

Flujos conversacionales en WhatsApp para restaurantes y delivery

Capacidades y limitaciones de WhatsApp Flows oficiales de Meta

WhatsApp Flows es una nueva funcionalidad de la plataforma WhatsApp Business que permite a las empresas crear experiencias interactivas **dentro del chat** de WhatsApp, a modo de mini-aplicaciones o formularios. Básicamente, un *Flow* es una secuencia de pantallas con elementos de interfaz visual que el usuario puede completar sin salir de la conversación ¹. Cada pantalla de un Flow contiene componentes como campos de texto (por ejemplo, para nombre, correo, números), títulos, subtítulos, **imágenes**, menús desplegables, casillas de verificación (checkboxes), botones de opción (radio buttons), selectores de fecha, entre otros ². Estos componentes se configuran para guiar al usuario paso a paso en la tarea deseada (p. ej., llenar un formulario de reserva, hacer un pedido, responder una encuesta) con solo tocar opciones o escribir datos breves en lugar de escribir mensajes largos. Los Flows se inician típicamente a través de un mensaje con un botón CTA (Call-To-Action) en la conversación – por ejemplo, un botón “*Hacer pedido*” o “*Reservar mesa*” que al tocarlo abre la interfaz interactiva dentro de WhatsApp ³ ⁴.

Ejemplo de experiencia con WhatsApp Flows: el usuario inicia un Flow (p. ej., registrarse a una promoción) desde un botón en el chat, completa formularios multi-pantalla con campos e indicaciones interactivas, y al finalizar se muestra un mensaje de confirmación en la conversación. Todo ocurre dentro de WhatsApp, sin redirigir a páginas externas.

Componentes visuales e interacciones: WhatsApp Flows soporta una variedad de elementos UI nativos. Por ejemplo, se pueden presentar opciones de respuesta mediante botones o listas desplegables en vez de pedir al usuario que escriba; se pueden solicitar fechas usando un calendario (Date Picker) nativo; e incluso es posible permitir la carga de imágenes o documentos dentro del flujo (útil para adjuntar una foto de un comprobante, identificación, etc.) ⁵ ⁶. Cada pantalla del Flow puede incluir hasta 8 componentes distintos, y un Flow completo puede tener como máximo 8 pantallas encadenadas ⁷. Esto impone una **limitación**: los Flows oficiales están diseñados para interacciones relativamente breves y estructuradas. No es viable crear un formulario interminable o un árbol de conversación muy profundo dentro de un solo Flow debido a estas restricciones de longitud. En la práctica, 8 pasos suelen ser suficientes para muchos casos de uso (por ejemplo, tomar un pedido paso a paso: selección de producto, opciones extra, dirección, método de pago, confirmación), pero procesos más complejos podrían requerir dividirse en varios Flows o combinar Flows con mensajes conversacionales adicionales.

Otro punto a considerar es que WhatsApp Flows es una función disponible **exclusivamente a través de la WhatsApp Business API** (ya sea API Cloud de Meta o proveedores oficiales). Para usar Flows, la empresa debe tener una cuenta de WhatsApp Business aprobada y cumplir ciertos requisitos de calidad (ej. nombre verificado, buen rating, etc.), además de crear plantillas de mensaje específicas para lanzar los Flows ⁸ ⁹. Esto significa que pequeños negocios que solo usan la app normal de WhatsApp Business (la aplicación móvil) *no* tienen acceso directo a crear Flows desde ahí; necesitarían pasar al entorno API mediante un proveedor. Asimismo, fuera de la ventana de 24 horas de sesión iniciada por el cliente, los

Flows solo pueden enviarse como mensajes plantilla previamente aprobados por Meta ¹⁰ (al igual que otros mensajes proactivos). Estas limitaciones implican cierta barrera de entrada y planificación: hay que diseñar el flujo en el **Flow Builder** (herramienta visual disponible en WhatsApp Manager de Facebook o vía JSON) ¹¹ ¹², someterlo a revisión si forma parte de una plantilla, y publicarlo antes de usarlo en producción ¹³ ¹⁴.

Capacidades destacadas: A pesar de sus límites, los Flows oficiales ofrecen funcionalidades potentes nativamente integradas en WhatsApp. Por un lado, simplifican *interacciones complejas* convirtiéndolas en una serie de pasos manejables dentro del chat ¹⁵. Por ejemplo, en lugar de una conversación desordenada con múltiples idas y venidas para hacer una reserva, un Flow puede guiar al cliente por pantallas bien definidas: selección de fecha, horario, número de personas y confirmación – todo con toques en botones en vez de escribir manualmente ¹⁶ ¹⁷. Esto reduce enormemente la fricción y los errores en la comunicación, ya que el usuario sigue un camino claro. Además, los Flows soportan integrar **respuestas automatizadas en tiempo real y conexiones con sistemas externos**. Es posible configurar un Flow “con endpoint” (conexión dinámica) que llame a un servidor o API para validar información o rellenar opciones en el momento ¹⁸ ¹⁹. Por ejemplo, un Flow de restaurante podría consultar una base de datos de menú para listar categorías o platos disponibles actualizados al momento, o un Flow de delivery podría conectarse a un sistema de tracking para mostrar opciones de horario de entrega según la zona del cliente. La integración con sistemas de pago también es factible: aunque WhatsApp tiene su propia función de pagos en ciertos países, un Flow puede incluir un paso final con un botón de pago (que redirige a una pasarela web externa) o puede finalizar informando de un monto y luego enviando un **link de pago** seguro. Desde la perspectiva de UX, todo esto ocurre sin sacar al cliente de WhatsApp – lo cual es clave, dado que mantener al usuario *dentro del chat* aumenta la probabilidad de completar la acción (menores tasas de abandono que si se les envía a un sitio web externo) ²⁰ ²¹.

Limitaciones y consideraciones: Los WhatsApp Flows, al ser estructurados, **no cubren bien interacciones libres o no previstas**. Si el usuario intenta desviarse (por ejemplo, escribiendo algo no relacionado cuando está en medio de un Flow), se encontrará con respuestas de error predeterminadas o podría salirse del flujo. Es importante diseñar buenos mensajes de error y educar al usuario a usar los botones en pantalla ²² ²³. Otro aspecto es que, por ahora, el contenido de un Flow debe predefinirse en gran medida (a menos que use endpoint dinámico). Esto implica que menús muy extensos (por ejemplo, 100 platos) no son prácticos de poner enteramente en un Flow estático debido al límite de componentes; en tal caso, quizá convenga usar el Flow solo para pasos generales y manejar la parte amplia con otro enfoque (como enviar el catálogo por secciones, o combinar con mensajes fuera del Flow). También hay que recordar que la **disponibilidad de Flows es reciente** (anunciada a fines de 2023 e implementada en 2024), por lo que los usuarios de WhatsApp recién se están familiarizando con esta modalidad. Sin embargo, las primeras métricas son muy prometedoras: en sectores como reservas y pedidos por chat, se han visto incrementos notables al implementar Flows. Por ejemplo, un estudio reportó un *50% de aumento en la cantidad de reservas* realizadas vía WhatsApp tras habilitar Flows interactivos, comparado con el modelo tradicional de atender por mensaje manual; igualmente, un *72% más de ingresos* generados por pedidos cuando se usan Flows versus el flujo conversacional previo ²⁴. Esto sugiere que la interfaz guiada y visual mejora la conversión y la satisfacción del cliente, al hacer el proceso más rápido y sencillo. En resumen, las **capacidades** de WhatsApp Flows – interfaces ricas con botones, formularios y lógica en el chat – habilitan casos de uso avanzados (formularios de registro, compras, reservas, soporte estructurado, etc.) con gran comodidad, mientras que las **limitaciones** obligan a mantener esos flujos acotados en pasos y disponibles solo bajo las políticas de la API Business. Para el sector de *food & beverage* específicamente, WhatsApp Flows abre la puerta a ofrecer **pedidos y reservas in-app** (dentro del mismo WhatsApp) de manera ágil. Un

cliente puede desde el menú de la empresa en WhatsApp tocar “Ordenar ahora” y completar su orden con un flujo interactivo en menos de un minuto. De hecho, en proyectos piloto se ha logrado que el *tiempo de hacer un pedido* completo vía WhatsApp (usando automatizaciones y Flows) sea de alrededor de **30 segundos**, comparado con los 5-10 minutos que podría tomar haciendo el pedido por una app tradicional o llamada telefónica ²⁵. Esta eficiencia es un cambio radical para el ecosistema de delivery.

