

Construindo um chatbot com **Take Blip**

PRIMEIROS PASSOS



EBOOK

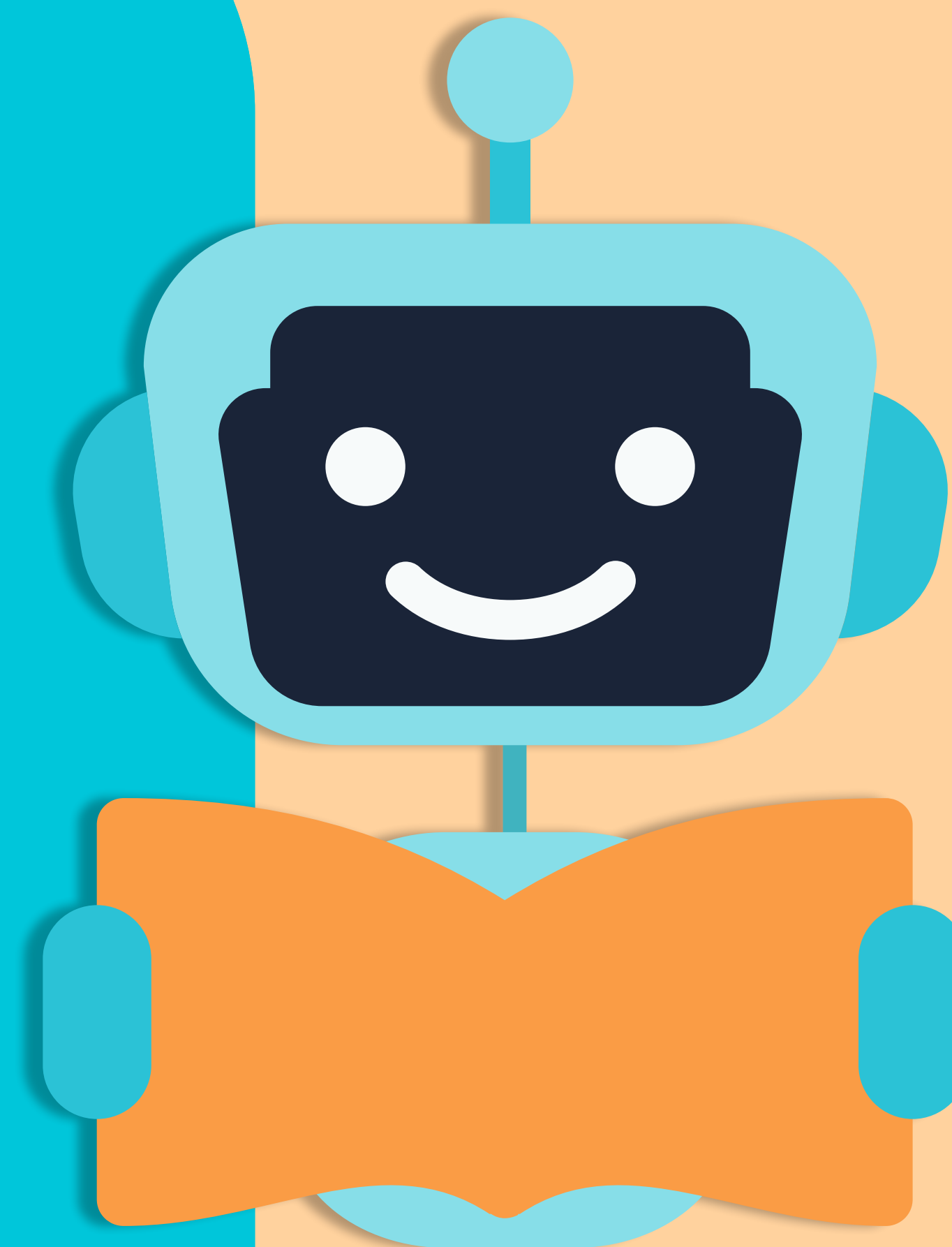
Introdução

Você já sabe que o uso de canais conversacionais — como Facebook Messenger, WhatsApp e Telegram — só tende a aumentar, não é mesmo?

De acordo com uma pesquisa realizada pela Statista, **42% da população mundial** utiliza um smartphone, e, destes, **87% usam aplicativo de mensagem**. Somente no WhatsApp foram trafegadas cerca de **100 bilhões** de mensagens por dia em 2020.

Porém, mais do que estar presente nessas canais, é importante atender o usuário de forma satisfatória. Por isso, investir em tecnologias como chatbots é fundamental para melhorar o atendimento ao cliente, otimizar processos na empresa e aumentar as vendas.

Pensando nisso, preparamos este guia básico especialmente para você que tem dúvidas sobre por onde começar a **construção de um chatbot utilizando a plataforma Take Blip**.



Neste e-book, vamos te ajudar a:

- 1** Desenhar o seu fluxo conversacional;
Conhecer as melhores práticas;
- 2** Explorar fundamentos essenciais para construção
de chatbots com Blip;
- 3** Conhecer várias outras dicas e sugestões para
que você consiga colocar seu projeto no ar!

Boa leitura!

Dica:

para aprender ainda mais, recomendamos a leitura de alguns materiais complementares, como vídeos de treinamentos e outras leituras ao longo do guia.

Canais de mensageria

Um conceito importante para a construção de chatbots e contatos inteligentes é: canais de mensageria”.

Aqui, entendemos todas as plataformas em que ocorrem trocas de mensagens, ou seja, aplicativos como o WhatsApp, Facebook Messenger, Telegram e mensagens de texto. Como as pessoas estão cada vez mais presentes nesses aplicativos, torna-se fundamental as empresas disponibilizarem atendimento por ali também.



Utilizar os canais conversacionais como ferramentas no dia a dia da empresa já deixou de ser um diferencial há muito tempo e se tornou uma obrigação. Mas sabemos que gerenciar diversos canais de comunicação, aplicativos de mensagem e telefones para contato não é simples. E é aí que entram os chatbots.

Os sistemas conversacionais como chatbots foram criados para facilitar o contato entre clientes e empresas. Além de acelerar o atendimento e, conseqüentemente, reduzir reclamações, também aumenta as chances de vendas e diminui custos operacionais.

Agora que já vimos essas informações importantes, vamos começar o projeto.



Leituras fundamentais

Selecionamos alguns conteúdos importantes que irão te ajudar durante o processo de criação do seu chatbot.

Confira:

- [Curso Criando chatbots no BliP](#)
- [Hierarquia \(ou arquitetura\) de bots e subbots](#)
- [Dicas de ferramentas](#)

Após o acesso aos materiais acima, você já pode seguir para o desenho do fluxo, mas não indicamos que você comece a construir em nossa ferramenta ainda.

Para auxiliar nesta etapa, ao longo dos passos elencados na trilha "Construindo chatbot com Take Blip — desenhando o fluxo" indicamos alguns materiais de apoio.

Todos eles estão hiperlinkados.



