Clara Páginas, Tu chatbot de recomendación avanzado

Reconocimiento Automático del Habla

Universitat Politècnica de València

Iñaki Diez Lambies

Tabla de contenido

[1. Introducción 2](#_Toc157514876)

[2. Objetivos del sistema 3](#_Toc157514877)

[3. Diseño 4](#_Toc157514878)

[4. Implementación 5](#_Toc157514879)

[4.1. Configuration 5](#_Toc157514880)

[4.2. Intents 7](#_Toc157514881)

[4.3. Utters 8](#_Toc157514882)

[4.4. Actions 9](#_Toc157514883)

[BooksDB 10](#_Toc157514884)

[4.5. Slots 11](#_Toc157514885)

[4.6. Rules 11](#_Toc157514886)

[4.7. Stories 13](#_Toc157514887)

[5. Resultados 14](#_Toc157514888)

[6. Conclusiones 17](#_Toc157514889)

# Introducción

En el umbral de la era digital, la interacción humana con las máquinas ha avanzado significativamente, dando lugar a la emergencia de *chatbots* inteligentes. Este trabajo se centra en la elaboración de un *chatbot* utilizando Rasa, una plataforma líder en la creación de *chatbots* basados en inteligencia artificial y aprendizaje automático. *Rasa* destaca por su capacidad para construir *chatbots* que no solo responden a consultas predefinidas, sino que también aprenden y se adaptan a las necesidades de los usuarios a través de interacciones enriquecidas.

El propósito de este documento es presentar el desarrollo y la implementación de un *chatbot* específico diseñado para un contexto determinado. A través de este proyecto, se busca explorar las capacidades de Rasa en la creación de un *chatbot* que no solo sea funcional y eficiente, sino también interactivo y capaz de realizar tareas a través de acciones personalizadas. Este trabajo abarcará desde la fase de diseño hasta la implementación, incluyendo la creación de *intents*, *utters*, *actions*, *slots*, *rules* y *stories*, elementos esenciales en la arquitectura de Rasa.

El *chatbot* desarrollado en este estudio tiene como objetivo demostrar cómo las tecnologías avanzadas de *chatbots* pueden ser aplicadas efectivamente para mejorar la interacción usuario-máquina, ofreciendo una experiencia de usuario más rica y personalizada. Al final del trabajo, se presentarán los resultados obtenidos, los desafíos enfrentados y las conclusiones derivadas de la experiencia de construir y operar un *chatbot* con *Rasa*, proporcionando así una visión integral de lo que implica el desarrollo de estas soluciones tecnológicas avanzadas en el ámbito actual.

# Objetivos del sistema

El *chatbot* diseñado no solo ejecuta funciones básicas como saludar, presentarse y despedirse, sino también para involucrarse en conversaciones más complejas y personalizadas, reaccionando ante las emociones del usuario y sosteniendo diálogos sobre temas variados como el clima, bromas, hobbies y libros favoritos. Además, una de sus funcionalidades clave es la capacidad de recomendar libros, lo que añade un valor educativo y cultural a la interacción.

Los objetivos del *chatbot* van más allá de la mera automatización de respuestas. En primer lugar, se busca establecer una comunicación fluida y natural, donde el *chatbot* no solo entienda y responda a las solicitudes del usuario, sino que también reconozca y reaccione adecuadamente a las emociones expresadas. Esto implica un nivel sofisticado de procesamiento del lenguaje natural y aprendizaje automático, permitiendo un aumento en la fluidez en las conversaciones generadas.

Otro objetivo importante es mantener al usuario comprometido y satisfecho. Esto se logra no solo a través de respuestas precisas y relevantes, sino también ofreciendo conversaciones interesantes y entretenidas. Hablar sobre temas como el tiempo, hobbies y libros favoritos ayuda a crear una experiencia más rica y personalizada. Al mismo tiempo, la capacidad de recomendar libros abre una ventana a experiencias educativas y culturales, lo que puede ser particularmente valioso en contextos donde se busca fomentar la lectura o proporcionar recomendaciones personalizadas.

Además, este *chatbot* está diseñado para ser intuitivo y fácil de usar, con un enfoque en la usabilidad para asegurar que los usuarios de todas las edades y niveles de habilidad tecnológica puedan interactuar con él de manera efectiva. La accesibilidad y la inclusión son también objetivos clave, garantizando que este sea útil para un amplio espectro de usuarios.