Herramientas externas para crear flujos personalizados en WhatsApp

Si bien los Flows oficiales de Meta ofrecen una solución nativa, muchas empresas optan por **herramientas de terceros** para diseñar experiencias conversacionales personalizadas en WhatsApp, a veces incluso combinando enfoques. Plataformas de chatbot y APIs de comunicación proporcionan constructores visuales de flujos (“flow builders”) o frameworks flexibles para crear conversaciones automatizadas. A continuación, revisamos algunas de las herramientas más destacadas – **Twilio Studio**, **Landbot**, **WATI**, **Zoko**, entre otras – y cómo se aplican en el contexto de restaurantes:

- **Twilio Studio (WhatsApp via Twilio API):** Twilio es un proveedor de API de WhatsApp Business que permite a desarrolladores y empresas integrar WhatsApp en sus aplicaciones. Twilio Studio es su entorno de *flow builder* visual, donde se pueden arrastrar y conectar bloques lógicos (enviar mensaje, esperar respuesta, condiciones, integraciones HTTP, etc.) para crear un **flujo conversacional** sin código pesado. En el caso de un restaurante, Twilio Studio permite construir un chatbot que, por ejemplo, salude al cliente, le muestre el menú (enviando mensajes de texto o imágenes del menú), pregunte qué desea ordenar, tome la orden y luego confirme la dirección de entrega. Cada uno de estos pasos es un bloque en el flujo de Twilio Studio, donde se pueden definir mensajes con opciones (usando las plantillas interactivas de WhatsApp como *List Messages* o *Quick Reply buttons* para simular un menú de opciones). La fortaleza de Twilio es su **flexibilidad y programación**, ya que se puede incorporar lógica personalizada (por ejemplo, llamar a una función que calcule el total del pedido o consulte stock antes de confirmar). Además, Twilio facilita la integración con SMS u otros canales en caso de ser necesario un canal alternativo. Sin embargo, a diferencia de los Flows oficiales, Twilio por sí solo no crea una interfaz “multiventana” dentro de WhatsApp – más bien, se apoya en los formatos interactivos estándar (listas, botones, mensajes de texto) para guiar la conversación. Esto significa que el resultado es más parecido a un chatbot conversacional tradicional, pero muy bien estructurado. Muchos restaurantes han usado Twilio API para sus bots; por ejemplo, existe [un tutorial de Twilio](#) donde se construye un bot para descubrir restaurantes veganos cercanos, demostrando cómo Twilio puede manejar ubicaciones, consultas y respuestas en WhatsApp. En resumen, Twilio brinda el *lienzo* para flujos altamente personalizados, ideal si se requiere integraciones complejas (POS, CRM, pagos) y se cuenta con recursos de desarrollo.
- **Landbot:** Landbot es una plataforma no-code enfocada en chatbots conversacionales con una experiencia muy rica, que puede desplegarse en la web, Messenger y también en WhatsApp. Landbot destaca por su **interfaz de diseño tipo drag-and-drop** extremadamente amigable, donde cada bloque representa una pregunta del bot o una acción. Para WhatsApp, Landbot permite crear un flujo conversacional que el usuario percibe como si hablara con un agente, pero con la ventaja de que Landbot gestiona la lógica. En restaurantes, Landbot ha sido utilizado para hacer chatbots de menú y pedidos. Por ejemplo, Landbot publicó un tutorial de “Restaurant Chatbot” donde enseña a

montar un bot de pedidos que puede mostrar el menú por categorías, con botones enriquecidos con imágenes de los platos, recolectar la orden y procesar pagos en línea ²⁶ ²⁷ . En ese enfoque, Landbot recomienda incluso usar un **chatbot web** (por ejemplo, un enlace o código QR que abre un chat estilo web) para mayor flexibilidad en UI; sin embargo, también soporta WhatsApp conectándose a la API oficial. En WhatsApp, Landbot tendrá que enviar mensajes con texto y opciones permitidas (por las políticas de WhatsApp), por lo que a veces la experiencia se adapta ligeramente (por ejemplo, en vez de carruseles gráficos, en WhatsApp se mandan listas de texto con opciones). Aun así, Landbot facilita la creación de flujos conversacionales complejos con condiciones, validaciones (se puede verificar que un número de mesa sea válido, o que el texto ingresado luzca como un email, etc.) sin escribir código. Herramientas como Landbot son útiles para restaurantes pequeños y medianos que quieren un bot en WhatsApp rápido, con la capacidad de manejar **menús, reservas, FAQs y promociones**. Un posible flujo Landbot para restaurante sería: el cliente escribe “Hola”, el bot muestra un saludo con opciones “1. Ver menú, 2. Reservar mesa, 3. Ver estado de mi pedido, 4. Hablar con humano”. Según la elección, el flujo salta a la sub-conversación correspondiente, todo esto configurable visualmente. La limitante de Landbot podría ser que, sin un desarrollador, las integraciones avanzadas (p. ej. consultar en tiempo real el inventario de un plato) requieren usar sus bloques de código o webhooks, lo cual es más técnico. No obstante, para muchas funciones típicas (mostrar un menú fijo, tomar datos del cliente, registrar un pedido en una hoja de cálculo o base de datos) Landbot ya trae integraciones sencillas (ej. con Google Sheets, correo, etc.).

- **WATI:** WATI (WhatsApp Team Inbox) es un proveedor oficial de WhatsApp Business API orientado a PYMEs, que ofrece tanto una plataforma de atención multiagente como herramientas de automatización (chatbots y Flows). En particular, WATI implementó recientemente la función de **WhatsApp Flows** dentro de su plataforma, permitiendo a sus usuarios diseñar Flows visuales con un builder integrado ²⁸ ²⁹ . Por ejemplo, un restaurante usando WATI podría crear un Flow de **“Feedback de Servicio”**: después de que un pedido es entregado, el bot envía un mensaje al cliente con un botón “Calificar experiencia”. Al tocarlo, se abre un Flow en WhatsApp donde el cliente llena una breve encuesta (calificación de 1 a 5, comentario opcional, subir foto del platillo si desea). Este Flow se habría construido en WATI de forma gráfica y publicado para su número de WhatsApp. Asimismo, WATI permite insertar el uso de Flows dentro de sus chatbots: en su constructor de bots, uno puede añadir un nodo especial que lanza un Flow de WhatsApp en cierto punto de la conversación ³⁰ ³¹ . Para restaurantes, esto es útil si combinas respuestas libres con formularios. Por ejemplo, supongamos que el bot primero pregunta por texto “¿Cuál es tu número de orden?” y el cliente lo envía; luego el bot verifica estado y detecta que *falta* información de dirección, entonces puede iniciar un Flow para recoger dirección y referencias en un formato estructurado (en lugar de seguir pidiendo datos uno por uno por texto). WATI también ofrece plantillas prediseñadas de Flows (como *“Generación de leads”* o *“Formulario de feedback”*) que se pueden editar ³² , lo cual acelera la implementación. Una ventaja de usar WATI es que está todo integrado: la misma plataforma maneja el envío de campañas, el inbox con agentes humanos y los Flows/chatbots, sin que el negocio deba programar con la API directamente. Esto baja la barrera para, digamos, un restaurante que no tiene equipo técnico pero quiere automatizar WhatsApp. En cuanto a **aplicabilidad en restaurantes**, WATI publica contenido enfocado a ese rubro. En su guía de “WhatsApp Business para Restaurantes” aconsejan usar respuestas rápidas para FAQs (horarios, ubicación), catálogos para mostrar el menú dentro del perfil de empresa de WhatsApp, y chatbots o Flows para reservas y pedidos en automático. Las funcionalidades de WATI permiten que un restaurante configure un mensaje de bienvenida que ofrezca opciones (usando emojis de comida para hacerlo atractivo, por ejemplo), implemente un flujo de reserva (solicitando fecha y hora mediante un Flow con Date Picker) y

notifique por WhatsApp cuando el pedido esté en camino. Dado que WATI está orientado a negocio, también proporciona métricas – por ejemplo cuántos clientes completaron el Flow, a cuántos hubo que transferir con un humano, etc., lo cual puede ayudar a mejorar la experiencia conversacional.

- **Zoko:** Zoko es otra solución enfocada en comercio conversacional por WhatsApp, conocida por su integración con **Shopify** y herramientas de ventas. Zoko incluye un *flow builder* robusto llamado “Zoko Flows”, que la compañía promociona como “el constructor de flujos más avanzado para WhatsApp” ³³. Con Zoko Flows, se puede automatizar prácticamente cualquier tarea repetitiva en WhatsApp mediante una interfaz de *canvas* visual: desde **confirmaciones de pedidos contra reembolso (COD)**, hasta flujos multistep que se conectan con Shopify u otras aplicaciones (por ejemplo, una secuencia para solicitar una reseña *post-compra*, conectada con la app Judge.me de reviews) ³⁴. Para un restaurante que vende a través de un sitio de ecommerce (o que tiene una tienda Shopify para pedidos online), Zoko resultaría útil porque sincroniza el catálogo de productos con WhatsApp – incluso permite enviar el catálogo dentro del chat – y cualquier flujo que se cree puede consultar ese catálogo. Piensa en **ventas adicionales (upselling)**: si el cliente pide una pizza, un *Flow de Zoko* podría detectarlo y automáticamente guiar al cliente en WhatsApp con algo como “¿Quieres añadir una bebida o postre?” mostrando opciones extraídas de Shopify. Otra capacidad de Zoko es la de manejar **broadcasts** (difusión) y luego Flows: por ejemplo, enviar una campaña masiva a clientes con un botón “Ver Promoción del Día” que dispara un Flow dentro del chat donde el cliente puede navegar la promoción e incluso generar su orden. Zoko, al igual que WATI, integra el soporte multiagente – importante en restaurantes para escalar a un humano si el bot no entiende una pregunta, como por ejemplo si un cliente escribe “¿El plato del día tiene gluten?” y el bot/Flow no está preparado, un agente puede intervenir en la misma interfaz. En suma, Zoko es más utilizado por **retailers en línea**, pero sus flujos personalizables tienen aplicación directa en food & beverage para automatizar confirmaciones de pedido, notificaciones de estado (un flujo para seguimiento de entrega, etc.) y en general para conectar WhatsApp con sistemas externos que el restaurante use.