Construindo um chatbot com **Take Blip**

DESENHANDO O FLUXO

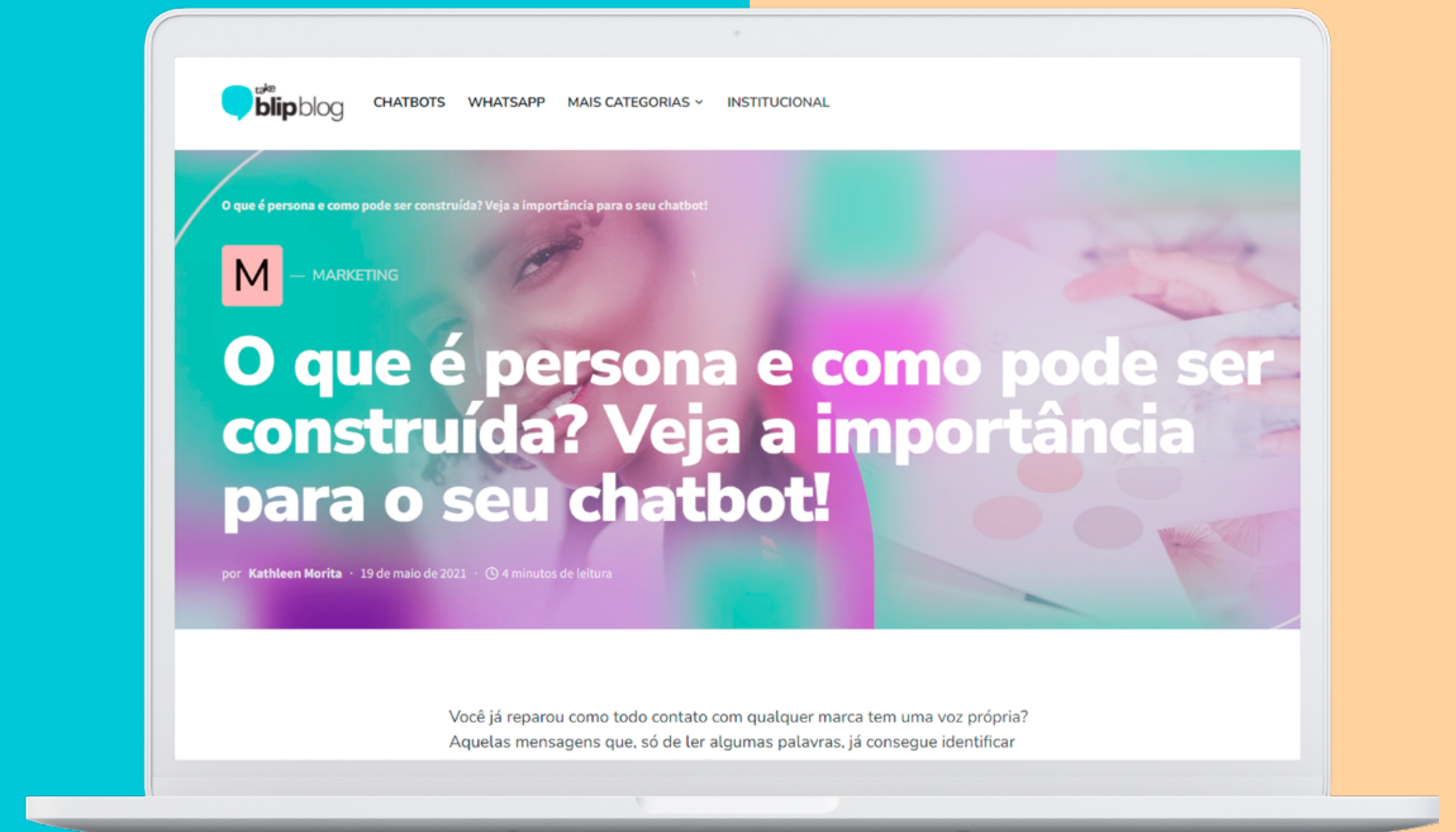


PASSO 01: planejar

Antes de construir o bot no Blip, **planeje como será o seu projeto, o desenho do fluxo e documente as etapas essenciais.** O que seu bot irá tratar? Pense nas primeiras automatizações que você gostaria.

Dica: crie uma persona para o seu bot. Apesar de não ser obrigatório, esta é uma boa prática para "humanizar" o seu bot. Pense no tom de voz, se utiliza emojis, gírias, se é mais sério etc. É interessante dar um nome também. Alinhe estes pontos junto ao seu time de marketing.

Temos um post-blog sobre [definição de persona](#) para te ajudar!



PASSO 02: desenhar o fluxo

Esquematize o fluxo conversacional do seu bot. Faça um desenho de como você imagina que o usuário irá interagir e seu bot irá tratar cada etapa. **Comece com um fluxo macro e depois parta para o micro.** Nomeie cada um dos blocos para facilitar a aplicação no Blip.

Dica de ferramentas: [draw.io](#) / [Miro](#) / [Figma](#)

Take Blip disponibiliza uma [biblioteca de componentes](#) para facilitar o desenho no Figma.

Caso você precise de ajuda, também temos templates pré-configurados e um time pronto para colocar seu chatbot no ar.



PASSO 03: primeira mensagem

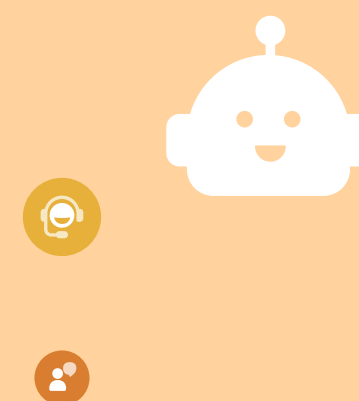
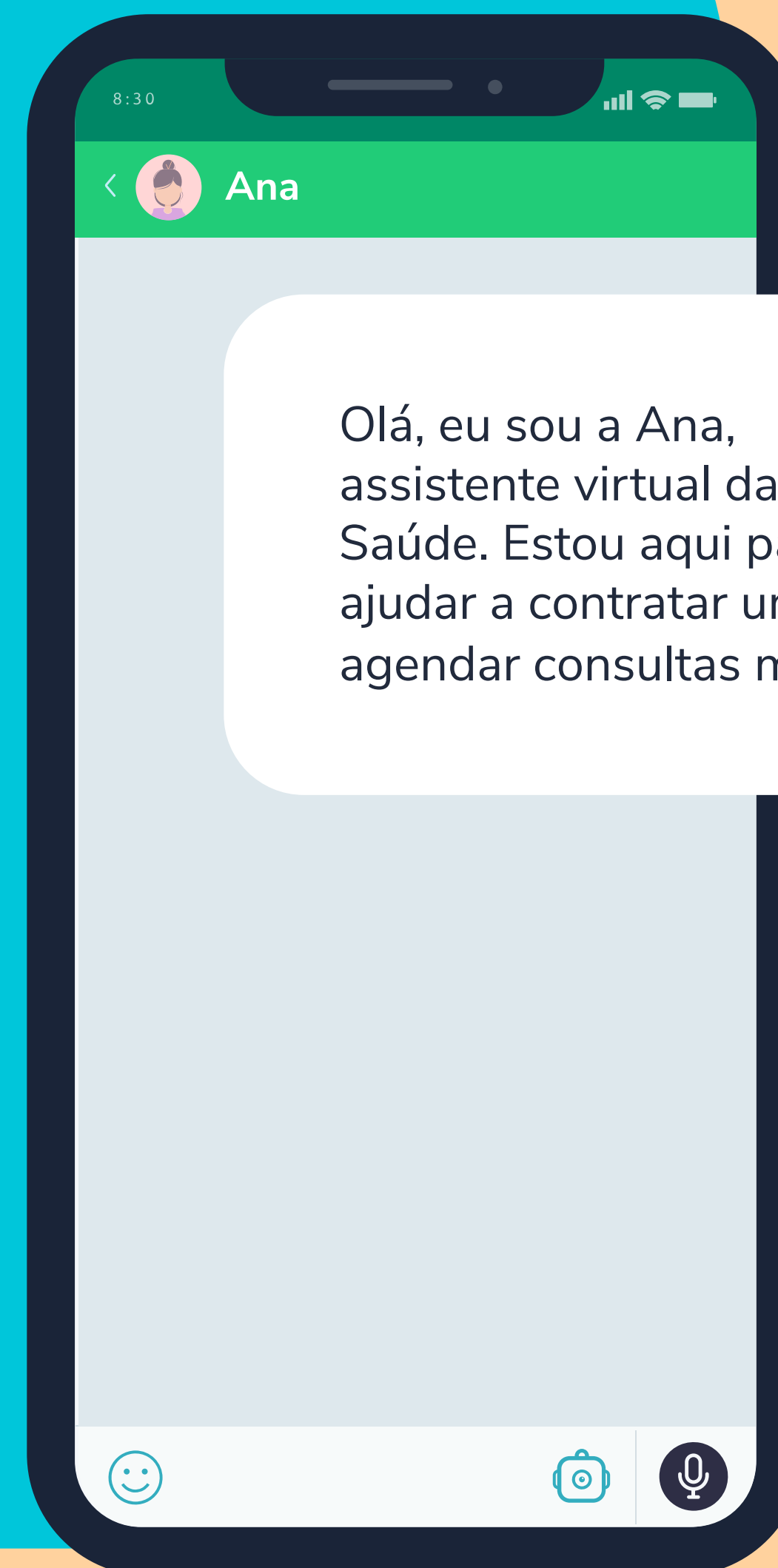
Saudar.

Avisar que é um Assistente Virtual.

Alinhar expectativa.

PASSO 04: perguntar o nome

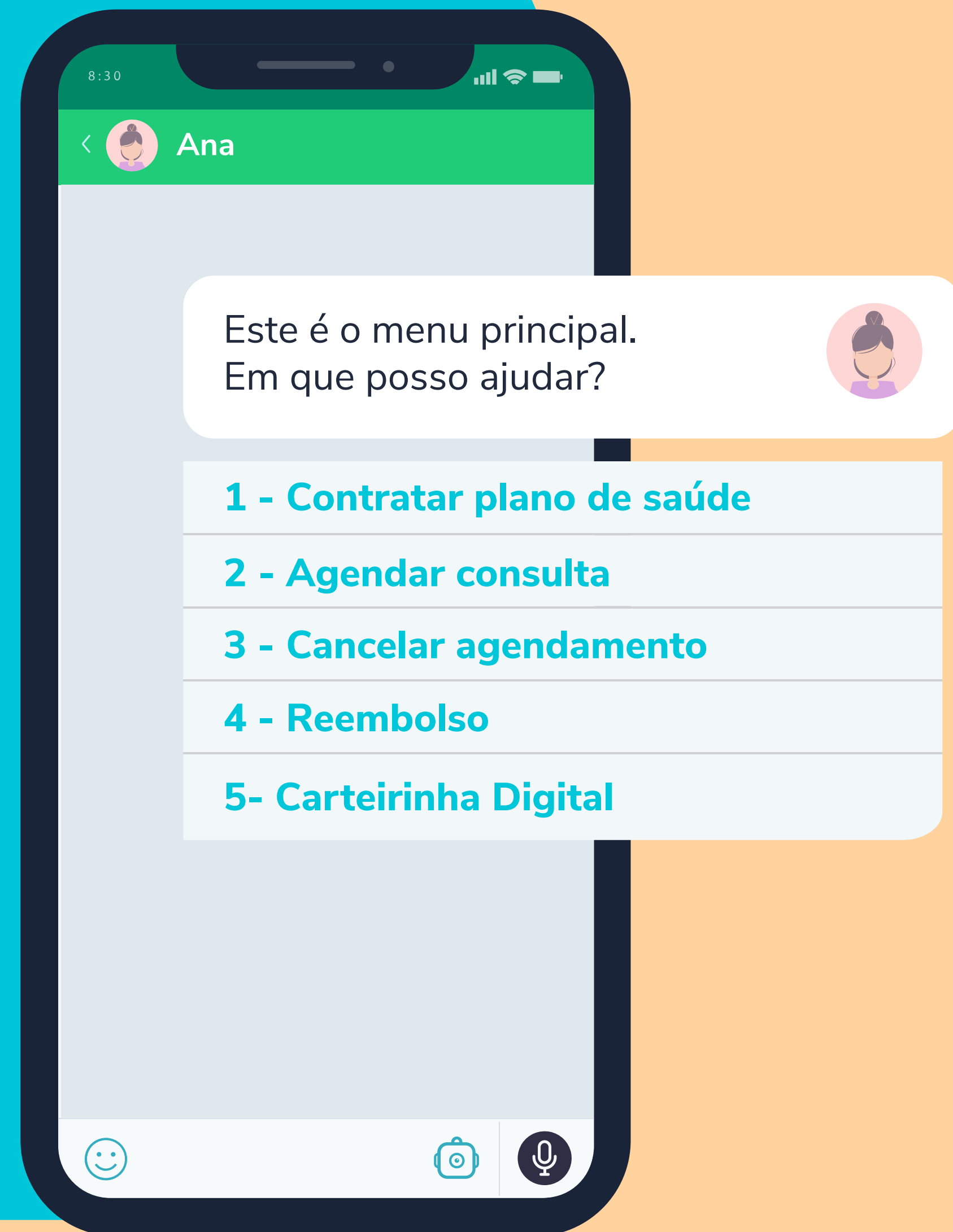
É uma boa prática perguntar o nome do usuário, salvá-lo em uma variável e também em Definir Contato. Desta forma, você consegue reaproveitar esta informação em seu fluxo.



