

Clase 3

Plataformas de Desarrollo de Chatbots

Por Gabriel Pierobon

25 de marzo de 2025

Índice

| | |
|---|-----------|
| 1. Objetivos de la Clase | 3 |
| 2. Plataformas Tradicionales de Chatbots | 3 |
| 2.1. Relevancia en la Era de los LLMs | 3 |
| 2.2. El lugar de Dialogflow hoy | 3 |
| 3. Evolución de Herramientas de Desarrollo de IA Bots | 3 |
| 4. Frameworks de IA Bots | 4 |
| 4.1. Cómo crear un bot sencillo con frameworks de IA Bots | 4 |
| 4.1.1. Pasos Generales para Desarrollar un Bot | 4 |
| 4.2. Principios Generales | 5 |
| 5. Flujo Conversacional | 5 |
| 5.1. ¿Qué es un Flujo Conversacional? | 5 |
| 5.2. Componentes de un Flujo Conversacional | 5 |
| 5.3. Ejemplos de Flujos Conversacionales | 6 |
| 5.4. Estructura de Flujos Conversacionales | 6 |
| 5.4.1. Estructura básica de un flujo conversacional | 6 |
| 5.4.2. Componentes de un flujo conversacional en acción | 7 |
| 5.4.3. Múltiples rutas de diálogos | 7 |
| 5.4.4. Manejo de errores y entradas inesperadas | 7 |
| 5.4.5. Flujo conversacional natural y fluido | 7 |
| 6. Trabajando con Intenciones | 8 |
| 6.1. ¿Qué son las Intenciones? | 8 |
| 6.2. Cómo Trabajar con Intenciones | 8 |
| 6.2.1. Identificación de Intenciones | 8 |
| 6.2.2. Mapeo de Intenciones a Acciones | 8 |
| 6.2.3. Manejo de Múltiples Intenciones | 9 |
| 6.3. Mejores Prácticas | 9 |
| 6.4. Conclusión sobre Intenciones | 10 |
| 7. Trabajando con Entidades | 10 |
| 7.1. ¿Qué son las Entidades? | 10 |
| 7.2. Cómo Trabajar con Entidades | 10 |
| 7.2.1. Identificación de Entidades | 10 |
| 7.2.2. Tipos de Entidades | 11 |
| 7.3. Mejores Prácticas | 11 |
| 7.4. Uso de Entidades en el Flujo Conversacional | 12 |
| 7.5. Conclusión sobre Entidades | 12 |
| 8. Trabajando con Contexto | 12 |
| 8.1. ¿Qué es el Contexto de la Conversación? | 12 |
| 8.2. Cómo Trabajar con el Contexto de la Conversación | 13 |
| 8.2.1. Mantenimiento del Contexto | 13 |
| 8.2.2. Tipos de Contexto | 13 |
| 8.3. Mejores Prácticas | 13 |

| | |
|--|-----------|
| 8.4. Uso del Contexto en el Flujo Conversacional | 14 |
| 8.5. Conclusión sobre Contexto | 14 |
| 9. Diseño de Flujos Conversacionales | 14 |
| 9.1. Mejores Prácticas | 14 |
| 9.2. Conclusiones sobre Flujos Conversacionales | 15 |
| 10.Fulfillments | 15 |
| 10.1. ¿Qué es un Fulfillment en un Chatbot? | 15 |
| 10.2. Explicación paso a paso | 16 |
| 11.Dialogflow de Google | 16 |
| 11.1. Implementación y Escalabilidad de Agentes Conversacionales | 16 |
| 11.2. Historia | 17 |
| 11.3. Características | 17 |
| 11.4. Arquitectura | 17 |
| 11.5. Ventajas y Desventajas | 17 |
| 11.6. Cómo trabajar con Dialogflow | 17 |
| 12.Microsoft Bot Framework | 17 |
| 12.1. Historia y Evolución | 17 |
| 12.2. Características Principales | 17 |
| 12.3. Arquitectura Básica | 18 |
| 12.4. Cómo trabajar con Microsoft Bot Framework | 18 |
| 13.Conclusiones Generales | 18 |
| 14.Referencias y Recursos Adicionales | 18 |

1. Objetivos de la Clase

En esta clase exploraremos:

1. Teoría: plataformas de desarrollo de chatbots
2. Por qué siguen siendo relevantes en la era actual
3. Demostración práctica: Google Dialogflow

2. Plataformas Tradicionales de Chatbots

2.1. Relevancia en la Era de los LLMs

Antes de que los modelos de lenguaje de gran escala (LLMs) dominaran la escena de la IA conversacional, plataformas como Dialogflow establecieron las bases para la interacción humano-máquina estructurada que continúa siendo esencial en muchos contextos empresariales.