Además de estas, existen otras herramientas notables: **Kommo (antes AmoCRM)** tiene también un constructor de chatbot para WhatsApp usado por restaurantes; **Gupshup** y **Infobip** ofrecen APIs y templates para flujos conversacionales (Infobip incluso permite crear WhatsApp Flows oficiales desde su plataforma ³⁵); **Engati**, **Yellow.ai**, **Dialogflow** con WhatsApp integration, etc., todas proporcionan maneras de implementar menús y conversaciones automatizadas. La elección de la herramienta depende de factores como el nivel de personalización requerido, la capacidad técnica del equipo, presupuesto y si se prefiere un enfoque más *visual* vs. más programable. Por ejemplo, un restaurante independiente pequeño tal vez opte por un Landbot o WATI por simplicidad; mientras que una cadena grande con IT propio podría ir con Twilio API o directamente con Flows de Meta vía Infobip, para tener mayor control y escalabilidad.

En cualquier caso, el objetivo de estas herramientas es permitir **flujos conversacionales personalizados** en WhatsApp que complementen o suplan a los Flows oficiales cuando se necesite mayor flexibilidad. Muchas de ellas pueden incluso trabajar *en conjunto*: no es excluyente usar, por decir, WhatsApp Flows oficiales para ciertas partes y un chatbot tradicional para otras. Un ejemplo híbrido podría ser: Twilio envía un mensaje inicial con un Flow oficial adjunto (por ser más amigable para llenar datos del pedido), y luego cuando el Flow termina, Twilio continúa la conversación con su lógica (por ejemplo, asignando un repartidor y enviando actualizaciones via mensajes de texto). Esta combinación maximiza lo mejor de ambos mundos.

Mejores prácticas de UI/UX conversacional para restaurantes en WhatsApp

Diseñar una experiencia conversacional efectiva en WhatsApp para food & beverage requiere seguir **buenas prácticas de UX** que contemplen las expectativas del usuario, las limitaciones del canal y la naturaleza de las tareas (pedir comida, reservar, etc.). A continuación, se presentan algunas recomendaciones clave de diseño de conversación y flujo para restaurantes y servicios de delivery:

- **Mantener la interacción guiada y sencilla:** La primera regla es evitar que el usuario tenga que *adivinar* qué hacer. En lugar de preguntas abiertas del tipo “¿En qué puedo ayudarte?” (que pueden disparar cualquier respuesta), es preferible ofrecer opciones claras. Por ejemplo, al iniciar el chat con un restaurante, se puede dar un menú de opciones: “1  Ver Menú, 2  Hacer un Pedido, 3  Reservar Mesa, 4  Hablar con Atención al Cliente”. Esto se puede implementar como un mensaje con botones numerados o una lista interactiva. Guiar al usuario reduce la fricción y evita confusiones, especialmente en WhatsApp donde la interacción ideal es rápida. Un flujo bien estructurado llevará al cliente paso a paso: primero elegir la categoría (ej. bebidas, entradas, platos fuertes, postres), luego el ítem específico, luego personalizaciones, etc., presentando *poca información a la vez* para no abrumar. En un chatbot conversacional esto implica enviar varios mensajes cortos en secuencia; en un Flow visual implica dividir en pantallas claras. En ambos casos, el cliente debe sentir que es un proceso **natural y conversacional** más que un formulario rígido, incluso si por detrás lo es.
- **Optimizar para móvil y WhatsApp:** Parece obvio, pero hay que diseñar pensando en pantallas pequeñas y en el contexto de WhatsApp. Evitar párrafos muy largos de texto – es mejor usar frases breves y, cuando sea posible, reemplazar texto por elementos interactivos. Por ejemplo, para preguntar la cantidad de comensales en una reserva, en vez de “Por favor ingresa el número de personas”, se puede enviar directamente una lista de opciones (“1 persona”, “2 personas”, “3”, “4+”) para que el usuario toque. Asimismo, usar el formato de **listas desplegadas** de WhatsApp (List Messages) es útil para menús de opciones largas (p. ej., 20 platos) ya que presentan las opciones en una vista de lista compacta dentro de la app. Siempre se debe probar la experiencia en un teléfono real: comprobar que los botones se leen bien, que las imágenes (si se envían fotos de platos) se ven adecuadas y no pesan demasiado (WhatsApp recomienda imágenes <300 KB para rapidez ⁵). La rapidez es esencial: los usuarios de WhatsApp esperan inmediatez, así que el bot debe responder en cuestión de 1-2 segundos a cada input para mantener la sensación de “chat”. Cualquier integración (por ejemplo, calculando total con un backend) debe estar optimizada para no introducir demoras perceptibles. Una buena práctica es enviar *señales de escritura* (“typing indicators”) si una operación tarda un poco, para que el usuario sepa que el bot está procesando.
- **Usar lenguaje claro, tono cercano y branding consistente:** La personalidad del bot o flujo debe reflejar la marca del restaurante, pero siempre con lenguaje sencillo. Se recomienda saludar usando el nombre del cliente si se tiene (“¡Hola Juan! Soy el asistente de *Restaurante X*”) y emplear un tono amable y servicial, como lo haría un buen mesero virtual. Incluir emojis apropiados puede dar calidez (por ejemplo,  para representar pizza en las opciones,  para confirmar). Sin embargo, no se debe abusar de los emojis o expresiones informales si el público del restaurante espera algo más serio (imaginemos un restaurante gourmet de lujo – tal vez usaría un tono más formal, pero aun así cortés). También es importante **gestionar las expectativas** con el lenguaje: si el pedido va a tardar 40 minutos, el bot debe comunicarlo claramente. Al final de un flujo de pedido, un buen mensaje

podría ser " ¡Listo! Tu pedido fue recibido. En aproximadamente 30 minutos nuestro repartidor te entregará tu orden. 📦 ¡Gracias por tu compra!". Esto confirma al usuario que todo salió bien e informa del siguiente paso. En cuanto al branding, si el restaurante tiene un estilo particular (colores, logos, slogans), en WhatsApp eso se puede reflejar en pequeños detalles: el **uso del nombre del restaurante**, quizá un slogan corto en el mensaje de bienvenida, y se puede incluir el logo en la foto de perfil de WhatsApp Business y en imágenes dentro de Flows.

- **Incluir contenido enriquecido (rich media) de forma estratégica:** A diferencia de un chatbot puramente de texto, WhatsApp permite enviar **imágenes, videos, ubicaciones, PDFs**, etc. En experiencias conversacionales de restaurantes, las imágenes de los platillos pueden elevar mucho la interacción – ver la foto de una hamburguesa al elegir puede tentar al cliente a agregarla. Si se va a mostrar el menú, es preferible no mandar *todo* el menú en PDF (eso obliga al usuario a abrir un archivo, hacer zoom... poco amigable). En su lugar, se pueden mandar **porciones del menú** según lo que pide el cliente. Por ejemplo, si el cliente elige "Ver Menú: Postres", el bot podría enviar 3-5 fotos de postres (usando mensajes multimedia separados o un mensaje con varias imágenes en fila, teniendo en cuenta que WhatsApp permite hasta 30 imágenes pero no conviene saturar). Cada imagen puede ir acompañada de un texto descriptivo breve y un número o letra para que el cliente seleccione. Alternativamente, usar un *mensaje de lista* con el nombre de cada postre y un botón "Ver foto" que devuelve la imagen bajo demanda, para controlar la cantidad de multimedia enviada. Las imágenes deben tener buena calidad pero tamaño reducido, y conviene que tengan formato cuadrado o vertical para que se aprecien bien en la pantalla del teléfono. Otro uso de rich media: enviar la **ubicación** del restaurante (pin de Google Maps) para ayudar al cliente a llegar, o solicitar al cliente que comparta su ubicación para calcular la entrega. Esto último es muy útil: en lugar de pedir que escriba su dirección, se le puede indicar "Por favor, comparte tu ubicación actual para calcular la ruta de delivery" – WhatsApp tiene un botón nativo para adjuntar ubicación que el usuario puede usar. En Flows, actualmente no hay un componente de mapa, así que esto se haría fuera del Flow con instrucciones. En general, el contenido enriquecido debe usarse cuando aporta valor visual o de usabilidad, pero evitar enviar medios innecesarios que consuman datos o distraigan del objetivo.