PASSO 05: criar menu principal

Deixe claro o que o usuário conseguirá fazer dentro daquele canal. Ofereça no máximo 5 opções. Se necessário, você pode quebrá-las em submenus.

Use palavras-chave e números. Nada de frases longas aqui. Com base no menu principal, fica mais fácil montar o subfluxo do bot.



PASSO 06: identificar a intenção

Tenha em mente **diferentes formas que o usuário possa responder para uma mesma opção**. Por exemplo, considerando o menu acima, o usuário poderia dar as seguintes respostas quanto à opção Contratar plano de saúde:

1, contratar, contratar plano, plano de saúde, quero comprar um plano.

Outra boa prática é utilizar palavras de confirmação após a escolha do usuário: certo, ok, entendi.

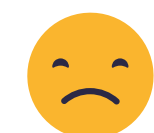


PASSO 07: prever inputs inesperados

Nem sempre o usuário irá dar a resposta que você espera, por isso, **antecipe inputs inesperados.**



- usuário responde conforme o esperado -> sequência caminho feliz



- usuário dá uma resposta errada -> sequência caminho triste



PASSO 08: alinhar expectativa ao longo do fluxo

Alguns processos podem levar um tempo, como quando é necessário consultar um dado na API ou é feito transbordo para um atendente. **Sinalize** isso para o cliente:

- **Aguarde um momento, por favor.**
- **Só 1min enquanto verifico.**
- **Aguarde um instante que já vamos te atender.**



PASSO 09: sinalizar transbordo

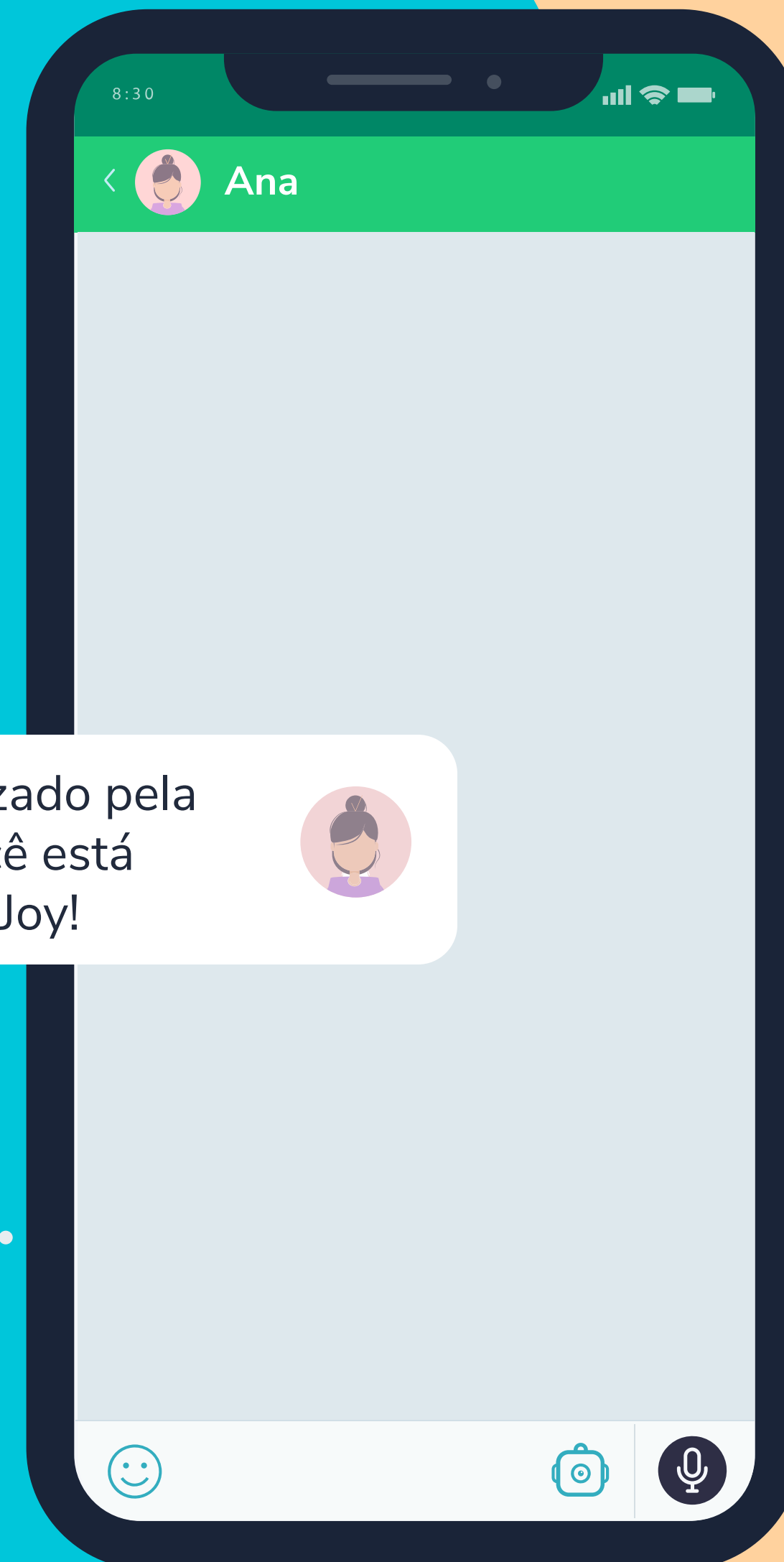
Quando é necessário transferência para atendimento humano, o bot fica “congelado” e o atendente assume. **Sinalize no fluxo as etapas onde ocorrem o transbordo.**



PASSO 10: mapear retorno

Após o encerramento do atendimento humano, retorne o usuário para o fluxo automatizado e informe que ele está interagindo com o bot novamente. **Planeje, ainda, como você quer direcionar o usuário a partir daí.**

O seu atendimento foi finalizado pela pessoa que lhe atendeu, você está novamente falando comigo, Joy!



PASSO 11: avaliar atendimento

Peça para o usuário **avaliar o atendimento realizado**.
Esta é uma excelente forma de mensurar a performance dos atendentes.

Conseguimos gerar um relatório dessas notas no Blip.