Fortalezas de las plataformas tradicionales

- **Control preciso:** Flujos de conversación predecibles y resultados consistentes, críticos para procesos empresariales donde la exactitud es prioritaria
- **Integración empresarial:** Conexión directa con sistemas de negocio, CRMs y bases de datos para ejecutar acciones concretas
- **Eficiencia de recursos:** Soluciones livianas que no requieren la infraestructura masiva de los LLMs
- **Privacidad y cumplimiento:** Mayor control sobre los datos procesados, facilitando el cumplimiento regulatorio

2.2. El lugar de Dialogflow hoy

En un ecosistema de IA conversacional maduro, Dialogflow y tecnologías similares coexisten con los LLMs, cada uno cubriendo diferentes necesidades:

- **LLMs:** Para conversaciones abiertas y complejas
- **Plataformas tradicionales:** Para interacciones estructuradas y transaccionales

3. Evolución de Herramientas de Desarrollo de IA Bots

- La evolución de las herramientas para crear bots de IA ha sido notable, pasando de sistemas basados en reglas simples a sofisticadas plataformas impulsadas por inteligencia artificial.
- Esta transformación ha traído consigo interfaces más intuitivas y capacidades de procesamiento del lenguaje natural (NLP) cada vez más avanzadas.
- En esta presentación, exploraremos las principales plataformas de desarrollo de bots, sus características clave, ventajas y desventajas, para ayudar a elegir la herramienta adecuada para cada proyecto.

4. Frameworks de IA Bots

Son plataformas diseñadas para facilitar la creación de asistentes virtuales o chatbots que pueden interactuar con usuarios a través de lenguaje natural (escrito o hablado).

Características principales

- **Procesamiento de Lenguaje Natural (NLP):** Utilizan algoritmos avanzados para entender y responder en lenguaje humano.
- **Multicanal:** Permiten la integración con diversos canales de comunicación, como aplicaciones web, móviles y mensajería instantánea (Slack, WhatsApp, Facebook Messenger, etc.).
- **Integración con otros servicios:** Se conectan fácilmente con herramientas empresariales y otros sistemas.
- **Fáciles de usar:** Ofrecen interfaces amigables para desarrollar y entrenar bots, con poca o nula necesidad de codificación avanzada.
- **Escalabilidad:** Están diseñados para manejar grandes volúmenes de interacciones a medida que crecen las necesidades del negocio.

4.1. Cómo crear un bot sencillo con frameworks de IA Bots

4.1.1. Pasos Generales para Desarrollar un Bot

Paso 1: Definir el Propósito del Bot

- ¿Qué problema resolverá?
- ¿Qué tipo de interacciones tendrá con los usuarios? (Respuestas automáticas, servicio al cliente, FAQs, etc.)

Paso 2: Diseñar las Conversaciones

- Establecer los flujos conversacionales: determinar qué preguntas hará el bot y qué respuestas ofrecerá.
- Definir intenciones (intents): las diferentes acciones que el bot debe comprender (por ejemplo, saludar, responder preguntas, procesar solicitudes).
- Identificar entidades: datos importantes dentro de las frases del usuario (nombres, fechas, lugares, etc.).

Paso 3: Entrenar el Bot

- Usar ejemplos de frases que los usuarios podrían decir para entrenar el modelo de Procesamiento de Lenguaje Natural (PLN) del framework.
- Probar y ajustar los diálogos en base a las respuestas del bot.

Paso 4: Conectar con Canales de Comunicación

- Integrar el bot con plataformas de mensajería como WhatsApp, Facebook Messenger, o un sitio web.
- Opcionalmente, habilitar la capacidad de interactuar por voz (usando servicios de voz a texto y texto a voz).

Paso 5: Probar el Bot

- Simular conversaciones y ajustar los flujos y respuestas según sea necesario.
- Asegurarse de que el bot maneje correctamente respuestas inesperadas o preguntas fuera de contexto.