Es por ello por lo que los objetivos de este proyecto se centran en la creación de una herramienta comunicativa avanzada, capaz de ofrecer interacciones enriquecedoras, educativas y emocionalmente inteligentes. Se busca que el *chatbot* resultante no solo sea un medio para obtener respuestas rápidas, sino una plataforma interactiva que mejore la experiencia general del usuario, proporcionando un servicio que sea a la vez informativo, entretenido y accesible para todos.

# Diseño

El diseño se centra en proporcionar una experiencia de usuario interactiva y personalizada. Este sistema se ha diseñado para manejar una variedad de interacciones humanas, desde conversaciones cotidianas hasta recomendaciones específicas de libros, lo que refleja una estructura compleja y bien pensada en su configuración.

En el núcleo del diseño se encuentran los componentes de *Rasa*: *intents*, *entities*, *slots*, *actions*, *responses*, y *stories*. Los *intents* capturan las intenciones del usuario, como saludar, despedirse, expresar emociones o solicitar recomendaciones de libros. Cada *intent* se define con ejemplos variados para garantizar un reconocimiento preciso de las intenciones del usuario en distintos contextos y formas de expresión. Las *entities*, como el género de un libro, se utilizan para extraer información específica de los mensajes del usuario, lo que permite al sistema proporcionar respuestas más personalizadas y relevantes.

Los *slots* actúan como contenedores de datos que el *chatbot* utiliza para almacenar y acceder a la información recopilada durante la conversación, lo que facilita la personalización de las interacciones. Por ejemplo, un *slot* puede almacenar el género de libro preferido por el usuario, lo que permite ofrecer recomendaciones más precisas.

Las *actions* son funciones que el *chatbot* ejecuta en respuesta a los *intents* del usuario. Estas pueden variar desde enviar un mensaje sencillo hasta realizar operaciones más complejas como consultar una base de datos de libros para hacer una recomendación. Las *responses* definen los mensajes específicos que se utilizarán para interactuar con el usuario, y son clave para garantizar que las conversaciones sean fluidas y naturales.

Por último, las *stories* son una representación de las posibles trayectorias de conversación, que guían al sistema sobre cómo responder en diferentes situaciones. Esto incluye desde un saludo inicial hasta el manejo de conversaciones más largas y complejas, incluyendo la reacción a diferentes emociones del usuario y la realización de recomendaciones de libros basadas en las preferencias expresadas.

En conjunto, estos componentes forman un sistema robusto y flexible, capaz de manejar una amplia gama de interacciones de manera efectiva. La arquitectura del *chatbot* ha sido diseñada para ser escalable y fácilmente adaptable a nuevas funcionalidades y mejoras, lo que garantiza su relevancia y utilidad a largo plazo. La inclusión de un sistema de *fallback* y la capacidad de reformular peticiones proporciona un mecanismo de seguridad para manejar situaciones en las que el sistema no comprenda completamente al usuario, asegurando así una experiencia de usuario coherente y satisfactoria.

Este diseño detallado refleja una comprensión profunda de las necesidades de los usuarios y las posibilidades de la tecnología actual, lo que lo posiciona como una herramienta valiosa en el ámbito de la interacción digital avanzada.

# Implementación

La implementación es un proceso que implica varios pasos clave, cada uno con su importancia y complejidad. En esta sección, se explorarán los componentes críticos que constituyen el corazón de la funcionalidad del *chatbot*. Estos componentes incluyen la configuración del sistema, la definición de intenciones del usuario, las respuestas, las acciones que puede realizar, la gestión de información a través de los slots, las reglas que guían el comportamiento del sistema y las historias que definen los flujos de conversación.

Cada uno de estos elementos juega un papel fundamental en la creación de un *chatbot* interactivo y eficaz. La configuración inicial establece las bases del procesamiento del lenguaje y la toma de decisiones. Los *intents* y *utters* forman la base de la interacción entre el usuario y el sistema, permitiendo una comunicación fluida. Las *actions* y *slots* son esenciales para manejar y almacenar información relevante durante las conversaciones. Por otro lado, las *rules* y *stories* aseguran que el sistema responda de manera coherente y lógica.

Avanzando en esta sección, se detallará cómo cada uno de estos componentes se implementa para construir un *chatbot* que no solo comprenda y responda a las solicitudes de los usuarios, sino que también ofrezca recomendaciones personalizadas y mantenga conversaciones significativas. La implementación efectiva de estos elementos es crucial para lograr que ofrezca una experiencia de usuario rica y satisfactoria.