- **Confirmaciones y posibilidad de corrección:** Dado que pedir comida o reservar son acciones críticas, siempre incluir un paso de **confirmación final** donde el usuario pueda revisar y aprobar los datos. Por ejemplo, antes de enviar la orden al restaurante, listar el resumen: "Vas a pedir: 1x Pizza Margarita – Sin cebolla; 2x Gaseosa; Total: \$25.000. *Dirección:* Calle 123, Bogotá. ¿Confirmar pedido? [Sí, confirmar] [Cancelar]". Esto se puede implementar con un mensaje de texto y que el usuario escriba "Sí" o "No", o idealmente con botones. En un Flow oficial es buena práctica usar la última pantalla como resumen y tener un botón de **Submit**. Si el usuario se da cuenta de un error (ej. se equivocó de dirección), debe tener un camino para corregirlo. Conversacionalmente, se puede permitir responder "No" y reiniciar el flujo de pedido desde el punto necesario (quizá repitiendo la pregunta de dirección). En un Flow, lamentablemente no hay botón de "Atrás" nativo durante la interacción, así que se debe educar al usuario a cancelar y comenzar de nuevo si hace falta (o diseñar flujos cortos para minimizar errores). También, tras la confirmación, proporcionar una respuesta afirmativa: "👏 Pedido confirmado. Número de orden: #12345. Te notificaremos cuando esté en camino.". Esa notificación de *pedido en camino* también es parte de la UX: conviene avisar al usuario en el momento oportuno (eso puede ser con un mensaje plantilla más adelante cuando el repartidor salga). En la parte de reservas, similar: después de que el cliente reserva, enviar un mensaje de confirmación con los detalles y quizás un **código de reserva** o enlace para cancelar si no puede asistir.

- **Manejo de errores y ayudas:** No todos los usuarios siguen el guion perfectamente. Algunos pueden escribir cosas como “hola quiero una mesa para mañana 8pm” en un solo mensaje inicial. El sistema debe ser capaz de manejar estos casos con gracia. Un chatbot con IA (NLP) podría interpretar esa frase y extraer la intención de reserva, pero si no se cuenta con NLP sofisticado, al menos se debe reconocer ciertas palabras clave para no descarrilar la conversación. Una buena práctica es tener **palabras clave** que el bot entienda en cualquier momento, como “MENÚ”, “RESERVAR”, “PEDIR”, “AYUDA” – de modo que si el usuario las envía, el bot pueda saltar al flujo correspondiente. También, definir un mensaje de fallback (cuando el bot no entiende) que no sea confuso: por ejemplo “Lo siento, no logré entender tu mensaje. Por favor escribe *MENÚ* para ver opciones o *AYUDA* si necesitas asistencia de un agente.”. En WhatsApp, el lenguaje debe ser cercano, pero es importante ser conciso en la corrección de errores. Si se usa un Flow oficial y el usuario en vez de tocar un botón escribe texto no esperado, WhatsApp enviará un mensaje de error automático (configurable al crear el Flow) para decir algo como “Por favor utiliza las opciones provistas” ²². Se debe redactar ese mensaje de manera amable. Igualmente, siempre ofrecer una **salida a humano**: es crucial en UX conversacional dar la opción de hablar con una persona real si el chatbot/Flow no resuelve lo que el cliente necesita. Para un restaurante, esto puede significar que en cualquier menú haya una opción “Hablar con representante” que notifique a un encargado (quizá vía la app WhatsApp Business multiagente o un sistema de atención) para tomar el relevo. Los Flows oficiales no tienen una opción interna de “derivar a humano”, pero se puede diseñar que si el usuario cancela o abandona, un agente le escriba manualmente aparte. En chatbots no oficiales, simplemente incluir esa rama en el flujo.

- **Personalización y contexto:** Un buen diseño conversacional aprovecha la información disponible para personalizar la experiencia. Si el sistema ya sabe el nombre del usuario (porque viene del CRM o porque lo ingresó antes en un Flow), usarlo en los mensajes siguientes (“Hola *María*, bienvenido de nuevo”). Si el usuario ya es cliente frecuente, se le puede ofrecer su platillo favorito: “¿Deseas pedir lo mismo que la vez pasada, *Sushi Combo 2*? Responde *Sí* o *No*”. Estas sorpresas agradables generan fidelidad. Por supuesto, esto implica tener integraciones con historiales de pedido. Otra personalización sencilla es según la hora: si el usuario escribe “hola” a las 9am, quizá el bot debe saludar con “¡Buenos días! ¿Buscas nuestro menú de desayunos?”; si es 10pm tal vez indicar “Estamos cerca del horario de cierre, pero aún puedes realizar un pedido.”. Estas lógicas de contexto muestran “empatía” de la interfaz con la situación del cliente.

- **Elementos conversacionales específicos para food & beverage:** En este sector, hay ciertos flujos típicos que conviene diseñar con especial cuidado:

- **Exploración de menú:** Como mencionamos, mejor por categorías para no saturar. Dentro de una categoría, limitar a ~5 ítems por página de flujo o por mensaje. Si hay más, paginar o agrupar (los Flows oficiales podrían tener varias pantallas consecutivas para sub-categorías). Incluir descripciones breves de cada plato (en WhatsApp la brevedad es vital, pero tampoco dejar solo el nombre si no es obvio). Precios siempre visibles junto al nombre para que el cliente no tenga que preguntar.
- **Personalización de pedidos:** Si el restaurante ofrece opciones (ej. término de cocción, ingredientes extras, sin sal, etc.), el flujo debe contemplar esas decisiones. Un Flow visual permite checkboxes para “Agregar extra queso” o un menú para término de carne. En chatbots, se puede hacer con preguntas encadenadas (“¿Desea ingredientes extra? 1.Queso, 2.Tocineta, 3.Ninguno”). Al final, recapitular todas las personalizaciones en la confirmación.

- *Pagos*: Si se habilita pago dentro de WhatsApp, asegurar la **seguridad** y confianza. WhatsApp Flows en algunos mercados podrían integrarse con WhatsApp Pay (por ejemplo, India con UPI). Donde no, es válido enviar un link de pago de una plataforma conocida (ej. MercadoPago, PayU) tras confirmar el pedido. Indicar claramente al cliente qué hacer: “A continuación te enviaremos un link de pago seguro. Una vez que completes el pago, recibirás confirmación aquí mismo.”. Y efectivamente monitorear vía webhook para confirmar y enviar “Pago recibido, ¡gracias! Estamos preparando tu orden.”. Si el pago es contra entrega (cash o datáfono), también indicarlo.
- *Tracking de delivery*: Los usuarios aprecian saber el estado de su orden. Si el sistema de delivery de Weats.ai puede enviar actualizaciones (p. ej., “El repartidor Juan ha recogido tu pedido y se dirige a tu dirección.”), integrarlo con WhatsApp. Esto puede ser mediante **notificaciones automáticas** en el chat (usando mensajes plantilla transaccionales) o incluso permitiendo que el usuario solicite el estado en cualquier momento con una palabra clave “ESTADO”. Una práctica de UX es proveer un **URL de seguimiento** (tracking) después de la compra, pero en lugar de un link externo, se puede ofrecer en el mismo WhatsApp: “Escribe *ESTADO* en cualquier momento para conocer el estado de tu entrega.”. El bot entonces podría responder con algo como “ Tu pedido #12345: *En ruta al destino*. Estimamos 10 min para entrega. [Ver mapa]”. El “[Ver mapa]” podría ser un enlace Google Maps con la ubicación del restaurante o un mensaje de ubicación del repartidor (aunque este último requeriría que el repartidor comparta su Live Location, lo cual ya es más manual). Lo importante es que el cliente se sienta informado y tranquilo.
- *Atención al cliente y reclamaciones*: Además del flujo feliz, hay que considerar situaciones como “mi pedido llegó incorrecto” o “se retrasó”. Tener preparada una ruta en el chatbot/Flow para estas quejas. Por ejemplo, en el menú principal, una opción “ Tengo un problema con mi pedido” que priorice la transferencia a un agente humano. Mientras más rápido se canalice un problema a solución (sea un reenvío de plato, un reembolso, etc.), mejor la experiencia. También se pueden automatizar ciertas cosas: si el cliente selecciona “Problema: Pedido incompleto”, quizás el bot podría responder “Lamentamos el inconveniente. ¿Falta algún ítem o está incorrecto? Por favor describe brevemente:” y luego pasar eso a un agente con contexto.


En general, la UX conversacional en restaurantes debe replicar la experiencia de servicio eficiente: **rápido, amable, personalizado y sin complicaciones**. Un último consejo: **testear con usuarios reales**. Hacer pruebas de usabilidad con unas cuantas personas (que no sepan cómo funciona el flujo) para ver dónde se traban o qué preguntas tienen, y ajustar en consecuencia. A veces un mensaje que es claro para el diseñador no lo es para el cliente promedio; iterar la redacción y el orden de pasos mejorará sustancialmente la tasa de finalización de los flujos.

Por las características de WhatsApp, un buen chatbot o Flow en esta plataforma puede incluso superar a una app nativa en satisfacción, porque brinda *inmediatez* y *comodidad*. Los clientes valoran poder realizar su pedido con solo enviar un mensaje, sin descargar nada adicional ³⁶, y tener la confianza de que si necesitan algo más, pueden escribir por el mismo medio. Aplicando estas buenas prácticas, la experiencia conversacional se sentirá natural, útil y acorde al estilo del restaurante, lo cual se traduce en mayor conversión de pedidos y fidelización.