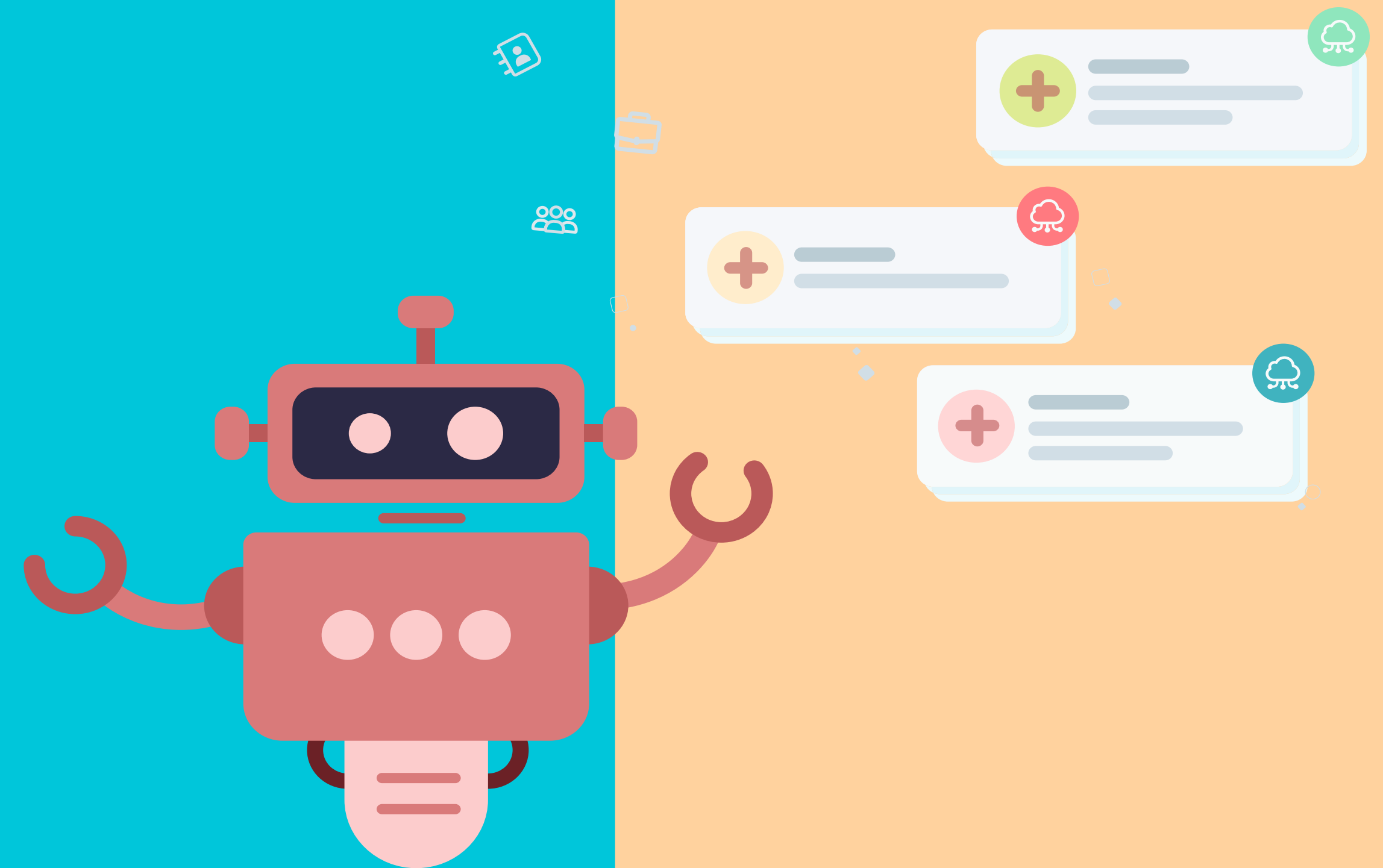


PASSO 12: implementar tracking

Termo utilizado para rastrear evento que acontecem no bot.
Sinalize no fluxo quais informações você gostaria de gerar relatório sobre, como:

- Caminho percorrido.
- Mensagens exibidas.
- Inputs abertos

Estes dados serão registrados no Blip por evento, mas também é possível analisar por usuários únicos. Falaremos sobre onde consultar as informações gerais logo mais.



Construindo um chatbot com **Take Blip**

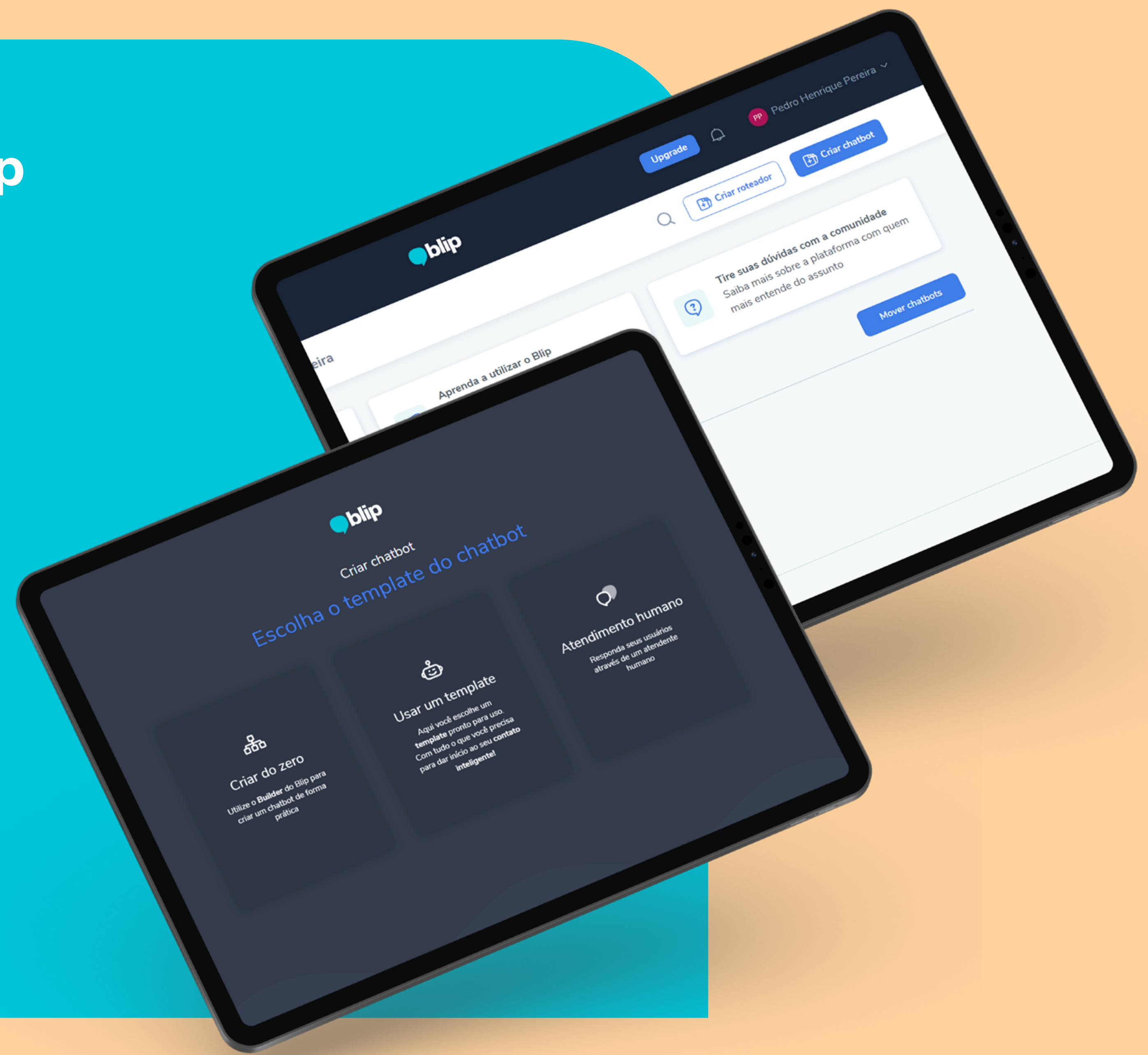
MÃOS NA MASSA



PASSO 01: acessar o Blip

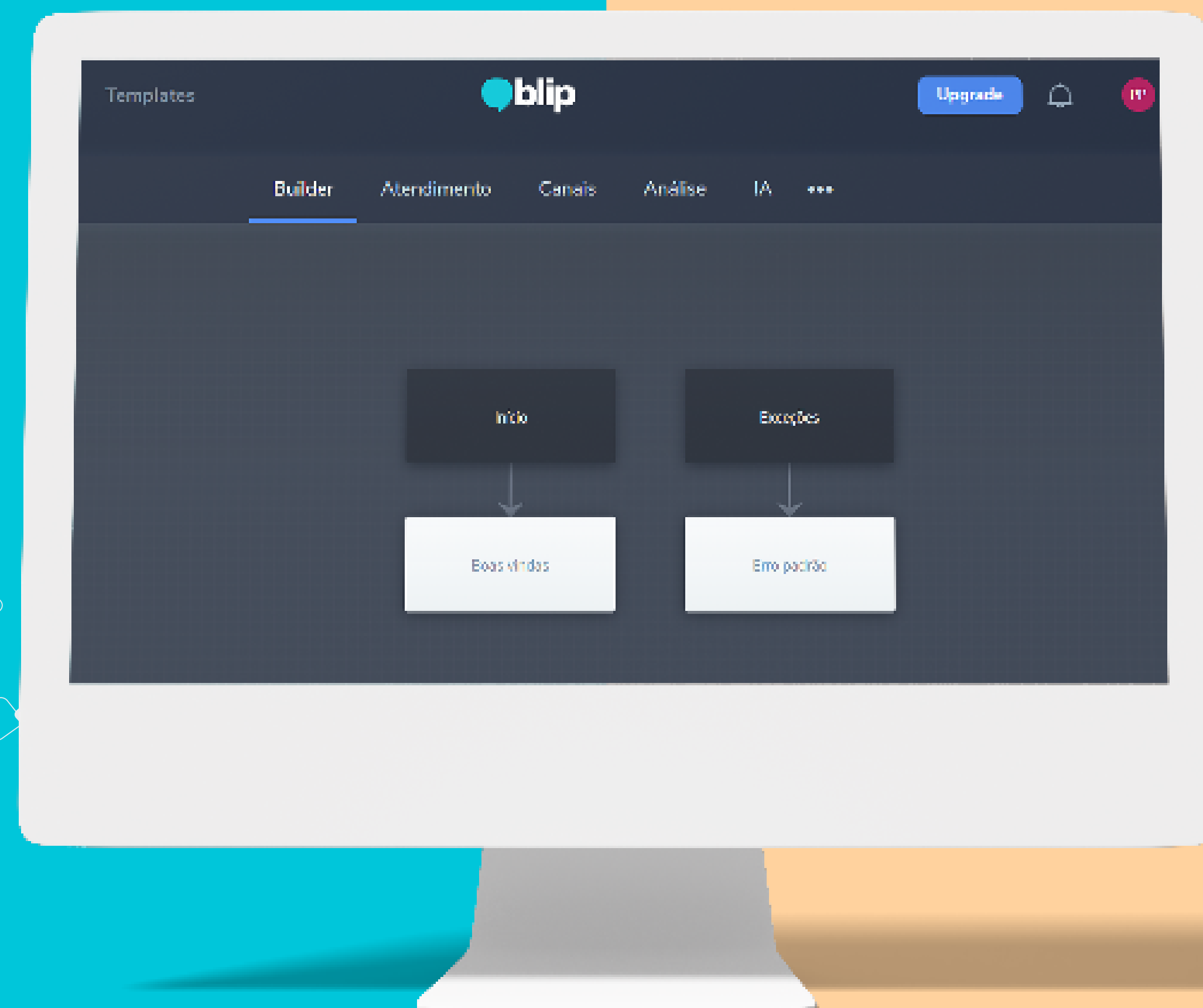
Após criar sua conta grátis no Blip, vamos dar continuidade ao projeto. Com seu fluxo pronto, o próximo passo será configurar o fluxo no Blip. Clique em **Criar chatbot**, dê um nome e escolha o tipo: criar do zero ou atendimento humano.