## Configuration

La configuración del *chatbot*, se basa en una serie de componentes que definen su funcionamiento y capacidad de procesamiento del lenguaje natural. Cada componente seleccionado tiene una función específica que contribuye al rendimiento general del sistema.

Los *pipelines* en *Rasa* son esenciales para el procesamiento del lenguaje natural (NLP). Actúan como una cadena de componentes que trabajan en secuencia para procesar y entender el texto del usuario. Esta cadena incluye pasos como tokenización, extracción de características, clasificación, analizar la gramática, entender el significado del texto o identificar entidades relevantes. Una configuración adecuada es crucial para la eficacia del *chatbot* en comprender y responder a las interacciones de los usuarios de manera precisa y natural como la que podemos encontrar en la Ilustración 1.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 1 - Implementación de los pipelines

El *WhitespaceTokenizer* divide el texto de entrada en palabras individuales, lo cual es esencial para el análisis posterior del lenguaje. Seguidamente, el *RegexFeaturizer* utiliza expresiones regulares para identificar patrones de texto, lo que ayuda a mejorar la comprensión de las intenciones del usuario. Por otro lado, el *LexicalSyntacticFeaturizer* analiza las características sintácticas y léxicas del texto, proporcionando un contexto más rico para la interpretación de las consultas.

Los *CountVectorsFeaturizers* transforman el texto en vectores numéricos que representan la frecuencia de las palabras y caracteres. Esta representación es fundamental para que los algoritmos de aprendizaje automático puedan procesar el texto. El uso de n-gramas (*min\_ngram* y *max\_ngram*) permite capturar la secuencia de palabras y caracteres, proporcionando un contexto más amplio.

El *DIETClassifier* es un componente versátil que maneja tanto la clasificación de intenciones como la extracción de entidades. Su entrenamiento con un número significativo de épocas garantiza que el modelo aprenda eficazmente de los datos de entrenamiento. La restricción de similitudes (con la opción *constrain\_similarities*) asegura que las predicciones sean más precisas y diferenciadas.

En siguiente lugar, el *EntitySynonymMapper* se utiliza para mapear sinónimos de entidades, lo que permite al *chatbot* reconocer diferentes maneras en que los usuarios pueden referirse a la misma cosa. Seguidamente, el *ResponseSelector* es clave para seleccionar respuestas predefinidas, especialmente útil en situaciones de preguntas frecuentes o respuestas estándar.

El *FallbackClassifier* proporciona un mecanismo de seguridad para situaciones en las que el sistema no está seguro de la intención del usuario. Este clasificador se activa cuando la confianza en la predicción de la intención es baja (por debajo del umbral establecido), ofreciendo así una respuesta de *fallback* o solicitando una aclaración.

Por otro lado, las *policies* determinan cómo el *chatbot* decide la próxima acción a tomar en respuesta a la entrada del usuario. Son algoritmos que analizan el estado actual de la conversación y predicen la acción más adecuada. En la Ilustración 2 podemos ver como estas políticas pueden basarse en reglas predefinidas, memorización de conversaciones pasadas, o modelos de aprendizaje profundo que entienden el contexto de la conversación. La configuración de estas influye en la fluidez, relevancia y coherencia de las conversaciones, asegurando que responda de manera lógica y consistente.

Texto

Descripción generada automáticamente

Ilustración 2 - Implementación de las polices

En primer lugar, la *MemoizationPolicy* recuerda las conversaciones pasadas para hacer predicciones más precisas. La *TEDPolicy* (*Transformer Embedding Dialogue*) utiliza un modelo de aprendizaje profundo para entender el contexto de la conversación y predecir la siguiente acción del sistema. Por último, la *RulePolicy* define reglas específicas para manejar situaciones predefinidas y garantizar respuestas coherentes.

Cada uno de estos componentes ha sido seleccionado y configurado con el objetivo de crear un chatbot que sea eficiente, preciso y capaz de manejar una variedad de interacciones de usuario de manera inteligente y natural. La configuración general refleja un equilibrio entre la precisión del procesamiento del lenguaje, la adaptabilidad a distintos contextos de conversación, y la capacidad de proporcionar respuestas útiles y pertinentes.

## Intents

La implementación de los *intents* en el *chatbot* es un aspecto que determina su capacidad para entender y responder adecuadamente a las solicitudes de los usuarios. Estos representan las diferentes intenciones o propósitos detrás de los mensajes de los usuarios. En este sistema, se han definido varios *intents* para abarcar un amplio rango de posibles interacciones.