Ejemplos de flujos efectivos en otras industrias (retail, banca, salud, ecommerce)

Para inspirar el diseño de la experiencia en Weats.ai, es útil analizar cómo se están utilizando flujos conversacionales y WhatsApp Flows en diversos sectores. A continuación, presentamos ejemplos de casos de uso en **retail/e-commerce**, **banca/finanzas** y **salud**, que muestran ideas aplicables:

- **Retail / E-commerce (comercio minorista):** Las marcas de retail están aprovechando WhatsApp para ofrecer **asistentes de compra personal** y soporte post-venta. Un ejemplo destacado es el uso de Flows para **shopping guiado**: en lugar de abrumar al cliente con un catálogo enorme, se le conduce por preguntas para acotar lo que busca. Imaginemos una tienda de ropa deportiva. Al escribir “quiero zapatillas para correr”, el bot puede iniciar un Flow preguntando el género (hombre/mujer), el tipo de pisada o superficie, rango de precio, etc. Conforme el cliente responde estas selecciones (con botones/dropdowns), el Flow muestra opciones que coinciden. Al final, el cliente ve 2-3 productos recomendados *como si tuviera un asesor personal*, y puede escoger uno para comprar ³⁷. Este enfoque de “quiz” interactivo aumenta la probabilidad de conversión porque simplifica la decisión y la personaliza. Otro uso en retail es el **seguimiento de pedidos**: empresas de e-commerce implementan Flows sencillos donde el cliente ingresa su número de orden y automáticamente recibe una actualización de estado (ej. “Tu pedido fue enviado, llegará aprox. el 5 de Nov”) ³⁸. Esto reduce carga al call center y da satisfacción inmediata al cliente. Asimismo, algunos retailers utilizan WhatsApp para **consultas de stock**: por ejemplo, una tienda de electrónica con un bot donde preguntas “¿Tienen disponible el modelo X de TV?” y el bot – integrado al inventario – responde al instante. Aunque podría ser conversacional, también puede hacerse con un Flow donde el cliente selecciona la categoría, luego el modelo, y el sistema le indica disponibilidad y quizás ofrece opciones: “Está disponible en color negro. ¿Deseas comprarlo? Te enviamos link de pago.”. En cuanto a *promociones*, marcas de retail han ejecutado campañas en WhatsApp combinadas con Flows: por ejemplo, enviar un mensaje masivo por WhatsApp con un catálogo interactivo (aprovechando **mensajes de lista o incluso mensajes multimedia tipo carousel dentro de los límites de WhatsApp Business**). Un caso real citado es el de marcas de belleza como Nivea que obtuvieron más del 200% del alcance objetivo de su campaña usando WhatsApp ³⁹, aunque eso fue con mensajes promocionales simples. Con Flows, se puede llevar más lejos: un cliente recibe “Rebajas de Black Friday – toca aquí para ver ofertas”, toca, se abre un Flow con categorías de productos en oferta, ve y elige, y termina generando un pedido, todo en chat. Esto incrementa las ventas impulsivas dado lo fácil del proceso. Para e-commerce puro, un flujo útil es el de **recuperación de carrito abandonado**: si un cliente dejó artículos sin comprar en la web y dio su contacto, se le puede enviar un WhatsApp: “Tienes artículos pendientes en tu carrito. ¿Quieres completarlo ahora?” con un botón que lanza un Flow mostrando esos artículos y permitiendo confirmarlos y pagar. Esto convierte a WhatsApp en un canal de cierre de venta muy poderoso.
- **Banca / Fintech:** El sector financiero, tradicionalmente formal, está incorporando experiencias tipo Flows para simplificar trámites que solían ser engorrosos. Un gran ejemplo es la **verificación de identidad (KYC)** a través de WhatsApp. Bancos y fintechs han empezado a permitir a sus clientes enviar documentos y datos mediante Flows seguros en WhatsApp, en lugar de tener que ir físicamente o hacer largas formas web. Por ejemplo, un Flow de KYC pedirá: “Toma una foto de tu identificación” (el usuario la sube en el Flow), luego “Ahora una selfie en tiempo real” (el usuario puede activar la cámara y enviarla, todo dentro del chat), después “Comparte un comprobante de domicilio”,

etc. Mediante validaciones automáticas, se puede checar que las imágenes sean claras y cumplan requisitos antes de dejarlas subir ⁴⁰. Todo este proceso guiado reduce mucho los tiempos: empresas han logrado transformar un trámite de días en algo de *minutos* vía WhatsApp. Otro caso en banca es la **solicitud de préstamos instantáneos**. Por ejemplo, un banco podría notificar: “¡Pre-aprobamos un préstamo de \$5,000! ¿Te interesa?” y si el cliente escribe “Sí”, un Flow le muestra opciones de monto a tomar, plazos, y le permite aceptar términos y condiciones ahí mismo, generando la aprobación al momento ⁴¹. Esto elimina formularios en sucursal. También seguros: compañías de seguros usan Flows para cotizaciones rápidas – el cliente elige tipo de seguro, responde 2-3 preguntas, y recibe su cotización personalizada sin hablar con un agente ⁴². En todos estos casos financieros, la clave es la **simplicidad y lenguaje claro** en los flujos, porque se lidia con datos sensibles. La inspiración para Weats.ai aquí puede no ser directa en cuanto a contenido, pero sí en cuanto a *confianza*: estos ejemplos demuestran que incluso para transacciones delicadas (dinero, identidad) la gente confía en un flujo bien diseñado en WhatsApp. Por tanto, para pagos en Weats o manejo de datos (direcciones, etc.), se puede emplear un tono y pasos similares para infundir seguridad (p. ej., iconos de candado , mensajes de “tus datos están seguros”, etc., como haría un banco).

- **Salud (clínicas, hospitales, bienestar):** En salud, WhatsApp está funcionando como un canal para mejorar la atención al paciente. Un uso común es la **programación de citas médicas** mediante Flows. Antes, para agendar cita uno debía llamar y esperar. Ahora, clínicas implementan un número WhatsApp donde un bot solicita: especialidad requerida, nombre del paciente, fecha preferida, y ofrece directamente las horas disponibles para elegir ⁴³ ⁴⁴. El paciente toca la hora que le conviene, confirma, y el Flow reserva la cita e incluso envía un recordatorio. Esto es 24/7 y sin intervención humana, liberando a recepcionistas. También se usa en laboratorios para entregar resultados: el paciente recibe un WhatsApp “Tus análisis están listos, responde 1 para ver resultados” – tras verificación (un código o DOB), el bot envía un PDF del resultado o un Flow con gráficos resumidos. En salud mental, hay chatbots que hacen preguntas de *screening* (como un breve cuestionario de síntomas) de forma conversacional, antes de recomendar consultar con un profesional. La enseñanza para Weats en este rubro es sobre **sensibilidad y claridad**: los flujos de salud suelen incorporar lenguaje empático y opciones muy explícitas (“1. Reprogramar cita, 2. Cancelar, 3. Hablar con asistente”). Para nuestro caso, podríamos aplicar esa misma empatía en flujos de servicio al cliente: por ejemplo, si un cliente se queja en Weats de un problema, un flujo podría decir “Lamentamos mucho este inconveniente . Para resolverlo, por favor indícanos... [opciones]”, similar a como un bot médico trataría con cuidado a un paciente frustrado. Otro punto de salud es la **confidencialidad** – importante en Fintech también – siempre hay que asegurarle al usuario qué se hace con sus datos (un restaurante al menos debe asegurar que su info no se comparte, etc., quizá en el footer de un Flow o en mensajes de privacy).

- **Otros sectores:** Aunque el enfoque principal son los anteriores, vale la pena mencionar brevemente un par de ejemplos más:

- **Gobierno/Servicios públicos:** Algunos municipios han lanzado bots de WhatsApp para trámites simples (pago de impuestos, agendar citas en oficinas gubernamentales). Por ejemplo, la ciudad de Buenos Aires tiene un chatbot por WhatsApp que permite reservar turno para renovar licencia de conducir. Esto es similar a reservar mesa: se selecciona fecha, hora y sede en un Flow. La lección aquí es cómo manejar **altas concurrencias** – si Weats.ai escala, puede haber muchos usuarios simultáneos. Los

Flows de Meta manejan escalabilidad en la nube de WhatsApp, pero los chatbots custom deben estar optimizados para concurrencia.

- **Educación:** Instituciones educativas usan WhatsApp para responder FAQs de admisiones o incluso inscribir a estudiantes en cursos. Un Flow podría recolectar datos de inscripción paso a paso. Esto muestra la versatilidad: cualquier proceso de formulario se puede traducir a Flow. En Weats.ai, formularios como registro de un nuevo restaurante aliado o onboarding de un repartidor se pueden hacer vía WhatsApp Flow, inspirándonos en estos casos educativos.

En suma, **WhatsApp se está convirtiendo en una plataforma omnipresente** para interacción cliente-servicio en muchas industrias. Los ejemplos de retail, banca y salud demuestran que los usuarios están dispuestos a hacer desde compras hasta trámites complejos mediante flujos conversacionales en WhatsApp siempre que estén bien diseñados. Para Weats.ai, la inspiración concreta sería: - Ofrecer una experiencia de **compra conversacional tan fácil como chatear con un amigo**, semejante a la de retail personal shopper. - Lograr la **confianza y eficiencia** de un banco, por ejemplo en pagos o en manejo de información (que el cliente sienta que pedir un domicilio es tan seguro y claro como hacer una transacción bancaria por chat). - Brindar la **rapidez y disponibilidad** que se ve en salud y otros (un asistente que nunca duerme y responde instantáneo, dentro de lo posible).