Se você seguiu este guia, escolha a opção criar do zero.



PASSO 02: entender os blocos de controle

Os **blocos de início** e **exceções** são padrões e não podem ser excluídos. Eles indicam o momento em que o usuário chegou ao fluxo (início) e para onde ele deve ser direcionado caso faça uma interação não mapeada (exceções).



PASSO 03: aprender sobre os blocos de conteúdo

Ao clicar sobre este bloco, como o de Boas Vindas, abrirá uma tela ao lado com algumas opções: conteúdo, condições de saída e ações:

[Acesse mais informações sobre os blocos de conteúdo aqui.](#)

Conteúdo:

sempre que o usuário alcança determinado estado (bloco), todo o conteúdo disponível ali é enviado para o cliente. Existem diversos tipos de conteúdo suportados pelo Blip, mas cada canal possui suas limitações.

Condições de saída:

através desta funcionalidade, é possível definir regras para a transição entre os blocos. Essas regras podem levar em consideração: o valor de variáveis, a entrada de dados feita pelos clientes, além de intenções e entidades analisadas no modelo de IA do bot. Para cada regra, deve ser definido um bloco alvo para o qual o usuário será enviado caso a condição seja satisfeita.

Ações:

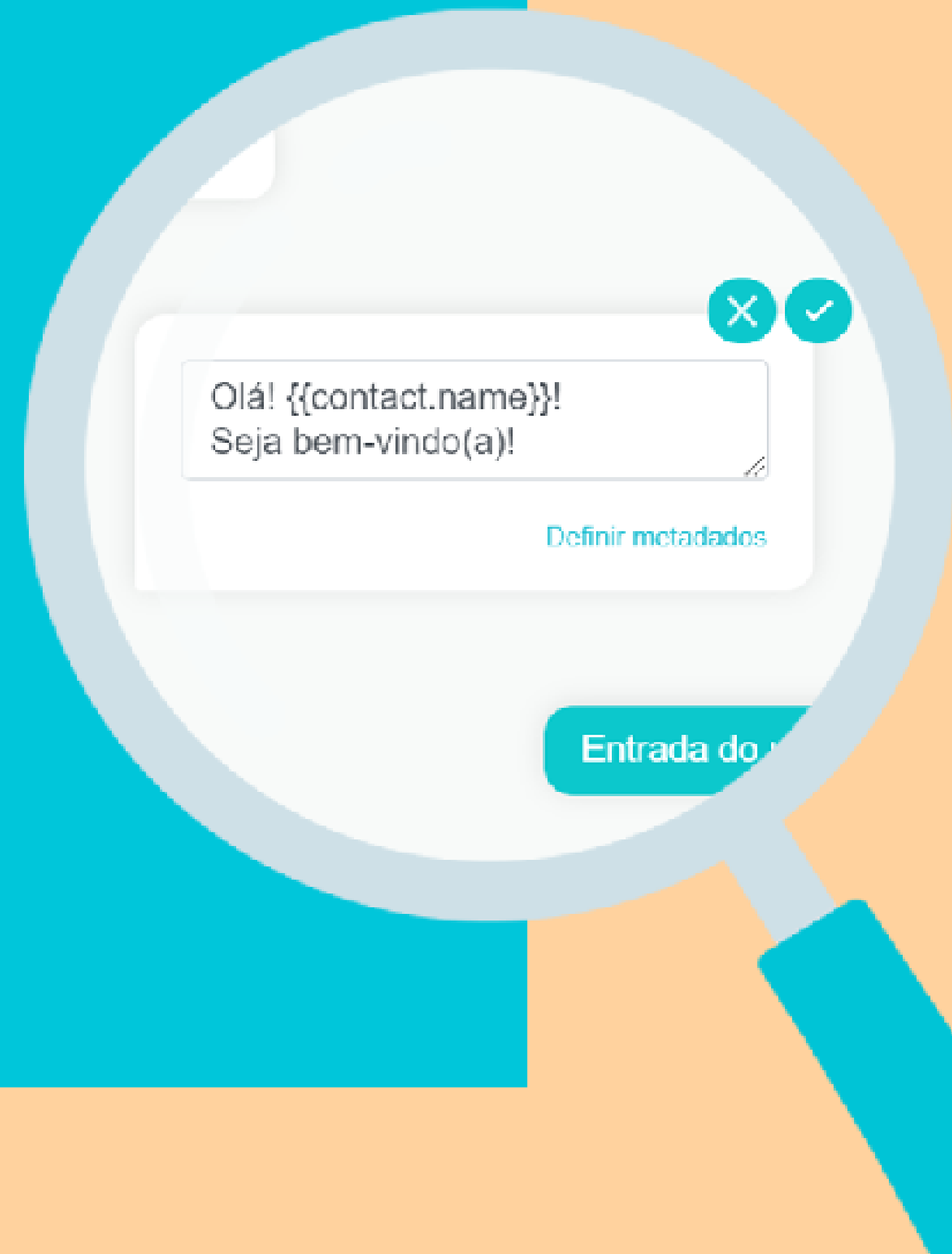
já a ação representa uma tarefa que deve ser executada antes do envio dos conteúdos de um bloco (ação de entrada), ou uma tarefa que deve ser executada após o envio dos conteúdos e]antes da transição entre o estado atual e o próximo (ação de saída).

PASSO 04: editar boas vindas

Agora que você já sabe os elementos que compõem um bloco, vamos editar a caixa de boas vindas. Por padrão, ela é criada com um texto e uma variável, mas você pode editar este campo conforme o conteúdo que você planejou para o seu fluxo.

Dica: utilize emojis e negritos, mas atente-se ao que é permitido no canal onde você irá publicar o seu bot.

Após escrever ou alterar um texto, dê o check para salvar. Confira [este material](#) sobre variáveis no Blip.



PASSO 05: indicar entrada do usuário

A entrada do usuário indica que, somente após o usuário realizar uma interação, ele seguirá para o próximo bloco. Você pode manter ou excluir este conteúdo conforme mapeado no seu fluxo.

Também é possível validar a entrada do usuário. Ou seja, para que ele continue o fluxo, deverá ter fornecido uma resposta válida, no formato que você precisa.

Obs.: um bloco pode conter 0 ou 1 entrada do usuário. Caso o bloco não possua uma entrada prevista, o usuário seguirá para a etapa seguinte do fluxo.

PASSO 06: adicionar novos blocos



Para adicionar um novo bloco, basta clicar neste **ícone +** que estará logo abaixo do nome seu bot. Caso você espere que o seu cliente responda ou escolha uma opção, mantenha a **“entrada do usuário”**.

Dica: assim como você fez em seu fluxograma, renomeie os blocos no Blip, assim ficará mais fácil visualizar seu fluxo. Para isso, é só clicar sobre **“Novo bloco”** e mudar o texto.

PASSO 07: configurar menu

Você deve **adicionar uma condição de saída para cada opção existente em seu menu**. A dica aqui é escolher o tipo **“Contém”** e colocar variações que o usuário pode responder para uma mesma opção. Pense nas palavras e números que podem ser utilizados e considere ainda erros de português.

Menu principal

Conteúdo Condições de saída Ações

CONDIÇÕES DE SAÍDA ⓘ 4/25

Se Resposta do usuári

Condição Contém

1 x um x un x

uma x status x solicitação x

atendimento x Valores

+

Ir para Status Paciente

Dicas:

- O Blip faz diferenciação de acentuação, mas não de maiúscula e minúscula.
- As condições de saída são avaliadas em ordem. Ou seja, a primeira opção que satisfizer a resposta do usuário irá direcionar a interação. Coloque as condições mais restritivas primeiro.
- Para direcionar o usuário a um bloco que não foi criado ainda, é só digitar o título/nome de referência e apertar **“enter”**. O bloco será criado e interligado automaticamente.

PASSO 08: adicionar atendente humano



Para adicionar um bloco de atendimento humano, basta passar o mouse sobre o ícone “**novo bloco**” e escolher a opção com o “**bonequinho**”.

Através das “**condições de saída**”, é possível configurar a mudança de estado do cliente após a finalização do atendimento para que o usuário seja direcionado para uma nova etapa do fluxo.

Recomendamos pelo menos **3 condições**:

- **Se ticket finalizado pelo atendente**
- **Se ticket finalizado pelo cliente**
- **Se ticket finalizado por inatividade do cliente**

CONDIÇÕES DE SAÍDA ⓘ

Se ticket finalizado pelo atendente

Ir para
Finalização

Se ticket finalizado pelo cliente

Ir para
Finalização

Falaremos mais sobre blocos de finalização, checagem e retorno do cliente adiante.

Você pode criar regras de atendimento, adicionar equipes e atendentes e realizar a verificação de horário de atendimento no próprio Blip. Nos materiais de apoio, você encontra mais informações.

Caso prefira, você pode fazer um fluxo específico para o atendimento humano e direcionar o usuário para outro bot. Isso é possível desde que você utilize hierarquia de bots e subbots.