Las intenciones de saludo (*greet*) y despedida (*farewell*) permiten al *chatbot* iniciar y cerrar conversaciones de manera natural. El *intent* de agradecimiento (*thanks*) reconoce y responde a expresiones de gratitud, mientras que los de emociones como felicidad (*happy*) y tristeza (*sad*) permiten al este interactuar de manera más empática y personalizada con los usuarios. Podemos ver estos en la Ilustración 3.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 3 - Implementación de los intents principales de interacción

Además, se han incluido intenciones para charlas informales o *chitchat*, como preguntar cómo está el usuario (*chitchat.ask\_how*), hablar del clima (*chitchat.weather*), discutir sobre hobbies (*chitchat.hobbies*), compartir libros favoritos (*chitchat.favorites*), e incluso contar chistes (*chitchat.jokes*). Estos *intents* enriquecen la experiencia del usuario, haciendo que la interacción con el sistema sea más variada y agradable.

Finalmente, los *intents* relacionados con la recomendación de libros (*request\_book\_recommendation* e *inform\_book\_genre*) son fundamentales para la funcionalidad principal del *chatbot*. Estos permiten que *Clara Páginas* ofrezca recomendaciones personalizadas basadas en las preferencias de género literario del usuario.

Cada uno de estos *intents* está acompañado de ejemplos concretos que ayudan al sistema a comprender mejor la variedad de formas en que los usuarios pueden expresar estas intenciones. Esta amplia gama de intenciones y ejemplos es esencial para construir un *chatbot* versátil y capaz de manejar diferentes tipos de interacciones de manera efectiva.

## Utters

Las respuestas o *utters* juegan un papel fundamental en la interacción del *chatbot* con los usuarios. Estas respuestas están predefinidas para cada *intent* identificado y permiten al sistema comunicarse de manera efectiva y natural.

La respuesta de presentación, *utter\_presentation*, introduce al *chatbot* como una bibliotecaria digital, estableciendo el contexto de su función y ofreciendo una gama de temas sobre los que el usuario puede preguntar. Las respuestas de saludo, *utter\_greet*, están diseñadas para iniciar la interacción y preguntar cómo puede ayudar, fomentando así una conversación fluida.

Las respuestas de *rephrase*, *utter\_ask\_rephrase*, y las respuestas predeterminadas, *utter\_default*, son cruciales para manejar situaciones donde el sistema no comprende completamente la entrada del usuario. Estas respuestas piden amablemente que el usuario reformule o repita su mensaje, manteniendo así la interacción y evitando confusiones.

Por otro lado, las respuestas emocionales como *utter\_happy* y *utter\_sad* que podemos ver en la Ilustración 4 reflejan la capacidad del *chatbot* para reconocer y responder a los estados emocionales del usuario, lo cual es esencial para crear una experiencia de usuario empática y personalizada.

Texto

Descripción generada automáticamente

Ilustración 4 - Implementación de algunos utters

En cuanto a la charla informal, se incluyen respuestas relacionadas con preguntar cómo está el usuario, hablar sobre el clima, hobbies, libros favoritos y contar chistes, lo que añade variedad y profundidad a las conversaciones. Estas respuestas no solo mantienen al usuario comprometido, sino que también reflejan la versatilidad del sistema para manejar diferentes tipos de diálogos.

Finalmente, la respuesta *utter\_ask\_book\_genre* invita a los usuarios a especificar sus preferencias de género literario, lo que es un paso crucial para proporcionar recomendaciones de libros personalizadas.

En conjunto, estas respuestas forman una base sólida para una amplia gama de interacciones posibles, permitiendo que *Clara Páginas* se comunique de manera efectiva, mantenga el interés del usuario y proporcione una experiencia de usuario rica y dinámica.

## Actions

En la implementación de las acciones de *Clara Páginas*, se destacan dos acciones principales: *ActionTwoStageFallback* y *ActionRecommendBook*. Además, se integra un componente adicional, *BooksDB*, para la gestión de recomendaciones de libros.

La *ActionTwoStageFallback* es una acción de recuperación que se activa cuando el *chatbot* no está seguro de la intención del usuario. Esta acción sugiere una posible intención alternativa basada en el ranking de intenciones y pide al usuario que aclare o reformule su petición. Este mecanismo mejora la precisión y relevancia de las respuestas de este.

Por otro lado, la *ActionRecommendBook* es fundamental para la funcionalidad de recomendación de libros, la podemos ver implementada en la Ilustración 5. Esta acción utiliza el slot *book\_genre* para determinar el género de libro preferido del usuario. Si el género está claro, el *chatbot* accede a *BooksDB* para encontrar libros relevantes.