Con estos referentes, pasemos a recomendaciones específicas para la implementación en Weats.ai considerando cada actor involucrado.

Recomendaciones específicas para el diseño de flows en Weats.ai (cliente, restaurante, repartidor)

Weats.ai se perfila como una plataforma de entrega de comida a domicilio impulsada por IA y WhatsApp, con un modelo “win-win-win” para clientes, restaurantes y repartidores (runners). Esto implica que habrá interacciones conversacionales distintas para cada tipo de usuario en el ecosistema. A continuación, se proponen recomendaciones concretas de flujos (ya sean Flows visuales oficiales, chatbots personalizados, o combinaciones) adaptados a las necesidades de cada rol en la plataforma:

1. Flujos para el cliente (usuario que realiza pedidos)

El cliente es el centro de la experiencia – quiere descubrir restaurantes/platos, hacer su pedido fácilmente, pagar y recibir su entrega con confianza. Para ellos recomendamos:

- **Flujo de inicio / onboarding del cliente:** Cuando un nuevo cliente escribe a Weats.ai por primera vez, es ideal tener un pequeño Flow de bienvenida que lo registre. Por ejemplo, un Flow que pida autorización para usar sus datos y preferencia de idioma, e incluso opcionalmente su dirección frecuente. *Pantalla 1:* “¡Hola! Bienvenido a Weats, tu asistente de pedidos. Para empezar, indícanos tu nombre:” (campo de texto). *Pantalla 2:* “Gracias, {Nombre}. ¿Cuál es tu dirección o zona principal de entrega?” (puede ser un input de texto o pedir ubicación en un paso fuera del Flow si es complejo). *Pantalla 3:* “¿Qué tipo de comida te gusta? (Selecciona tus preferencias para recomendaciones)” con opciones multi-selección (checkbox) de categorías: Italiana, Rápida, Japonesa, Saludable, etc. *Pantalla 4:* “¡Listo! Podemos personalizar tu experiencia con esos datos. Presiona **Comenzar** para ver opciones de pedido.” (botón Finalizar). Este Flow inicial (máximo 4 pantallas) captura información útil para luego filtrar restaurantes o ofrecer sugerencias al cliente.

Además, rompe el hielo de forma interactiva. Si el cliente prefiere saltarlo, siempre debe haber opción de *Skip* ("O puedes escribir *OMITIR* para saltar este paso").

- **Flujo de exploración de menú y pedido conversacional:** Este es el corazón. Inspirándonos en los ejemplos anteriores, proponemos un **Flow multistep para hacer pedidos**. Lo ideal es que sea dinámico por restaurante, pero dado el límite de 8 pantallas, quizá se deba dividir por secciones. Un enfoque es tener un Flow por cada restaurante, lanzado cuando el cliente decide un restaurante específico. Por ejemplo: el cliente escribe "Quiero pedir a Restaurante Pizzería XYZ". El bot puede responder con un *mensaje interactivo de lista* de restaurantes cercanos si no especificó cuál, o si ya lo hizo, directamente iniciar el Flow correspondiente al menú de XYZ. Dentro de ese Flow (que podría llamarse "Flow Pedido XYZ"), las pantallas serían:
 - **Elegir categoría:** lista de las categorías de menú de XYZ (Entrada, Pizzas, Bebidas, Postres...).
 - **Elegir ítem:** según la categoría seleccionada, mostrar 5 platos de esa categoría con nombre y precio (si hay más de 5, tal vez sub-dividir o que el Flow use scroll interno). El cliente selecciona uno.
 - **Confirmar o personalizar ítem:** si el ítem elegido tiene opciones, aquí presentarlas. Ej: "Selecciona tamaño de la pizza" con opciones radio (Personal, Mediana, Familiar), luego quizás otra pantalla: "Ingredientes extra" con checkboxes (tocino, extra queso, etc.) – esta podría ser la pantalla 4 si hay extras.
 - **¿Agregar otro ítem?:** Se podría en la misma pantalla preguntar si desea algo más con botones "Añadir otro plato" o "Proceder al pago". Pero podría ser mejor finalizar el Flow del primer ítem y luego, de regreso en chat, preguntar si añadir otro. Aquí hay un detalle: los Flows oficiales terminan cuando acabas sus pantallas; podríamos diseñar que tras finalizarlos, el bot en modo texto pregunte si seguir pidiendo. Alternativamente, usar un Flow monolítico para múltiples ítems, pero eso puede salirse de límite. Quizá más fácil: **Flow por cada plato** – es decir, el cliente va agregando platos uno por uno mediante Flows repetitivos. Esto requiere diseñar la lógica del bot fuera de los Flows para sumar todo.
 - **Confirmación de pedido y pago:** Una vez el cliente indicó que ya no quiere agregar más, se inicia un **Flow de Checkout** separado. En ese Flow: mostrar resumen de pedido (lista de ítems con cantidad, precio, total) y campos para método de pago (radio buttons, ej. "Efectivo", "Tarjeta/Link"), propina si aplica, y confirmación final con botón. Este Flow de checkout puede también pedir confirmación de dirección de entrega (auto-rellenada con la que dio en onboarding pero editable).
 - **Confirmación final:** Tras enviar el pedido, mostrar pantalla de agradecimiento con número de orden y ETA estimado.

Este enfoque combina Flows y chat: Flows para la parte estructurada (elegir platos, llenar detalles) y la conversación normal para moverse entre Flows ("¿quieres añadir algo más?" etc.). Una ventaja es que los Flows pueden aprovechar componentes como date-picker si es un **pedido programado** (p. ej., "¿Para qué hora deseas tu pedido? [selecciona fecha/hora]").

También hay que manejar **casos especiales**: Si un usuario en medio del proceso pregunta algo (ej. "¿la pizza familiar alcanza para 4 personas?"), el bot debe reconocer esa desviación. Se podría integrar una capa de AI (por eso Weats tiene IA) para responder a preguntas frecuentes contextuales, o en su defecto ofrecer "Te

comunico con un agente para consultas especiales". La IA de cliente debe estar entrenada en FAQs de restaurantes, ingredientes, tiempos de entrega, etc., para complementar el flujo.

- **Flujo de estado de pedido (tracking):** Aunque se puede manejar con simples notificaciones push, también se podría ofrecer un **Flow de tracking** si el cliente solicita información detallada. Por ejemplo, el cliente envía "ESTADO" y el bot inicia un Flow "Seguimiento de Pedido" que muestra una pantalla con un indicador (tal vez texto pseudo-gráfico) del estado: " Pedido en preparación" o " Tu pedido está en camino", quizás con etapas completadas. Incluso se podría incorporar un **mapa estático** (imagen) con la ubicación del restaurante y del destino marcada, aunque WhatsApp no tiene mapita dinámico dentro del Flow. Una alternativa es proporcionar un enlace de tracking externo, pero ya que la filosofía es WhatsApp-first, quizás mejor mantenerlo a base de texto actualizaciones. Este Flow de tracking podría tener un botón de "¿Problema con el pedido?" llevando a asistencia.
- **Flujo de feedback y promociones post-venta:** Después de completarse un pedido, Weats.ai puede automatizar un Flow de **recolección de feedback**. Por ejemplo, a las 2 horas de entregado, el cliente recibe un mensaje: "¿Cómo calificarías tu experiencia con Weats y el restaurante? ¡Tu opinión nos ayuda a mejorar!" con un botón "Dar Feedback". Al tocar, Flow de 2-3 pantallas: calificación 1-5 estrellas (se puede simular con 5 botones ★), un campo opcional de comentario, quizás una pregunta "¿Recomendarías este servicio? Sí/No". Al finalizar, se le puede ofrecer un cupón: "Gracias por tu feedback. Aquí tienes un código de 10% para tu próximo pedido.". Esto cierra el ciclo y fomenta la retención.

Otra idea es un Flow de **referidos o lealtad**: "Invita a un amigo y gana créditos. Ingresas el número de WhatsApp de tu amigo:" etc., pero esto ya es adición.

En general, para los clientes, las recomendaciones se centran en: flujo de pedido eficiente, comunicación transparente, y facilidad para soporte. Dado que Weats.ai busca ser muy rápido (30 segundos por pedido según su visión), cada paso del flujo cliente debe estar optimizado y posiblemente apoyado por IA para completar detalles (por ejemplo, la IA podría predecir "¿Quieres lo de siempre?" acortando pasos).