Paso 6: Implementar y Monitorear

- Desplegar el bot en producción en el canal elegido.
- Utilizar herramientas de analítica del framework para monitorear el rendimiento del bot y seguir entrenando el modelo según los comentarios y la interacción de los usuarios.

4.2. Principios Generales

- **Procesamiento de Lenguaje Natural (PLN):** Utilizan modelos de IA para interpretar las entradas de los usuarios y extraer intenciones y entidades.
- **Flujos Conversacionales:** Los bots se diseñan para seguir estructuras lógicas de conversación, con múltiples rutas para distintas respuestas y acciones.
- **Entrenamiento Continuo:** El rendimiento del bot mejora con el tiempo gracias a los ajustes basados en la retroalimentación y el uso.

5. Flujo Conversacional

5.1. ¿Qué es un Flujo Conversacional?

Un flujo conversacional es una secuencia estructurada de interacciones entre un bot y un usuario. Define cómo el bot responderá a las entradas del usuario y guiará la conversación hacia un objetivo específico. Los flujos conversacionales son esenciales para crear experiencias de chatbot efectivas y naturales.

5.2. Componentes de un Flujo Conversacional

- **Intenciones:** Lo que el usuario quiere lograr.
- **Entidades:** Información específica en la entrada del usuario.
- **Diálogo:** La secuencia de mensajes intercambiados.
- **Lógica de decisión:** Cómo el bot elige sus respuestas.
- **Acciones:** Lo que el bot hace en respuesta a las entradas.

5.3. Ejemplos de Flujos Conversacionales

Ejemplo 1: Reserva de Restaurant

- **Bot:** Bienvenido a nuestro servicio de reservas. ¿En qué puedo ayudarte hoy?
- **Usuario:** Quiero hacer una reserva para cenar.
- **Bot:** Perfecto. ¿Para cuántas personas sería la reserva?
- **Usuario:** Para 4 personas.
- **Bot:** Entendido. ¿Para qué día y hora te gustaría hacer la reserva?
- **Usuario:** Este viernes a las 8 PM.
- **Bot:** Excelente. He reservado una mesa para 4 personas este viernes a las 8 PM. ¿Hay algo más en lo que pueda ayudarte?

Ejemplo 2: Soporte Técnico

- **Bot:** Hola, soy el asistente de soporte técnico. ¿En qué puedo ayudarte hoy?
- **Usuario:** Mi impresora no funciona.
- **Bot:** Lamento escuchar eso. Vamos a resolverlo juntos. Primero, ¿podrías decirme el modelo de tu impresora?
- **Usuario:** Es una HP OfficeJet Pro 9015.
- **Bot:** Gracias. ¿Has comprobado si hay papel en la bandeja y si los cartuchos de tinta tienen suficiente tinta?
- **Usuario:** Sí, he comprobado ambas cosas y están bien.
- **Bot:** Entiendo. Vamos a intentar reiniciar la impresora. Por favor, desconéctala de la corriente, espera 30 segundos y vuelve a conectarla. Avísame cuando lo hayas hecho.

5.4. Estructura de Flujos Conversacionales

5.4.1. Estructura básica de un flujo conversacional

Este esquema ilustra el esqueleto fundamental de un flujo conversacional típico:

- Comienza con un saludo del bot
- Seguido por la identificación de la intención del usuario
- Dependiendo de esta intención, el bot puede proporcionar información, procesar una solicitud o finalizar la conversación
- Incluye un bucle para manejar preguntas adicionales, asegurando que todas las necesidades del usuario sean atendidas antes de concluir la interacción

5.4.2. Componentes de un flujo conversacional en acción

Este flujo muestra cómo interactúan los diferentes componentes en un escenario de reserva:

- Destaca la identificación de intenciones (como hacer una reserva o obtener información)
- La extracción de entidades relevantes (fecha y número de personas)
- La lógica de decisión basada en estas entidades
- El diagrama ilustra cómo estos elementos trabajan juntos para guiar la conversación hacia un resultado satisfactorio, ya sea confirmando una reserva o sugiriendo alternativas

5.4.3. Múltiples rutas de diálogos

Este flujo ilustra cómo un bot puede manejar múltiples intenciones del usuario en un solo sistema:

- Abarca diferentes escenarios como hacer una reserva, consultar un menú o preguntar por la ubicación
- Cada ruta tiene sus propias subramificaciones y puntos de decisión
- Demuestra la flexibilidad necesaria en un sistema conversacional complejo para adaptarse a diversas necesidades del usuario

5.4.4. Manejo de errores y entradas inesperadas

Este diagrama se centra en cómo un sistema conversacional maneja situaciones no ideales:

- Muestra el proceso de validación de entradas y las diferentes rutas para manejar errores
- Incluye entradas no entendidas, formatos incorrectos o solicitudes fuera del alcance del bot
- Incorpora ciclos de clarificación y corrección
- Ofrece la opción de ayuda humana cuando sea necesario
- Ilustra la importancia de la robustez en el diseño de flujos conversacionales

5.4.5. Flujo conversacional natural y fluido

Este flujo representa un enfoque más avanzado y natural para las conversaciones bot-humano:

- Comienza con un saludo personalizado y se adapta continuamente al contexto de la conversación
- Incluye elementos como el manejo de digresiones y la capacidad de recordar preferencias del usuario

- Permite una experiencia más fluida y personalizada
- El diagrama muestra cómo un bot puede mantener una conversación más natural, similar a la interacción humana, ofreciendo relevancia y continuidad a lo largo de la interacción

6. Trabajando con Intenciones

6.1. ¿Qué son las Intenciones?

En el contexto de los flujos conversacionales, las intenciones representan el propósito o el objetivo del usuario en una interacción específica. Son la base para comprender qué quiere lograr el usuario y cómo el bot debe responder. Las intenciones ayudan a estructurar la conversación y a proporcionar respuestas relevantes.

Importancia de las intenciones

- **Comprensión del contexto:** Ayudan al bot a entender el contexto de la conversación.
- **Precisión en las respuestas:** Permiten al bot proporcionar respuestas relevantes y precisas.
- **Guía del flujo:** Dirigen el flujo de la conversación hacia los objetivos del usuario.
- **Personalización:** Facilitan la personalización de la experiencia del usuario.
- **Eficiencia:** Mejoran la eficiencia de la interacción al dirigir la conversación de manera efectiva.

6.2. Cómo Trabajar con Intenciones

6.2.1. Identificación de Intenciones

El primer paso es identificar las posibles intenciones del usuario. Esto implica:

- Analizar el texto de entrada del usuario.
- Utilizar técnicas de procesamiento de lenguaje natural (PLN).
- Clasificar la entrada en una categoría predefinida de intenciones.

Ejemplo:

Entrada del usuario: "Quiero reservar una mesa para dos esta noche"

Intención identificada: Hacer una reserva

6.2.2. Mapeo de Intenciones a Acciones

Una vez identificada la intención, el bot debe mapearla a una acción o respuesta específica:

- Definir un conjunto de acciones para cada intención.

- Crear un sistema de reglas o utilizar aprendizaje automático para determinar la acción apropiada.
- Considerar el contexto de la conversación al seleccionar la acción.

Ejemplo:

Intención: Hacer una reserva

Acción: Iniciar el proceso de reserva, solicitando fecha, hora y número de personas

6.2.3. Manejo de Múltiples Intenciones

En conversaciones complejas, es posible que surjan múltiples intenciones. El bot debe ser capaz de manejarlas adecuadamente:

- Priorizar intenciones basándose en la relevancia y urgencia.
- Abordar la intención principal primero.
- Almacenar intenciones secundarias para su posterior seguimiento.
- Permitir al usuario cambiar entre intenciones de manera fluida.

Ejemplo:

Entrada del usuario: "Quiero hacer una reserva, pero primero, ¿tienen opciones vegetarianas?"

Intenciones identificadas: 1) Consultar menú, 2) Hacer reserva

Acción: Responder sobre opciones vegetarianas primero, luego iniciar proceso de reserva

6.3. Mejores Prácticas

- Definir intenciones claras y específicas que cubran los casos de uso más comunes.
- Entrenar el modelo de lenguaje con una amplia variedad de ejemplos y variaciones lingüísticas.
- Implementar un mecanismo de retroalimentación para mejorar la precisión en la identificación de intenciones.
- Manejar intenciones ambiguas o no reconocidas con respuestas de clarificación.
- Actualizar y refinar las intenciones regularmente basándose en las interacciones reales de los usuarios.
- Considerar el contexto de la conversación al interpretar las intenciones.
- Diseñar flujos conversacionales flexibles que puedan adaptarse a cambios en las intenciones del usuario.