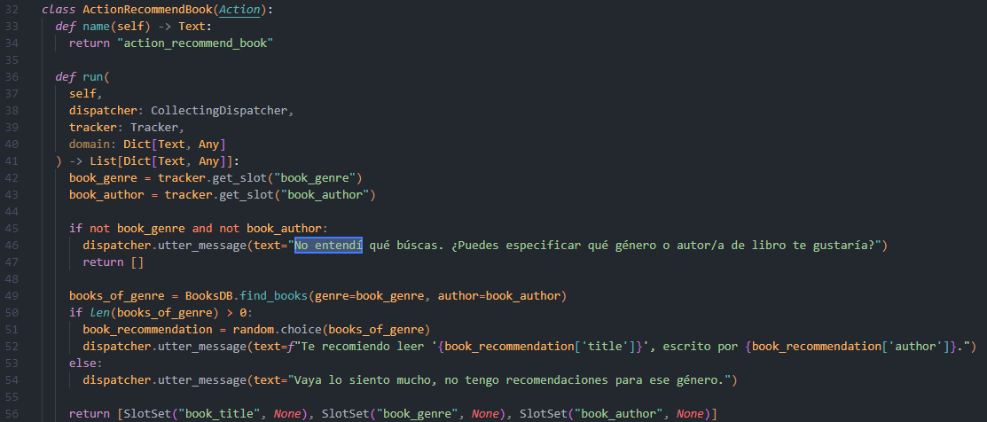


Ilustración 5 - Implementación de ASctionRecommendBook

También encontramos la acción *ActionRetrieveBookDescription* que nos permite conseguir la descripción de un libro del cual se nos brinde el título. Es muy importante en la interacción con el usuario disponer de esta acción para que este consiga una recomendación lo más informada posible, sabiendo en profundidad la temática del libro a recomendar.

### BooksDB

La implementación de *BooksDB* utiliza una API de libros externa, lo cual expande significativamente el conocimiento y la capacidad del *chatbot* para recomendar libros. En lugar de depender de una base de datos interna con capacidad limitada, el *chatbot* puede acceder a una fuente rica y diversa de información de libros. Esto le permite ofrecer recomendaciones basadas en una variedad más amplia de títulos, autores y géneros, lo que mejora la personalización y relevancia para el usuario.

Texto

Descripción generada automáticamente

Ilustración 6 - Implementación de BooksDB

La clase *BooksDB* cuenta con métodos que realizan consultas directamente a la API de *Google Books* utilizando parámetros opcionales como título, género y autor. La función *find\_books* construye una URL de búsqueda dinámica que se ajusta según los parámetros proporcionados. La flexibilidad de esta función permite al *chatbot* realizar búsquedas más precisas y ajustarse mejor a las necesidades de los usuarios.

El método *recommend\_random\_book* añade un elemento de descubrimiento y diversión, seleccionando al azar un género de una lista predefinida y luego recomendando un libro de ese género. Esto no solo introduce variedad en las recomendaciones, sino que también aprovecha la naturaleza impredecible de la selección para posiblemente presentar a los usuarios libros en los que no habrían pensado ellos mismos.

El sistema, al utilizar esta nueva implementación, puede ofrecer una experiencia de usuario mejorada gracias a su acceso en tiempo real a una base de datos de libros mucho más extensa y actualizada, y su habilidad para proporcionar sugerencias detalladas que incluyen título, autor y descripción de los libros. Además, al manejar errores y verificar el código de estado de la respuesta de la API, el *chatbot* puede manejar con gracia situaciones donde la API no está disponible o la consulta no produce resultados, garantizando así una interacción fluida y sin frustraciones para el usuario.

## Slots

Los *slots* son utilizados para almacenar y manejar información específica durante la conversación. En este caso, el slot *book\_genre* juega un rol crucial. Está configurado para capturar el género del libro que el usuario desea explorar. Este *slot* es de tipo texto y está configurado para no influir directamente en el flujo de la conversación, lo que significa que su valor no altera el curso de la interacción del *chatbot* con el usuario. En la Ilustración 7 vemos como está implementado.