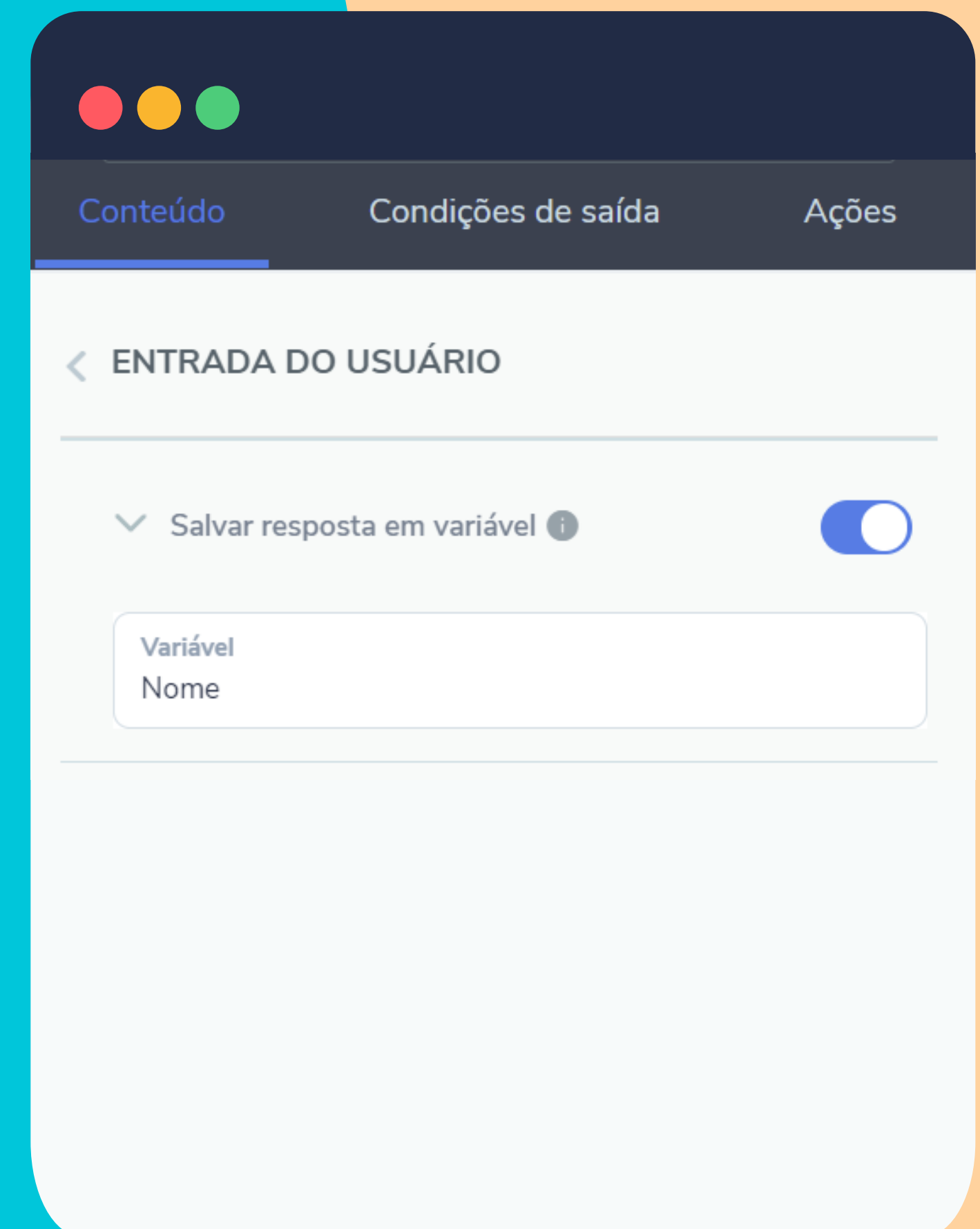


PASSO 09: salvar resposta do usuário

Salvar a resposta do usuário é ótimo para **confirmar uma informação com o próprio cliente e adicionar informação no contato**, por exemplo. Para isso, basta clicar sobre a entrada do usuário. Você deverá habilitar a opção **“Salvar resposta em variável”** e definir um nome para referência.

Para utilizar a variável criada, coloca-se o nome entre chaves: **{{exemplo}}**. Você deve seguir o exato formato para utilizar: maiúsculo / minúsculo.

Em ações você determina como essa informação será salva. Por exemplo, para salvar o nome do usuário, você pode escolher o **“Tipo de validação - texto”**, incluir uma instrução e, em ações, selecionar **“Ações de saída” > “Definir Contato”**. Dentro do campo **“Nome”**, salvamos a variável entre chaves, por exemplo, **{{Nome}}**.



PASSO 10: validar resposta do usuário

Caso você solicite algum dado do usuário, como número do telefone, data de nascimento ou CPF, é possível fazer uma verificação prévia através do próprio Blip. Fazemos isso através da opção **“Validar entrada do usuário”**.

Na mesma tela de **“Entrada o usuário”**, há 4 tipos de validação. São quatro opções de tipo de validação: Texto, Número, Data e Expressão Regular.

Texto:

verifica se a resposta do usuário é um texto (inclui números, pontuações e caracteres especiais). É indicado quando você quer barrar o envio de áudio, imagens e outros tipos de mídia.

Número:

verifica se a resposta apresentada é numeral. Inclui números inteiros, positivos, negativos e decimais (separados por ponto ou vírgula).

Data:

apura se a resposta apresentada é uma data. Contempla diferentes formados, como: dd/mm/aaaa, mm/dd/aaaa, dd-mm-aaaa, mm-dd-aaaa, dd-mm, dd/mm, mm-dd, mm-dd.

Expressão regular:

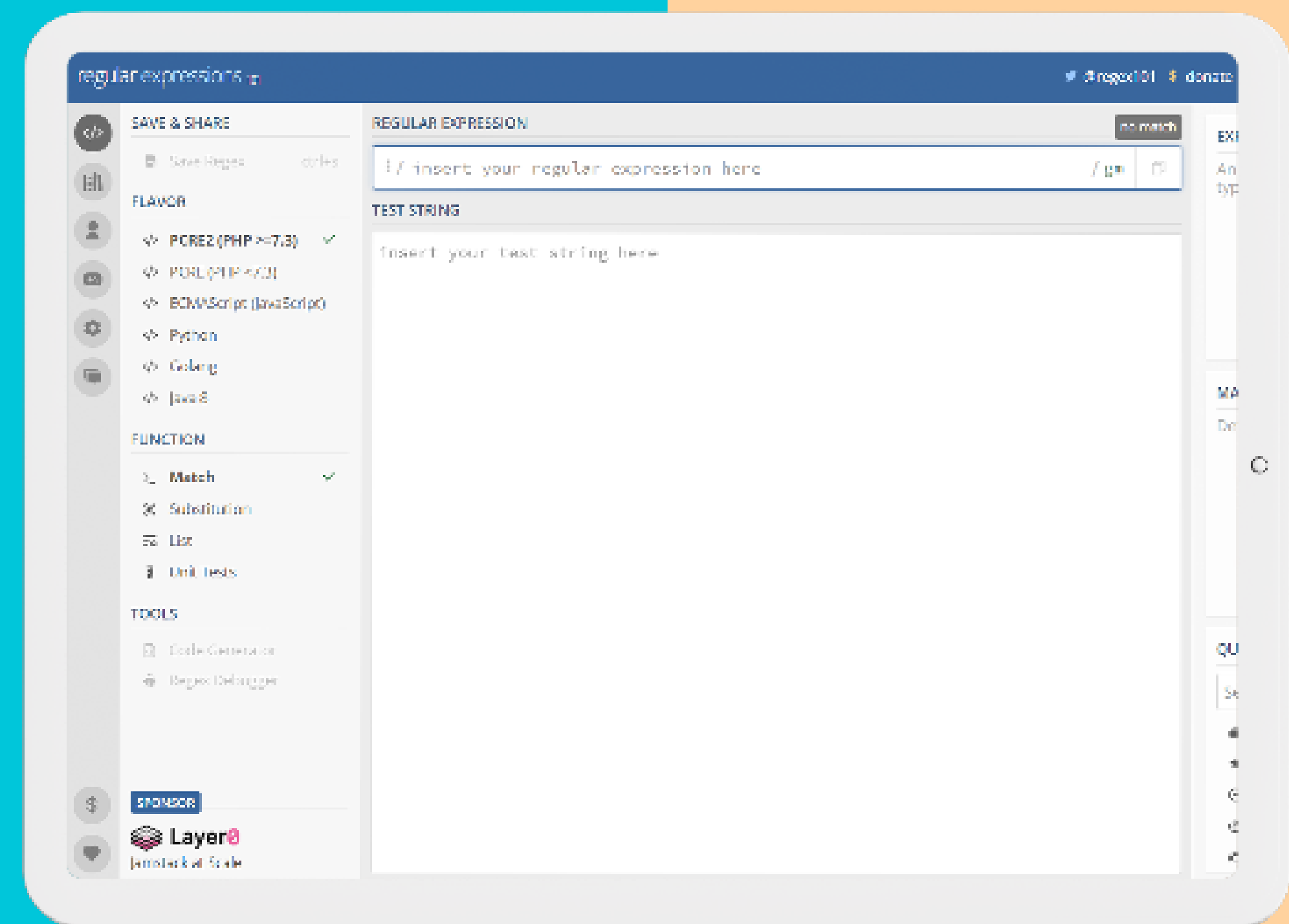
também chamada de regex, é a opção com mais possibilidades. Aqui, você informará um padrão e o Blip verificará se a resposta do usuário corresponde ao padrão cadastrado. É uma ótima forma de validar CEP, CPF, email, telefone e qualquer outro dado que tenha um padrão a ser seguido.