6.4. Conclusión sobre Intenciones

El trabajo efectivo con intenciones es fundamental para crear flujos conversacionales naturales y eficientes. Al comprender y manejar adecuadamente las intenciones del usuario, los bots pueden proporcionar una experiencia más satisfactoria y alcanzar los objetivos de la interacción de manera más efectiva. La clave está en la identificación precisa, el mapeo adecuado a acciones y la capacidad de manejar la complejidad de las conversaciones humanas.

7. Trabajando con Entidades

7.1. ¿Qué son las Entidades?

En el contexto de los flujos conversacionales, las entidades son elementos específicos de información extraídos de la entrada del usuario. Representan los datos clave necesarios para cumplir con la intención del usuario. Las entidades proporcionan los detalles cruciales que el bot necesita para realizar acciones o proporcionar respuestas precisas.

Importancia de las entidades

- **Precisión:** Permiten al bot extraer información específica y relevante de las entradas del usuario.
- **Personalización:** Facilitan respuestas más personalizadas basadas en los datos proporcionados por el usuario.
- **Eficiencia:** Ayudan a recopilar información necesaria sin tener que hacer preguntas adicionales.
- **Contextualización:** Proporcionan contexto adicional a las intenciones del usuario.
- **Acción específica:** Permiten al bot realizar acciones más precisas y relevantes.

7.2. Cómo Trabajar con Entidades

7.2.1. Identificación de Entidades

El proceso de identificación de entidades implica:

- Analizar el texto de entrada del usuario para detectar información clave.
- Utilizar técnicas de procesamiento de lenguaje natural (PLN) para extraer datos estructurados.
- Clasificar la información extraída en tipos de entidades predefinidos.

Ejemplo:

Entrada del usuario: "Quiero reservar una mesa para dos personas mañana a las 8 PM"

Entidades identificadas:

- Número de personas: dos
- Fecha: mañana
- Hora: 8 PM

7.2.2. Tipos de Entidades

Existen varios tipos de entidades comunes en los flujos conversacionales:

- **Numéricas:** Cantidades, edades, precios, etc.
- **Temporales:** Fechas, horas, duraciones.
- **Nominales:** Nombres de personas, lugares, productos.
- **Categóricas:** Tipos de productos, categorías de servicios.
- **Personalizadas:** Entidades específicas del dominio de la aplicación.

Ejemplo:

Entrada del usuario: "Busco un vuelo de Madrid a París para el próximo viernes"

Entidades identificadas:

- Origen (Nominal): Madrid
- Destino (Nominal): París
- Fecha (Temporal): próximo viernes

7.3. Mejores Prácticas

- Definir claramente los tipos de entidades relevantes para tu dominio específico.
- Entrenar el modelo de PLN con una amplia variedad de ejemplos para mejorar la precisión en la extracción de entidades.
- Implementar mecanismos de validación para asegurar la corrección de las entidades extraídas.
- Manejar casos de entidades faltantes o ambiguas solicitando clarificación al usuario.
- Actualizar y refinar regularmente el modelo de entidades basándose en las interacciones reales de los usuarios.
- Considerar el contexto de la conversación al interpretar las entidades.
- Diseñar flujos conversacionales que puedan adaptarse a diferentes combinaciones de entidades.

7.4. Uso de Entidades en el Flujo Conversacional

Una vez identificadas las entidades, se pueden utilizar de varias maneras:

- Completar información necesaria para realizar una acción.
- Personalizar respuestas basadas en los datos extraídos.
- Guiar el flujo de la conversación solicitando información faltante.
- Validar y confirmar la información proporcionada por el usuario.

Ejemplo:

Bot: "He encontrado una mesa disponible para dos personas mañana a las 8 PM.
¿Desea confirmar esta reserva?"

Usuario: "Sí, por favor"

Bot: "Perfecto. Su reserva para dos personas mañana a las 8 PM ha sido confirmada.
¿Necesita algo más?"

7.5. Conclusión sobre Entidades

El trabajo efectivo con entidades es crucial para crear flujos conversacionales precisos y eficientes. Al extraer y utilizar correctamente las entidades, los bots pueden proporcionar respuestas más relevantes y personalizadas, mejorando significativamente la experiencia del usuario. La clave está en la identificación precisa, la categorización adecuada y el uso contextual de las entidades para guiar la conversación hacia resultados satisfactorios.