Texto

Descripción generada automáticamente

Ilustración 7 - Implementación de los slots y las actions

La información recogida en el *slot book\_genre* se extrae directamente de las entidades identificadas en los mensajes del usuario, específicamente cuando el *intent inform\_book\_genre* es reconocido. Así, cuando un usuario expresa su preferencia por un género literario, el sistema utiliza esta información para buscar recomendaciones de libros pertinentes. La habilidad de manejar y almacenar esta información esencial permite que el sistema realice recomendaciones de libros de manera más precisa y adaptada a los gustos individuales del usuario.

## Rules

Las reglas definidas para *Clara Páginas* juegan un papel esencial en estructurar cómo el *chatbot* responde a diferentes intenciones de los usuarios. Estas reglas aseguran respuestas coherentes y pertinentes, alineadas con las expectativas de los usuarios. Las podemos ver implementadas en la Ilustración 8.

La regla de saludo estándar establece que, cuando el usuario expresa un saludo (*intent greet*), el sistema primero se presenta (*utter\_presentation*) y luego pregunta cómo puede ayudar (*utter\_greet*). Esta secuencia crea una interacción inicial amigable y acogedora.

La implementación del *Two-Stage Fallback* es una regla importante para manejar casos donde el *chatbot* no está seguro de la intención del usuario. En estos casos, se activa un proceso de dos pasos para aclarar la intención del usuario, mejorando así la precisión en la comunicación.

Texto

Descripción generada automáticamente

Ilustración 8 - Implementación de la rules

La regla de despedida estándar asegura que, cuando el usuario indica la intención de terminar la conversación (*farewell*), el sistema responde de manera cortés y apropiada con *utter\_farewell*, cerrando la interacción de manera amigable.

Además, hay reglas específicas para reacciones emocionales. Por ejemplo, cuando el usuario expresa felicidad (*happy*), el *chatbot* responde de manera positiva y alentadora (*utter\_happy*). De forma similar, cuando el usuario está triste (*sad*), ofrece una respuesta empática (*utter\_sad*).

Las reacciones a los agradecimientos y disculpas también están definidas. Cuando el usuario agradece (*thanks*), el *chatbot* responde con *utter\_thanks*, y cuando se disculpa (*apologize*), el *chatbot* ofrece una respuesta conciliadora con *utter\_apologize*.

Estas reglas demuestran la capacidad del sistema para manejar una variedad de situaciones comunicativas, respondiendo de manera apropiada y manteniendo una experiencia de usuario coherente y satisfactoria. Cada regla contribuye a la fluidez de la conversación y refuerza la percepción del *chatbot* como un asistente virtual competente y sensible.

## Stories

Los *stories* son un componente fundamental que guía al *chatbot* en cómo responder en diferentes situaciones de diálogo. Actúan como ejemplos de conversaciones que enseñan al *chatbot* los flujos de interacción posibles. Cada *story* describe una secuencia de pasos que incluyen intenciones del usuario (*intents*) y acciones del *chatbot* (*actions*).

En *Clara Páginas*, se han definido diversas *stories* para cubrir un amplio rango de interacciones. Por ejemplo, la *story* titulada *happy greeting* enseña al sistema cómo responder a un saludo seguido de una expresión de felicidad, mientras que *sad greeting* muestra la respuesta adecuada a un saludo seguido de una expresión de tristeza. Estas *stories* ayudan al *chatbot* a manejar respuestas emocionales de manera apropiada.

Imagen de la pantalla de un celular de un mensaje en letras negras

Descripción generada automáticamente con confianza baja

Ilustración 9 - Implementación de algunos stories

Además, hay *stories* que combinan saludos con *chitchat* sobre el clima, hobbies y libros favoritos, lo que permite al *chatbot* participar en conversaciones más informales y variadas. Las *stories* relacionadas con la recomendación de libros, como *book recommendation - mystery*, guían al *chatbot* en el proceso de ofrecer sugerencias de libros basadas en géneros específicos como las que vemos en la Ilustración 9.

Cada *story* contribuye a la formación del sistema, permitiéndole aprender patrones de interacción y responder de manera coherente y contextual. Al incluir una variedad de *stories*, el *chatbot* se vuelve más versátil y capaz de manejar una amplia gama de conversaciones, lo que resulta en una experiencia de usuario más rica y satisfactoria.

# Resultados

En esta sección, la evaluación del *chatbot* se centra en dos áreas principales: el análisis de interacciones y los desafíos técnicos encontrados. A través del análisis de interacciones específicas, que se observan en las capturas de pantalla proporcionadas, podemos evaluar la capacidad del sistema para gestionar conversaciones en tiempo real y su habilidad para proporcionar respuestas coherentes y contextuales.