O site [regex101](https://regex101.com) funciona como um acervo digital de expressões regulares. Lá você pode pesquisar, validar e copiar expressões para utilizar no Blip.

Exemplos de expressões regulares:

- CPF (sem pontos): `^\d{11}$`
- Email: `^([\w.-]+)@([\w-]+)((.(\w){2,3})+)$`
- CEP: `^\d{5}-\d{3}$`
- Telefone: `^[1-9]{2}-[2-9][0-9]{7,8}$`

Você também pode fazer esse tipo de validação nas condições de saída. É só colocar “**Se resposta do usuário corresponde à regex**” e informar o valor.



PASSO 11: testar

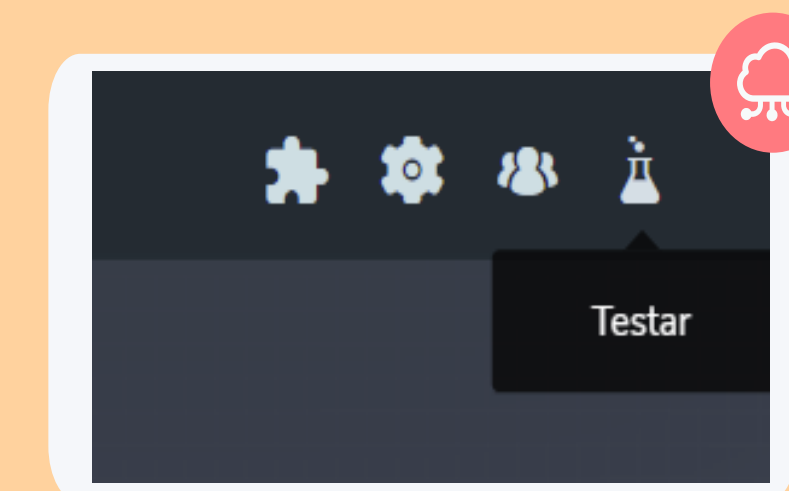
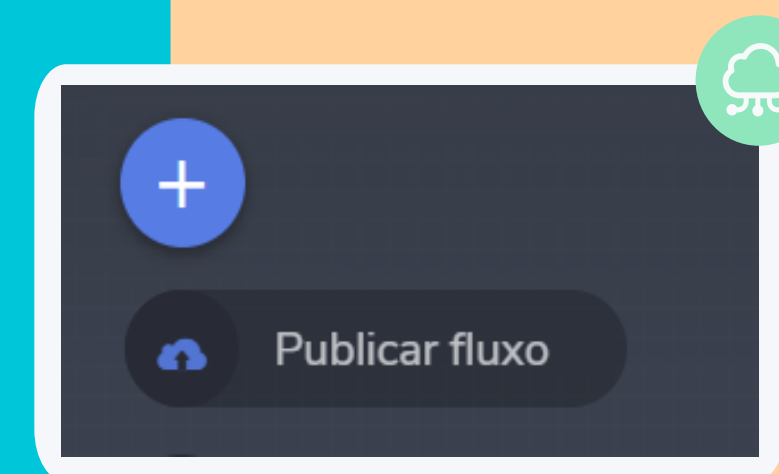
Sempre teste seu bot antes de publicar no canal/divulgar para cliente e também após qualquer mudança no fluxo. Você pode testar de duas formas: pelo chat de teste no builder ou através do Debug.

É necessário que sua alteração tenha sido publicada no builder para o teste funcionar corretamente. Para isso, clique na nuvem abaixo da opção de adicionar bloco.

Para acessar a opção de teste padrão, clique no tubo de ensaio no canto superior direito. Basta interagir com o chat para iniciar o teste.

Com a ferramenta **Debug**, é possível **analisar todo o comportamento do fluxo do seu bot e acompanhar como acontece a execução em cada um dos blocos que compõem a conversa da sua aplicação.**

Sempre que for chamar a API, indicamos usar o Debug para enter quais informações estão sendo recebidas e como estão sendo tratadas no Blip.



PASSO 12: consultar API

< REQUISIÇÃO HTTP

Essa ação permite realizar uma requisição HTTP para qualquer API pública

Método

URL

Caso você queira consultar numa API uma informação dada pelo usuário, será necessário realizar uma **“Requisição HTTP”**.

O Blip aceita todos os métodos de requisição. Ao final da URL da API, inclua a variável a ser consultada entre chaves **{}**.

✓ CABEÇALHOS

Key	Value
-----	-------

+ Adicionar cabeçalho

X

Caso não seja utilizada uma API aberta, pode ser necessário realizar autenticação. Isso é feito na opção **“Cabeçalhos”**.

O próximo passo será definir as variáveis de retorno. Você pode utilizar esses retornos para exibir as informações para o usuário, ou para armazenar alguma delas no contato, por exemplo.

SALVAR RESPOSTA ⓘ

Variável para status da resposta

Variável para o corpo da resposta

Em **“Variável para status da resposta”**, você determina a variável que será pesquisada na API. Já no campo **“Variável para corpo da resposta”**, você inclui a variável que contém a informação a ser retornada para o usuário, caso se aplique.

Você pode utilizar estas informações para determinar o caminho que o usuário irá seguir no fluxo.

Confira um exemplo de como utilizar as informações que foram consultadas na API nas condições de saída e direcionar o usuário pelo fluxo:

se variável igual a Nome do Campo Variável para status da resposta valores 404 > Ir para Bloco para onde o usuário será direcionado.

404 é o valor padrão de erro quando a API não encontra uma informação.

se resposta do usuário existe > Ir para Bloco para onde o usuário será direcionado.

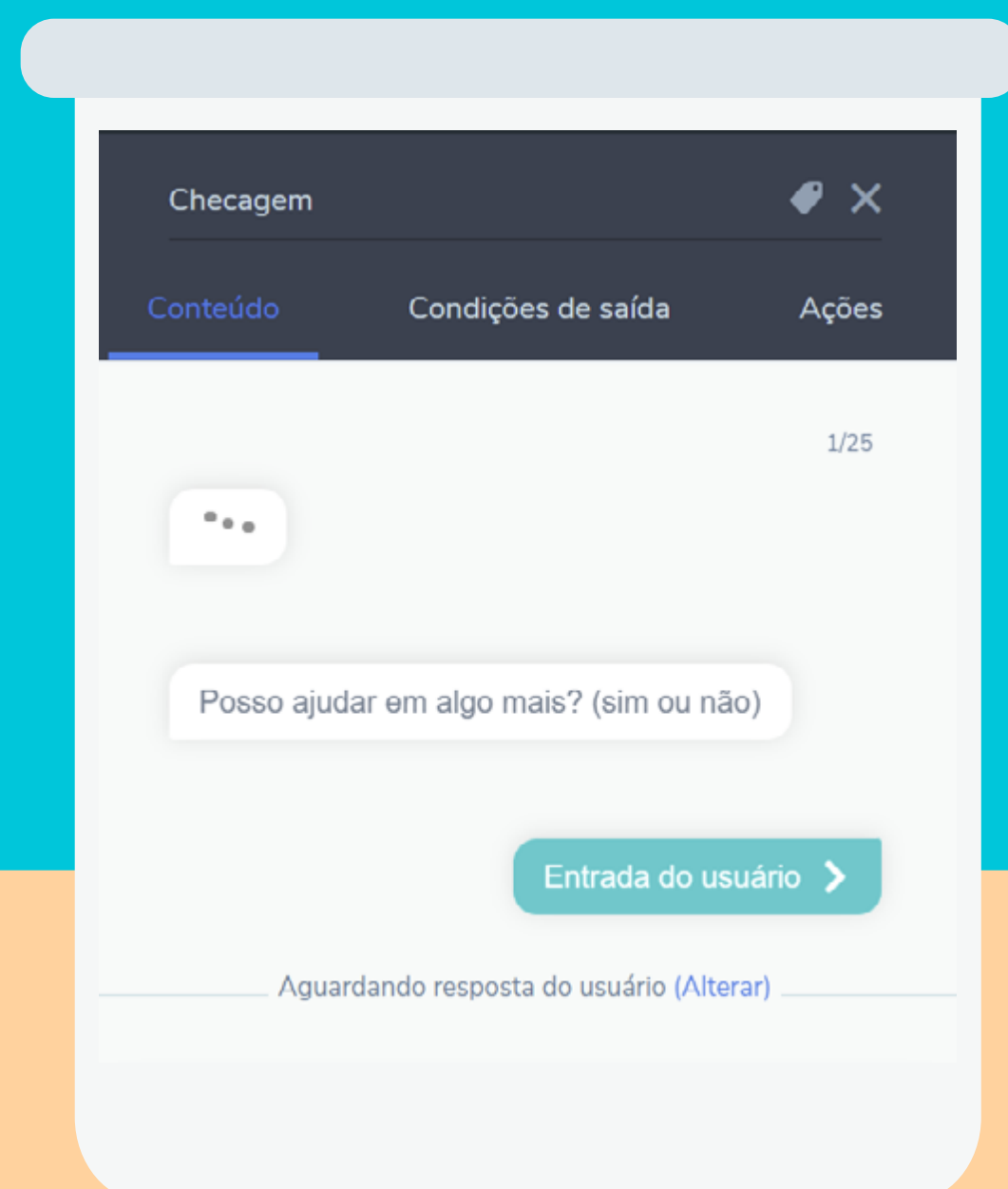
Para retornar a informação para o usuário, utilizamos o parâmetro **VariávelParaCorpodeResposta@VariáveldaAPIinterna**. Ou seja, retorno da API + campo dentro daquele retorno. Isso será feito no bloco seguinte, que corresponde ao sucesso da informação.