8. Trabajando con Contexto

8.1. ¿Qué es el Contexto de la Conversación?

El contexto de la conversación se refiere a la información acumulada y el estado actual de una interacción entre un usuario y un chatbot. Incluye no solo lo que se ha dicho explícitamente, sino también información implícita, el historial de la conversación, y cualquier dato relevante que pueda influir en la interpretación de los mensajes subsiguientes y en las respuestas del bot.

Importancia del contexto

- **Continuidad:** Permite mantener una conversación coherente a lo largo del tiempo.
- **Comprensión profunda:** Ayuda al bot a entender referencias y alusiones a partes anteriores de la conversación.
- **Personalización:** Facilita respuestas más relevantes y personalizadas basadas en interacciones previas.
- **Eficiencia:** Evita la necesidad de repetir información ya proporcionada.
- **Resolución de ambigüedades:** Ayuda a interpretar correctamente mensajes que podrían ser ambiguos fuera de contexto.

8.2. Cómo Trabajar con el Contexto de la Conversación

8.2.1. Mantenimiento del Contexto

El proceso de mantener el contexto implica:

- Almacenar información clave de cada interacción.
- Actualizar el contexto con nueva información relevante.
- Gestionar la duración y relevancia del contexto almacenado.

Ejemplo:

Usuario: "Me gustaría reservar una habitación"

Bot: Claro, ¿para cuántas personas necesita la habitación?"

Usuario: "Para dos personas"

Bot: Entendido, ¿y para qué fechas le gustaría hacer la reserva?"

Usuario: "Del 15 al 20 de julio"

Bot: "Perfecto. Tengo disponible una habitación doble del 15 al 20 de julio. ¿Desea proceder con la reserva?"

Contexto mantenido: Tipo de habitación (doble), número de personas (2), fechas (15-20 julio)

8.2.2. Tipos de Contexto

Existen varios tipos de contexto que son relevantes en los flujos conversacionales:

- **Contexto de sesión:** Información específica de la conversación actual.
- **Contexto de usuario:** Datos del usuario que persisten entre sesiones.
- **Contexto de dominio:** Conocimiento específico del área o tema de la conversación.
- **Contexto emocional:** Tono y estado emocional percibido del usuario.

Ejemplo:

Contexto de sesión: Búsqueda actual de vuelos

Contexto de usuario: Preferencias de asiento, membresía de viajero frecuente

Contexto de dominio: Conocimiento sobre aerolíneas, rutas y regulaciones

Contexto emocional: Usuario frustrado por cancelaciones previas

8.3. Mejores Prácticas

- Implementar un sistema robusto de gestión de estado para mantener el contexto.
- Balancear la retención de información contextual con la privacidad del usuario.
- Diseñar mecanismos para olvidar contexto irrelevante o desactualizado.
- Utilizar el contexto para mejorar la precisión de la comprensión del lenguaje natural.
- Permitir a los usuarios corregir o actualizar el contexto si es necesario.

- Considerar el contexto al diseñar los flujos de conversación para crear experiencias más naturales.
- Implementar técnicas de desambiguación basadas en el contexto.

8.4. Uso del Contexto en el Flujo Conversacional

El contexto se utiliza de varias maneras en un flujo conversacional:

- Interpretar correctamente las entradas del usuario.
- Proporcionar respuestas más relevantes y personalizadas.
- Anticipar las necesidades del usuario basándose en interacciones previas.
- Manejar referencias anafóricas (por ejemplo, ".eso", "él", ".allí").
- Adaptar el tono y estilo de la comunicación.

Ejemplo:

Usuario: "¿Cuál es el precio de eso?"

Bot: ".El precio de la habitación doble para las fechas del 15 al 20 de julio es de 150€ por noche. ¿Le gustaría proceder con la reserva o necesita más información?"

El bot utiliza el contexto para entender a qué se refiere ".eso" y proporciona una respuesta específica.

8.5. Conclusión sobre Contexto

El manejo efectivo del contexto de la conversación es fundamental para crear chatbots y asistentes virtuales que puedan mantener conversaciones naturales y significativas. Al utilizar adecuadamente el contexto, los bots pueden proporcionar respuestas más precisas, personalizadas y relevantes, mejorando significativamente la experiencia del usuario. La clave está en mantener un equilibrio entre la retención de información útil y la flexibilidad para adaptarse a los cambios en la dirección de la conversación.