En el ejemplo de chat de la Ilustración 10, vemos que *Clara Páginas* maneja adecuadamente el saludo inicial y la presentación, guiando la conversación con naturalidad y ofreciendo ayuda de forma proactiva. Cuando el usuario inicia un diálogo casual preguntando cómo está el *chatbot* y luego expresando felicidad, el *chatbot* responde con entusiasmo y mantiene la conversación en un tono positivo. Además, el sistema maneja con éxito una solicitud de chiste, demostrando su habilidad para participar en *chitchat* ligero y entretenimiento, aspectos que son cruciales para una experiencia de usuario agradable.

Texto

Descripción generada automáticamente

Ilustración 10 - Conversación de ejemplo número 1

El segundo ejemplo de la Ilustración 11 muestra la funcionalidad central del *chatbot*: la recomendación de libros. El sistema pregunta primero sobre las preferencias del usuario y luego proporciona una sugerencia acorde al género de interés. Frente a la insatisfacción del usuario con la primera recomendación, el *chatbot* realiza una segunda sugerencia sin problemas, lo que indica una buena gestión de slots y una base de datos de libros bien integrada.

Texto

Descripción generada automáticamente

Ilustración 11 - Conversación de ejemplo número 2

En la conversación mostrada en la Ilustración 12, el sistema desempeña eficazmente su papel como asistente virtual centrado en la recomendación de libros. El *chatbot* inicia la conversación con un saludo amistoso y se presenta como una bibliotecaria digital, lo cual establece el contexto y la funcionalidad principal del sistema. A continuación, ofrece su asistencia en la búsqueda de libros y está abierto a otras preguntas, lo que demuestra versatilidad y una disposición a ayudar en un rango más amplio de temas.

Cuando el usuario consulta sobre el estado, se responde de manera programada pero amigable, manteniendo un tono de conversación ligero y agradable, lo cual es esencial para una interacción placentera y natural.

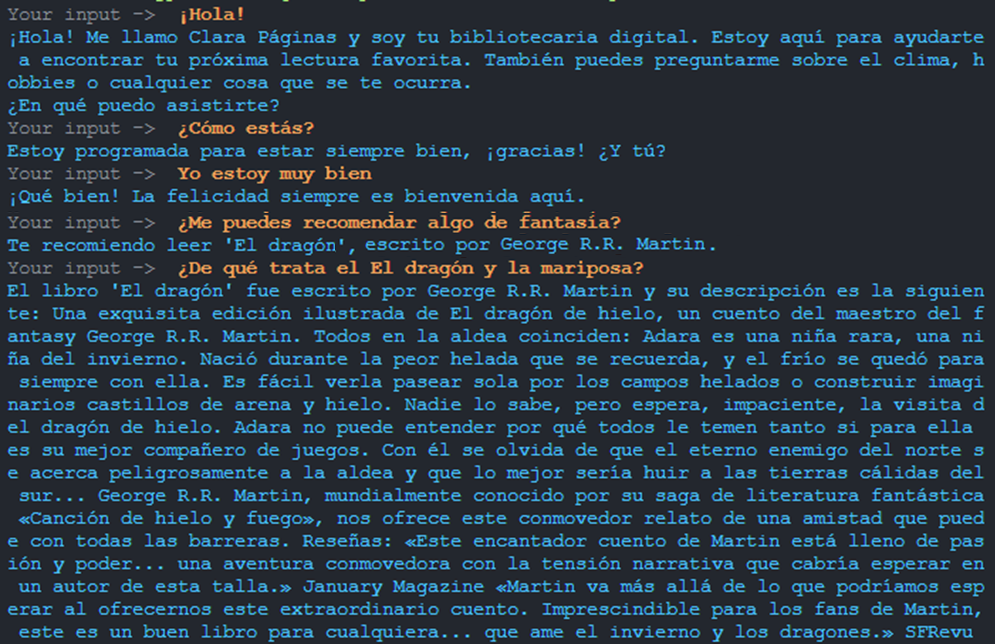


Ilustración 12 - Conversación de ejemplo número 3

Al pasar a la funcionalidad principal, el *chatbot* maneja con destreza la solicitud de recomendación de un libro de fantasía. Responde con una recomendación específica y, ante una nueva solicitud del usuario, ofrece información detallada sobre otro libro, "El dragón”. La capacidad del chatbot para proporcionar no solo el título y el autor, sino también una sinopsis detallada y una cita de reseñas, enriquece la recomendación y le proporciona al usuario un contexto más completo para tomar una decisión informada.