2. Flujos para el restaurante (socios comerciales en la plataforma)

El restaurante (dueño/administrador) interactuará con Weats.ai seguramente para recibir los pedidos y actualizarlos. Tradicionalmente esto podría hacerse en una app panel, pero dado el enfoque WhatsApp-first, es probable que los restaurantes también se comuniquen vía WhatsApp con la plataforma, quizá mediante un *bot de restaurante* o incluso usando un grupo/trillaje con un agente humano. Pero asumiendo que Weats quiere automatizar también ese lado, algunas ideas:

- **Flujo de recepción y confirmación de pedido:** Cuando un cliente hace un pedido a un restaurante, el restaurante necesita confirmarlo (o rechazarlo si no puede cumplir). Weats.ai podría implementar un **Flow dirigido al restaurante** para procesar cada nuevo pedido. Por ejemplo, tan pronto entra un pedido, el sistema envía un mensaje WhatsApp al número del restaurante (quizá al dueño o encargado en ese momento) con un resumen: "Nuevo Pedido #12345: 1x Pizza Margarita, 2x Coca Cola – Entrega 8:30 PM a Calle 123. Total \$25.000." y dos botones: "**Aceptar**" o "**Rechazar**". Al tocar, se abre un Flow (podría ser un Flow ultra corto integrado en la notificación misma) con pantallas: si Aceptar, pasar a pantalla "Indica tiempo estimado de preparación (en minutos): [input number]" y

luego finaliza con “¡Gracias! Pedido confirmado.”. Esa información de tiempo se devuelve a Weats.ai para notificar al cliente. Si el restaurante toca Rechazar, el Flow podría requerir un motivo: pantalla con opciones “Motivo de rechazo: 1. Ingrediente agotado, 2. Demasiada demanda, 3. Otro” y quizás un campo de texto para detalle. Luego finaliza y Weats.ai sabrá que debe informar al cliente y quizás redirigirlo a otra opción. Este mini-Flow facilita que el restaurante no tenga que teclear nada para confirmar pedidos – solo unos toques rápidos en WhatsApp. Muy conveniente ya que el personal de cocina puede no estar frente a un PC.

- **Flujo de actualización de estado de pedido:** Similar al tracking del cliente pero desde el lado restaurante. Podría ser que el restaurante envíe mensajes o comandos para decir “pedido listo”. Un flujo puede simplificarlo: Weats podría enviar al restaurante un mensaje cuando sea hora de actualizar, por ejemplo “¿El pedido #12345 ya está listo para recoger?” con botón “Marcar como Listo”. Ese botón podría disparar un Flow de confirmación: “Confirmar que el pedido #12345 está listo para entrega al repartidor.” [Confirmar]. Al confirmar, Weats notifica al runner y al cliente. Si se quiere más detalle, podría haber estados intermedios (preparación iniciada, etc.), pero probablemente con un *marcar listo* sea suficiente. Nuevamente, esto ahorra que el restaurante llame o escriba manualmente.
- **Flujo de gestión de menú o horario:** Para mantener la plataforma actualizada, se podría permitir a restaurantes actualizar ciertos datos vía WhatsApp. Un ejemplo: un Flow de **cambio de disponibilidad de plato**. Si ese día se acaba un ingrediente clave, el restaurante podría abrir un Flow tipo “Actualizar menú” donde seleccione el plato que quiere desactivar temporalmente. Pantalla 1: lista de sus platos, pantalla 2: elegir acción (Desactivar/Activar), pantalla 3: confirmar. Eso mandaría la info al backend de Weats para no ofrecer ese plato a clientes. Otro: Flow para **cambiar horario de atención** (“Hoy cerramos más temprano por mantenimiento”). Esto puede ser engorroso de hacer por chat manual, así que un Flow brinda una interfaz controlada. Sin embargo, hay que tener cuidado, pues un error podría causar confusión; tal vez sea mejor que esas cosas las maneje un dashboard admin, pero si la visión es todo por WhatsApp, se puede habilitar con autenticación (por seguridad, el Flow del restaurante puede requerir un PIN o password al iniciar, para que solo el dueño autorizado haga cambios).
- **Notificaciones y soporte al restaurante:** No todos los contactos con restaurantes serán flujos. A veces necesitarán soporte de Weats (ej. “No llegó el repartidor”). Debe haber un canal para que pidan ayuda, quizás un chatbot separado o en el mismo hilo con ciertos comandos. Un Flow útil podría ser un **Flow de reporte de incidente**: el restaurante escoge “Reportar un problema”, selecciona el tipo (runner retrasado, cliente no recoge, etc.) y Weats.ai procesa eso alertando a un equipo humano. Esto aseguraría que se capturan todos los datos necesarios en un incidente sin llamadas confusas.

En resumen para restaurantes: **automatizar la confirmación de pedidos y actualizaciones** es primordial. Les ahorra tiempo y reduce errores de comunicación. Las recomendaciones son mantener esos flujos para restaurante muy breves y directos, ya que estarán ocupados. Ideal que puedan completarlos con uno o dos taps. Y siempre dar feedback inmediato (“Pedido confirmado exitosamente” o “Recibimos tu cambio en el menú”). Dado que los restaurantes pueden no ser tan tech-savvy, los mensajes deben ser muy claros en indicaciones. Probablemente los Flows oficiales de WhatsApp encajan bien aquí, porque presentan una interfaz sencilla que el personal puede manejar sin capacitación especial (solo es WhatsApp al fin y al cabo).

3. Flujos para el runner/repartidor (entregas)

El repartidor, llamado *runner* en Weats, es el tercer pilar. Ellos se encargan de recoger la orden en el restaurante y llevarla al cliente. Weats.ai podría tener muchos repartidores independientes conectados via WhatsApp para asignarles pedidos. Gestionar a los runners por WhatsApp es todo un desafío pero factible con automatización. Algunas recomendaciones:

- **Flujo de asignación de pedido al repartidor:** Cuando un pedido está listo para envío, el sistema debe encontrar un repartidor disponible y notificarle. Esto se puede hacer tipo broadcast a varios o a uno en específico. Un posible flujo: Weats envía un mensaje al runner más cercano: “ *Nueva entrega asignada:* Restaurante: Pizzería XYZ (Calle 50 #10), Destino: Calle 123 #45, Pago: \$5.000. ¿Aceptas la entrega? Sí/No”. El runner toca Sí, y eso confirma que la toma. Si no responde en X segundos, se podría ofertar a otro. Este proceso puede ser manejado con un Flow para asegurar respuesta rápida: el runner toca Sí -> Flow confirma “Asignada a ti. Dirígete al restaurante.”; toca No -> Flow devuelve “Entendido, no te preocupes. Buscando otro repartidor.”. Incluso se puede hacer con simple botones sin Flow, pero un Flow permite, por ejemplo, si dice No, preguntar el porqué (descanso, moto averiada, etc.) para registro.
- **Flujo de navegación y actualización durante entrega:** Aunque la navegación (mapas, rutas) normalmente se haría fuera (Google Maps app), el runner podría necesitar notificar ciertos eventos. Por ejemplo, al llegar al restaurante, podría enviar “● Recogí el pedido #12345”. Un Flow puede simplificarlo con un botón “Pedido recogido” que Weats le manda una vez aceptó el pedido. Lo toca -> Flow: “¿Confirmas que recogiste el pedido del restaurante? [Sí]” -> Listo, envía confirmación al sistema. Luego al llegar al cliente, otro botón “Entregado” -> Flow: “¿El pedido fue entregado exitosamente? [Sí, entregado] [Hubo un problema]”. Si “Hubo un problema”, abrir subpaso: opciones “Cliente no estaba, No encuentra dirección, Otro” y campo texto, luego finalizar enviando esa alerta a soporte. Este Flow de finalización captura la *prueba de entrega*. Incluso se podría incorporar en el Flow la opción de subir una foto (ej. foto de la puerta o del cliente recibiendo) como comprobante en caso de ausencia ⁵ ⁶ . Esa imagen queda adjunta al registro del pedido. Esto es algo que algunas apps hacen (foto del paquete entregado), y WhatsApp lo soporta.
- **Flujo de inicio/cierre de turno del repartidor:** Si los runners no están siempre disponibles 24/7, se les puede dar control para indicar cuando están activos. Un sencillo Flow podría servir para *loguearse* al comenzar a trabajar: “¿Iniciar turno? [Sí] [No]” con [Sí] quizás pidiendo su ubicación actual (para ubicarlos en el mapa del sistema) y marcándolos como disponibles. O simplemente un comando “Disponible” que setea un flag. Al terminar el día, “Finalizar turno [Confirmar]”. Esto podría ser conversacional sin Flow, pero usar Flows aseguran que confirmen y tal vez registren cuántas horas estuvo activo.
- **Comunicación con runner:** Al igual que con restaurantes, debe haber forma de pedir ayuda. Ej: si su moto se daña durante una entrega, enviar “Problema” y que un Flow le pregunte qué ocurre y notifique a soporte para reasignar el pedido a otro. En general, la interfaz con repartidores debe ser lo más ágil posible, ya que muchos estarán conduciendo. Por seguridad, conviene que la interacción sea con toques y mensajes breves (no ponernos a pedir texto largo mientras maneja). WhatsApp incluso permite mensajes de voz, pero procesarlos automáticamente es complejo; mejor quedarse con botones.

- **Información y motivación:** Weats podría también usar WhatsApp para enviar a los runners su resumen de día (ganancias, pedidos completados). Quizás no un Flow sino un mensaje diario: “Hoy completaste 5 entregas, ganaste \$X. ¡Buen trabajo! ”. Si se quisiera, un Flow para “Consultar balance actual” donde el runner toca un botón y recibe su saldo acumulado, etc. Son añadidos que mejoran la transparencia y confianza del colaborador con la plataforma.