9. Diseño de Flujos Conversacionales

9.1. Mejores Prácticas

- Mantén las conversaciones enfocadas y evita diálogos innecesariamente largos.
- Ofrece opciones claras cuando sea posible para guiar al usuario.
- Usa elementos visuales (botones, carruseles) cuando sea apropiado para mejorar la experiencia.
- Personaliza las respuestas basándote en el historial y preferencias del usuario.
- Implementa una personalidad coherente para tu bot que se alinee con tu marca.
- Proporciona una forma fácil para que los usuarios se comuniquen con un humano si es necesario.
- Actualiza y mejora continuamente tus flujos basándote en los datos y feedback de los usuarios.

9.2. Conclusiones sobre Flujos Conversacionales

Diseñar un buen flujo conversacional es un proceso iterativo que requiere planificación cuidadosa, creatividad y un profundo entendimiento de las necesidades del usuario. Al seguir estos pasos y mejores prácticas, puedes crear experiencias conversacionales que sean efectivas, agradables y que verdaderamente aporten valor a tus usuarios. Recuerda que el objetivo final es crear una interacción que se sienta natural y útil, como si el usuario estuviera hablando con un asistente humano bien informado y servicial.

10. Fulfillments

10.1. ¿Qué es un Fulfillment en un Chatbot?

- Fulfillment es la acción que toma el chatbot después de entender lo que el usuario necesita.
- El chatbot no solo responde con un mensaje, sino que realiza una acción para cumplir con el pedido del usuario.

Ejemplo simple

Imagina que estás en una cafetería. Pides un café. El barista (como el chatbot) no solo te dice "¡Claro!", sino que prepara el café y te lo entrega. Preparar el café es el fulfillment.

En un Chatbot: Cuando le pides a un chatbot: "Reserva una mesa en un restaurante", el fulfillment es que el chatbot:

- Busca disponibilidad en el restaurante.
- Reserva la mesa para ti.
- Te confirma la reserva.

Así, el chatbot pasa de simplemente hablar, a hacer algo útil por ti.

Ejemplo de Flujo de Fulfillment

```
// Cuando el usuario pide "Reservar una mesa", el chatbot sigue estos pasos:
Mostrar mensaje: "¡Hola! ¿En qué restaurante te gustaría reservar una mesa?"
// El usuario da el nombre del restaurante.
Leer respuesta del usuario: nombre_restaurante
Mostrar mensaje: "¿Para cuántas personas?"
// El usuario indica el número de personas.
Leer respuesta del usuario: numero_personas
Mostrar mensaje: "¿A qué hora te gustaría la reserva?"
// El usuario indica la hora deseada.
Leer respuesta del usuario: hora_reserva
// El chatbot consulta si hay mesas disponibles en el restaurante a esa hora.
Consultar disponibilidad en el sistema del restaurante:
    Si hay mesa disponible:
        // Si hay una mesa, se procede a la reserva.
        Hacer la reserva para nombre_restaurante, numero_personas y hora_reserva
        Mostrar mensaje: "¡Listo! Tu mesa ha sido reservada para {numero_personas} per
    Si no hay mesa disponible:
        Mostrar mensaje: "Lo siento, no hay mesas disponibles en ese restaurante a esa
        Leer nueva respuesta del usuario y repetir el proceso
Fin del chatbot
```

10.2. Explicación paso a paso

1. **Inicio del chatbot:** El chatbot comienza con un saludo y pregunta en qué restaurante se desea reservar una mesa.
2. **Interacción con el usuario:** El chatbot va pidiendo la información necesaria (nombre del restaurante, número de personas y hora de la reserva).
3. **Consulta la disponibilidad:** El chatbot "pregunta" al sistema del restaurante si hay mesas disponibles para esos datos.
4. **Confirmación o nueva opción:** Si hay disponibilidad, confirma la reserva al usuario. Si no la hay, ofrece nuevas opciones o cambios.