Además, al ofrecer alternativas cuando el usuario solicita una recomendación adicional, el chatbot muestra flexibilidad y comprensión de las preferencias del usuario, lo cual es crucial para un servicio de recomendación personalizado. La transición suave entre recomendaciones sugiere que el chatbot tiene acceso a una base de datos bien estructurada y puede manejar consultas complejas con facilidad.

Por último, en la Ilustración 13, vemos otra interacción entre el usuario y el *chatbot* Clara Páginas. La conversación comienza con el sistema saludando y ofreciendo su asistencia, lo que establece un tono amable y servicial desde el comienzo. Seguidamente, el usuario solicita una recomendación de un libro de J.K. Rowling, a lo que el *chatbot* responde recomendando "El cerdito de Navidad". Este comportamiento indica que el sistema tiene la capacidad de buscar en su base de datos por autor, lo cual es una característica valiosa para los usuarios interesados en obras de autores específicos. Sin embargo, el usuario no está satisfecho con la recomendación y lo expresa claramente.

El *chatbot* responde a la insatisfacción del usuario con una disculpa cortés y la disposición a ofrecer otra recomendación, lo cual es un buen ejemplo de manejo de errores en la experiencia del usuario. Esta habilidad de recuperación es crucial para mantener la satisfacción del usuario y el flujo de la conversación.

Texto

Descripción generada automáticamente

Ilustración 13 - Conversación de ejemplo número 4

Posteriormente, cuando el usuario pide una recomendación de un libro de fantasía sin especificar un autor, Clara Páginas recomienda "El dragón y la mariposa" de Michael Ende. Esto demuestra que el *chatbot* puede manejar solicitudes basadas tanto en autores específicos como en géneros literarios.

La conversación muestra cómo se puede ofrecer recomendaciones personalizadas basadas en solicitudes específicas y adaptarse a las preferencias cambiantes del usuario. Además, el chatbot muestra cortesía y adaptabilidad, lo cual son cualidades deseables para un asistente virtual enfocado en la recomendación de libros.

Sin embargo, estos ejemplos también revelan desafíos técnicos. Por ejemplo, el proceso de *fallback* podría mejorarse para manejar de forma más efectiva las situaciones en las que el usuario no está satisfecho con la respuesta. También, hay un reto en la interpretación del lenguaje natural, especialmente en matices y preferencias personales que pueden no estar explícitamente declaradas por el usuario.

Además, la capacidad del *chatbot* para manejar múltiples rondas de interacción sin perder el contexto es fundamental. Esto puede requerir una mejora continua del modelo de diálogo, refinamiento de la lógica de las *stories* y posiblemente la implementación de un manejo de estado más sofisticado para mantener conversaciones coherentes y fluidas a lo largo de interacciones más largas y complejas.

En resumen, *Clara Páginas* demuestra un rendimiento prometedor en términos de interacción y recomendaciones, pero como todo sistema de este tipo, requiere un ajuste continuo y optimización para manejar la variedad de expresiones humanas y satisfacer las expectativas de los usuarios de manera consistente. La retroalimentación continua de los usuarios y el análisis detallado de las interacciones, como se muestra en las capturas de pantalla, serán vitales para el perfeccionamiento continuo del *chatbot*.

# Conclusiones

En conclusión, el desarrollo y la implementación del *chatbot* *Clara Páginas* han demostrado ser un proyecto exitoso y revelador. La capacidad de conducir conversaciones fluidas, gestionar respuestas emocionales y proporcionar recomendaciones de libros personalizadas son indicativos de un diseño sofisticado y una ejecución técnica competente. Los ejemplos de interacciones analizados ilustran la habilidad del *chatbot* para mantener diálogos coherentes y significativos, reflejando una comprensión efectiva del lenguaje natural y las expectativas de los usuarios.

Sin embargo, los desafíos técnicos identificados subrayan la necesidad de una mejora continua, particularmente en la gestión de contextos complejos y en la flexibilidad de la conversación. El futuro del *chatbot* se ve prometedor, con oportunidades para refinamientos en el procesamiento del lenguaje y personalización más profunda.

Este proyecto subraya la importancia del aprendizaje continuo y la adaptación en el campo de los *chatbots*, donde la retroalimentación del usuario y el análisis detallado son cruciales para la evolución del sistema. Con un compromiso continuo con la mejora, *Clara Páginas* puede evolucionar aún más para convertirse en un asistente virtual indispensable para amantes de los libros y usuarios que buscan una interacción enriquecedora y humana.