En síntesis para repartidores: **Rapidez y simplicidad** son cruciales. Todos los flujos deben requerir mínimo esfuerzo cognitivo (idealmente one-tap confirmations). Además, como son colaboradores, se les puede dar cierto control (entrar/salir turno) y soporte dedicado. Al usar WhatsApp, se aprovecha que muchos repartidores ya usan WhatsApp constantemente, así no necesitan aprender a usar otra app. Sin embargo, hay que cuidar la confiabilidad: un problema podría ser la conectividad o si el runner tiene el teléfono apagado. Por eso, la plataforma debe tener fallback (quizá SMS o llamada si no responde en WhatsApp). Pero desde el diseño de flujo, asumir siempre confirmaciones explícitas en cada paso importante. Y mantener un log de tiempo (WhatsApp marca hora de cada mensaje) que sirve para métricas de performance (ej., cuánto tardó desde recogida hasta entrega – calculable con las marcas de “Pedido recogido” y “Entregado”).

Enfoque visual vs. conversacional: ¿cuándo usar cada uno?

Al diseñar estos flujos en Weats.ai, cabe hacer una reflexión final sobre **enfoques visuales (Flows oficiales)** vs **enfoques conversacionales tradicionales (bot de texto)**. No es que uno sea inherentemente mejor que el otro; de hecho, se complementan.

Ventajas de los Flows visuales: ofrecen una experiencia **estructurada y sin ambigüedad**. El usuario ve pantallas con campos claros y botones, similar a llenar un formulario dentro de WhatsApp. Esto reduce la posibilidad de error (no hay que interpretar lenguaje libre) y agiliza la interacción (un tap es más rápido que tipear una frase entera). En casos como formularios de registro, encuestas de satisfacción, o pasos de checkout con muchos datos, los Flows brillan porque condensan varias entradas en un flujo coherente y amigable ⁴⁵ ⁴⁶. Para usuarios menos hábiles tecnológicamente, puede ser más fácil tocar “Opción 2” que tener que escribir “Opción 2” manualmente. Además, la interfaz visual da una sensación de “app-like” dentro de WhatsApp, lo cual profesionaliza la percepción del servicio.

Ventajas de la conversación libre (chatbot): ofrece **flexibilidad y naturalidad**. Hay usuarios que prefieren teclear lo que desean en sus propias palabras (“Quiero repetir mi pedido anterior”) en vez de navegar opciones. Un buen motor conversacional con NLP puede captar estas intenciones y saltar a la parte relevante del flujo, ofreciendo una experiencia más personalizada. Asimismo, la conversación tradicional permite manejar mejor las desviaciones y el contexto: por ejemplo, si el usuario cambia de tema de repente (“por cierto, ¿qué promociones tienen hoy?”), un chatbot con IA puede responder a eso y luego retornar al flujo principal. Con un Flow puro, esa desviación podría romper el proceso. También, en interacciones muy sencillas (ej. “consulta mi saldo”), usar un Flow sería sobrepeso; un simple mensaje/respuesta es suficiente. Los chatbots conversacionales tienden a ser mejores para simular un diálogo humano, lo cual puede aumentar la cercanía y confianza con ciertos perfiles de clientes.

Nuestra recomendación es adoptar un enfoque **híbrido inteligente**: utilizar Flows visuales oficiales de WhatsApp para los sub-procesos bien definidos (formularios de varios pasos, elecciones de catálogo, inputs estructurados), e hilo conversacional tradicional para el *orquestador* entre esos sub-procesos y para manejo de lenguaje natural. Por ejemplo, en la experiencia del *cliente* descrita: la selección de productos y

personalización se puede implementar con Flows (mejor UI), mientras que la posibilidad de preguntar algo o de agregar otro producto se maneja conversacionalmente por el bot. Este enfoque “visual + chat” aprovecha lo mejor: la **eficiencia de la UI** y la **flexibilidad de la conversación**. Además, distribuir la lógica así puede prevenir saturar un único Flow con demasiadas ramificaciones.

Desde la perspectiva de Weats.ai, también es importante la **analítica**: WhatsApp Flows oficiales reportan métricas de completitud, etc. (p. ej., sabemos cuántos clientes abandonan en pantalla 3 de 5). Los chatbots por su parte permiten *tracking* personalizado de intenciones y mensajes. Combinando ambos, Weats puede obtener datos muy granulares para mejorar la UX continuamente.

En conclusión, **WhatsApp como plataforma conversacional** tiene un potencial enorme para revolucionar el negocio de restaurantes y delivery, permitiendo experiencias fluidas sin necesidad de apps nativas. Al investigar las capacidades de WhatsApp Flows oficiales y las soluciones personalizadas existentes, queda claro que podemos diseñar en Weats.ai una interacción donde el cliente pida comida en segundos con un par de toques, el restaurante confirme al instante de forma organizada, y el repartidor gestione la entrega eficientemente – todo dentro de WhatsApp. Los ejemplos de otras industrias nos muestran que la gente ya está adoptando estos flujos para todo tipo de tareas, por lo que comer y pedir delivery no será la excepción. Aplicando las mejores prácticas de UX conversacional (usabilidad, claridad, empatía) y aprovechando tanto las interfaces visuales de Flows como la potencia de la IA conversacional, Weats.ai puede ofrecer una **experiencia pionera**: rápida como mensaje de texto, estructurada como aplicación, y cercana como hablar con el restaurante de la esquina. Los beneficios esperados son mayores tasas de conversión de pedidos, costos operativos muy bajos (al automatizar 90%+ de las interacciones rutinarias) y satisfacción tanto de clientes como de restaurantes y repartidores al sentir un sistema ágil y “justo” para todos. En otras palabras, un ecosistema optimizado donde WhatsApp no es solo el medio, sino el **entorno integral** de la experiencia de entrega de comida.

Fuentes: Las características técnicas de WhatsApp Flows (componentes, pantallas, etc.) se han referenciado de la documentación de Meta y recursos de terceros ¹ ² ⁷. Estadísticas de mejora con Flows provienen de casos de estudio en la industria ²⁴. Ejemplos de usos en banca, retail y salud han sido tomados de análisis publicados por plataformas especializadas ⁴⁰ ³⁷ ⁴⁷, mientras que las mejores prácticas de diseño conversacional se han elaborado con base en guías de UX y experiencias reportadas en proyectos de chatbots restauranteros ⁴⁸ ⁴⁹. Estas referencias respaldan las recomendaciones aquí expuestas y demuestran la viabilidad de su implementación en 2025, año en que WhatsApp se consolida como canal preferido para una variedad de interacciones comerciales ⁵⁰ ⁵¹. Hemos integrado dichas mejores prácticas al contexto específico de Weats.ai para brindar una hoja de ruta sólida hacia una experiencia conversacional innovadora en el sector de entregas de comida.

- 1 2 24 **WhatsApp Flows: Qué es, beneficios y casos de uso claves**
<https://yoizen.com/blog/whatsapp-flows-que-es-beneficios-y-casos-de-uso-claves/>
- 3 4 15 18 45 **WhatsApp Flows: A Game-changer for Interactive Business Messaging**
<http://chakrahq.com/article/whatsapp-flows-business-api-explained/>
- 5 6 **Different Input Types Available in the WhatsApp Flow Builder | Wabo.ai**
<https://help.wabo.ai/whatsapp-flows/different-input-types-available-in-the-whatsapp-flow-builder>
- 7 11 12 22 23 **WhatsApp flow | WotNot Help Center**
<https://help.wotnot.io/build/action-blocks/whatsapp-flow>
- 8 9 28 29 30 31 32 **How to Create and Use WhatsApp Flows in Wati | Wati.io Help Center**
<https://support.wati.io/en/articles/11463180-how-to-create-and-use-whatsapp-flows-in-wati>
- 10 13 14 19 35 **WhatsApp Business Messaging: Create and manage WhatsApp Flows for customer journeys | Infobip**
<https://www.infobip.com/docs/whatsapp/whatsapp-flows/build-whatsapp-flows>
- 16 17 38 43 44 46 **WhatsApp Flows: How to Create and Use it. (With Examples) | Engati**
<https://www.engati.ai/blog/create-whatsapp-flows>
- 20 21 50 **WhatsApp Flows: Simplifying Customer Conversations**
<https://gallabox.com/blog/whatsapp-flow>
- 25 **Proyecto WPFoods: El Futuro De La Entrega Justa.pdf**
file:///file_00000000a4e061f79266383729f6e171
- 26 27 36 **Restaurant Chatbot: No-Code Tutorial | Landbot**
<https://landbot.io/blog/restaurant-chatbot>
- 33 34 **Zoko - Better Business on WhatsApp**
<https://www.zoko.io/>
- 37 40 41 42 47 **9 Game-changing ways to use WhatsApp Flows for your business | Fyno**
<https://www.fyno.io/blog/9-game-changing-ways-to-use-whatsapp-flows-for-your-business-cm3mk4ro5002pvnqr4s13zay7>
- 39 **Entendiendo WhatsApp Flows: Personalizando recorridos del cliente in-app - Infobip**
<https://www.infobip.com/es/blog/conoce-whatsapp-flows>
- 48 49 **WhatsApp Chatbot for Restaurant: Automate Orders & Payments**
<https://www.go4whatsapp.com/blog/automate-orders-payments-in-restaurant-with-whatsapp-chatbots/>
- 51 **WhatsApp Flows: Funciones y Características Clave | Sinch**
<https://sinch.com/es/blog/whatsapp-flows-caracteristica-y-funcionalidades-de-la-plataforma/>