11. Dialogflow de Google

11.1. Implementación y Escalabilidad de Agentes Conversacionales

El uso de Dialogflow facilita la creación de agentes inteligentes con capacidades de procesamiento de lenguaje natural. Gracias a su arquitectura basada en intents y contexts, y su integración con Google Cloud, es posible generar soluciones a gran escala, ofreciendo una experiencia personalizada y eficiente para los usuarios. Aprovechar esta plataforma permite una mejora continua y una mayor interacción con los clientes a través de múltiples canales de comunicación.

11.2. Historia

Originalmente Api.ai, adquirido por Google en 2016. Lanzamiento de Dialogflow ES (Essential) y CX (Customer Experience).

11.3. Características

Interfaz visual intuitiva, agentes preentrenados, integración con Google Cloud y servicios de IA, soporte nativo para múltiples idiomas y plataformas.

11.4. Arquitectura

Basada en intents, entities, y contexts. Fulfillment para lógica personalizada.

11.5. Ventajas y Desventajas

- **Ventajas:** Facilidad de uso, potentes capacidades de NLP.
- **Desventajas:** Menor flexibilidad para lógica compleja, puede ser costoso a gran escala.

11.6. Cómo trabajar con Dialogflow

1. **Definición de Intenciones (Intents):** Crear las intenciones del usuario (preguntas o comandos) que el chatbot debe comprender y responder.
2. **Entrenamiento con Ejemplos:** Proporcionar frases de ejemplo para entrenar al chatbot a reconocer las variaciones de las intenciones.
3. **Creación de Entidades:** Definir entidades (variables) que extraen información clave, como fechas, ubicaciones o nombres.
4. **Configuración de Respuestas:** Especificar las respuestas del chatbot, ya sea con texto, integraciones externas o acciones personalizadas (fulfillment).
5. **Pruebas y Optimización:** Probar el chatbot en el simulador de Dialogflow y ajustar las respuestas o entrenamientos según sea necesario para mejorar su precisión.

12. Microsoft Bot Framework

12.1. Historia y Evolución

Lanzado en 2016 como parte de la estrategia de IA de Microsoft. Actualizaciones significativas con la versión 4.0 en 2018.

12.2. Características Principales

SDK para .NET y JavaScript, Bot Framework Composer para diseño visual, integración con servicios cognitivos de Azure (LUIS, QnA Maker), soporte para múltiples canales.

12.3. Arquitectura Básica

Patrón de "turnos" en la conversación. Componentes clave: Adaptadores de canal, Middleware, Administrador de diálogo.

12.4. Cómo trabajar con Microsoft Bot Framework

1. **Definición del Diálogo:** Crear diálogos estructurados en el bot que guiarán la interacción del usuario, usando lenguajes como C# o Node.js.
2. **Uso de LUIS (Language Understanding):** Integrar LUIS para que el bot reconozca las intenciones del usuario y extraiga entidades relevantes.
3. **Desarrollo con SDK:** Utilizar el SDK de Bot Framework para construir y conectar las interacciones del bot con diferentes plataformas de mensajería.
4. **Configuración de Canales:** Publicar el bot en múltiples canales, como Microsoft Teams, Slack o sitios web, a través del Bot Framework Service.
5. **Pruebas y Monitoreo:** Probar el bot en el simulador de Bot Framework y monitorear su rendimiento con herramientas como Application Insights para ajustes y mejoras.

13. Conclusiones Generales

- Las plataformas tradicionales de chatbots siguen siendo relevantes en la era de los LLMs, especialmente para casos de uso empresariales donde se requiere control, precisión y eficiencia.
- El diseño efectivo de flujos conversacionales requiere comprender intenciones, entidades y contexto.
- Frameworks como Dialogflow y Microsoft Bot Framework proporcionan herramientas robustas para crear experiencias conversacionales estructuradas.
- La implementación de fulfillments permite que los chatbots pasen de simplemente conversar a realizar acciones concretas que resuelven necesidades del usuario.
- El futuro probablemente verá una convergencia entre las capacidades de los LLMs y las ventajas estructurales de las plataformas tradicionales.

14. Referencias y Recursos Adicionales

- Documentación oficial de Google Dialogflow: <https://cloud.google.com/dialogflow/docs>
- Documentación de Microsoft Bot Framework: <https://docs.microsoft.com/en-us/azure/bot-service/>
- "Building Conversational Experiences with Dialogflow Google Cloud (2023)
- "Designing Voice User Interfaces Cathy Pearl (O'Reilly)
- Conversational UX Design Robert J. Moore y Raphael Arar (ACM Books)