiento Basado en Casos (CBR). El sistema aprende de sus errores.

1. El "Caso" Fallido: Un "caso" es la secuencia de 5 preguntas y respuestas que llevaron a una adivinanza incorrecta. (Ej. Adivinó **Daphne** pero era **Francesca**).
2. Revisión del Caso: El sistema pide al usuario que "revise" el caso.
 - El usuario introduce el nombre correcto: **Francesca Bridgerton**.
 - El usuario introduce la pregunta diferenciadora: **¿Tu personaje es conocido por sus talentos artísticos?**
 - El usuario da la respuesta para Francesca: **si**.
3. Retención del Caso: El sistema "retiene" este nuevo conocimiento.
 - Añade el nuevo atributo **es_artista** al diccionario de preguntas.
 - Actualiza el JSON: añade **es_artista: "sí"** a **Francesca** y **es_artista: "no"** (la respuesta opuesta) a **Daphne**.
4. Reutilización: La próxima vez, el sistema podrá usar esta nueva pregunta para resolver casos similares.

Manual de Usuario

Requisitos del Sistema

Para ejecutar este programa, se necesita:

- Un sistema operativo (Windows, macOS, Linux).
- Python 3 instalado.
- La librería **Pillow** de Python.

Guía de Uso

1. Iniciar el Juego: Ejecute el script de Python (usualmente haciendo doble clic o desde la terminal con **python bridgerton_gui.py**).
2. Pantalla de Bienvenida: Se mostrará la pantalla inicial con la imagen de Bridgerton y un texto de bienvenida. Presione el botón "Comenzar a Jugar".
3. Jugar la Partida:
 - El sistema le mostrará una serie de 5 preguntas aleatorias.
 - Piense en un personaje de la base de datos (Daphne, Anthony, Penelope, Kate, Violet, etc.).
 - Responda a cada pregunta haciendo clic en el botón "Sí" o "No".
4. Adivinanza Final: Después de las 5 preguntas (o antes, si el sistema ya está seguro), se abrirá una nueva ventana.
 - Esta ventana mostrará la foto y el nombre del personaje que el sistema cree que es.

- Si es correcto: Presione "Sí, ¡adivinaste!".
- Si es incorrecto: Presione "No".

El Proceso de Aprendizaje

Si usted presiona "No" en la adivinanza final, el sistema activará el modo de aprendizaje:

1. Ventana 1: El sistema preguntará: "¿Cuál era el nombre de tu personaje?". Escriba el nombre (ej. **Lady Whistledown**).
2. Ventana 2: Preguntará: "Escribe una pregunta (sí/no) que diferencie a [Personaje que adivinó] de [Tu personaje]". (ej. ¿Es una escritora secreta?).
3. Ventana 3: Preguntará la respuesta para su personaje: "Y para Lady Whistledown, ¿la respuesta a esa pregunta sería 'sí' o 'no'?". Escriba sí.

¡Listo! El sistema responderá "¡He actualizado mi base de conocimientos!" y la próxima vez podrá preguntar ¿Es una escritora secreta?.

Conclusiones

El desarrollo de este proyecto permitió aplicar exitosamente los fundamentos teóricos de los Sistemas Expertos en una aplicación práctica y funcional.

Se logró construir un Motor de Inferencia capaz de razonar mediante encadenamiento hacia adelante, utilizando una Base de Conocimientos de *hechos* para aplicar *reglas* de filtrado. La implementación de la interfaz gráfica y la gestión de archivos JSON complementaron el núcleo de IA del proyecto.

La característica más relevante es la capacidad de Adquisición de Conocimiento (aprendizaje), que simula el Razonamiento Basado en Casos y permite que el sistema mejore con cada partida fallida, cumpliendo con todos los requisitos de la práctica.

https://github.com/vianneyaguila/SE_AguilaRivera_7F/tree/main/Adivina%20quien

[Akinator Bridgerton by vianneyaguila](#)