[Consulte a documentação do Blip para mais informações.](#) É recomendado, ainda, que você conte com o apoio de um desenvolvedor nesta etapa.



PASSO 13: incluir checagem

Outra boa prática é **incluir um fluxo de checagem para onde o usuário será encaminhado ao final de cada etapa**. Assim, você valida se o seu cliente necessita de algo mais ou se deseja encerrar o atendimento. Conforme resposta, você pode enviar o usuário de volta ao menu, por exemplo, ou para o bloco de encerramento.



PASSO 14: pesquisa de satisfação

Existem algumas formas de realizar a pesquisa de satisfação através do próprio fluxo do chatbot. Em todos, será necessário **criar um bloco de pesquisa**, após o bloco de atendimento humano, caso você queira avaliar seu atendente, ou na etapa do fluxo de preferência.

Você pode seguir a metodologia que preferir: NPS, CSAT ou qualquer outra que sua empresa adote. Para este exemplo, vamos pedir ao usuário que avalie o atendimento que recebeu numa escala de 0 a 5.



1 - Após o bloco de atendimento humano, **criamos um** bloco de finalização, onde informamos que o usuário está conversando com a assistente virtual novamente. **Esta é uma excelente prática! Dessa forma, fica claro para o usuário que ele não está mais conversando com o atendente humano.**



2 - Nas condições de saída deste bloco, colocamos: se resposta do usuário condição existe ir para Pesquisa de Satisfação.

3 - Nós também excluimos a entrada do usuário no bloco de finalização. Assim, ele seguirá para o próximo bloco independente de uma interação.

4 - No bloco de Pesquisa de Satisfação, vamos pedir para o usuário avaliar numa escala de 0 a 5 e salvar essa entrada do usuário.



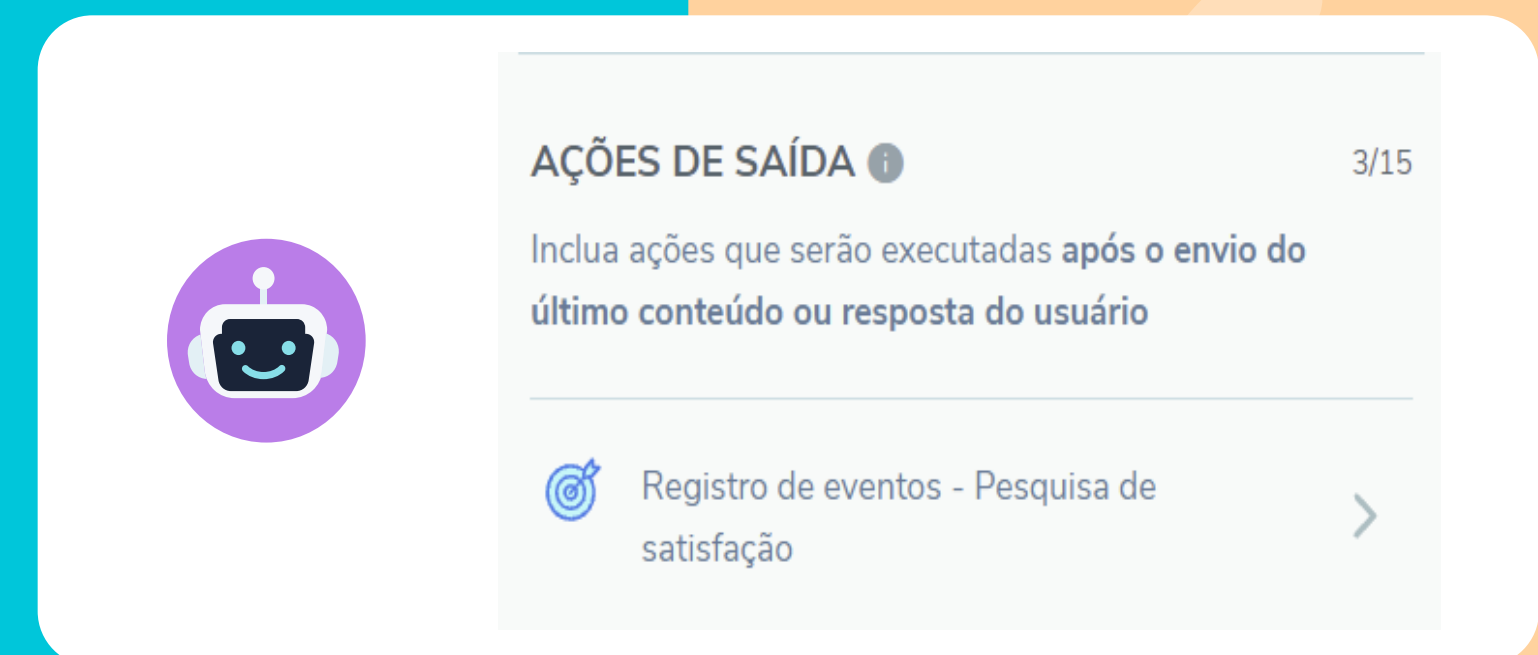
5 - Vamos clicar na entrada do usuário e, no bloco de conteúdo, fazemos as seguintes configurações:

a - salvar resposta em variável e atribuir um nome,

b - validar entrada do usuário tipo número, pois queremos a avaliação numa escala de 0 a 5, e incluir uma instrução,

c- definir tempo de inatividade de para que a sessão seja expirada caso o usuário pare de interagir com o bot.

6 - Nas **Ações > Ações de Saída, criamos um registro de evento:**



7 - No registro de evento, vamos dar um nome para a categoria para facilitar a identificação do que está sendo registrado e, na ação, vamos incluir nossa variável entre chaves: `{{score}}`.

8 - Nas “**Condições para registro de eventos**”, colocamos: **se resposta do usuário condição igual a** valores 0 1 2 3 4 5. Ou seja, esse evento somente será registrado e incluído num relatório caso o usuário dê alguma dessas respostas. Isso evita que seja registrada uma informação que não nos interessa aqui.

9 - Já nas “**Condições de Saída**”, vamos direcionar o usuário para dois caminhos:
se resposta do usuário condição igual a valores 3, 2, 1, 0 ir para Justificar nota.
se resposta do usuário condição igual a valores 4, 5 ir para Checagem.

Você pode direcionar o usuário para um etapa de justificativa, se julgar interessante, onde o usuário poderá dar mais clareza da sua nota. Isso é interessante, principalmente, para avaliações abaixo do esperado. Após a resposta do usuário, você o direciona para o agradecimento

Nas “Condições de Saída” do bloco de agradecimento, atribuímos: se resposta do usuário condição existe ir para Checagem. Assim como fizemos para clientes com nota superior a 3.

Conteúdo Condições de saída Ações

< REGISTRO DE EVENTOS - ✎

Os eventos são agregados por categoria, ação e dia. Saiba como criar relatórios e gráficos personalizados

Categoria Pesquisa Satisfação ✓

Ação {{score}} ✓

Rótulo (opcional)

Valor (opcional)

+ Adicionar informações extras

CONDIÇÃO PARA REGISTRO DE EVENTOS

Adicionar condições para o registro desse evento

Se Resposta do usuário

Condição Igual a

5 x 4 x 3 x

2 x 1 x 0 x

Valores

+

PASSO 15: encerramento do fluxo

Como falamos, **sempre direcione o usuário para um bloco de Checagem antes de encerrar o fluxo**. Já no bloco de encerramento, uma boa prática é variar as fraseologias de despedida.

Para revezar as fraseologia no Blip, usa-se um sintaxe parecido com o utilizado com as variáveis.

Todas as opções de frases devem ficar entre chaves **{{ }}**. Para diferenciá-las entre si, usa-se o type |

Neste bloco, nós também mantemos a entrada do usuário. Assim, caso ele interaja com o bot novamente, iremos encaminhar para o bloco **“Retorno mesmo cliente”**.



PASSO 16: retorno mesmo cliente

Se você tiver salvo o nome do seu cliente dentro do contato, podemos usar a variável **{{contacta.name}}** no bloco de retorno, dessa forma:

Essa prática é interessante, pois deixa seu bot mais **“simpático”** e **“amigável”**, não sendo necessário perguntar o seu nome novamente.

A partir dali, o usuário seguirá para fluxo normal, não sendo necessária uma entrada do usuário. Basta modificar as condições de saída.

Lembre-se de salvar a variável do nome na entrada do usuário e criar uma ação de saída **“Definir contato”**.



PASSO 17: criar registro de eventos

Registro de evento é uma ação que permite que eventos importantes ocorridos durante o fluxo conversacional sejam registrados para análises futuras. **Dessa forma, conseguimos mapear, por exemplo:**

- Número de mensagens contendo uma palavra específica;
- Número de imagens recebidas pelo contato;
- Quantidade de clientes em cada canal;
- Quantos clientes recusaram atendimento humano.

Nós já vimos como criar um registro de evento para salvar a nota do usuário. Se você já tiver mapeado no desenho do seu fluxo os dados deseja registrar, não se esqueça de implementar no seu chatbot. **Todo registro de evento é realizado nas ações do bloco, seja entrada ou saída.**

Além disso, o Registro de Eventos possui três componentes: categoria, ação e parâmetros extras.

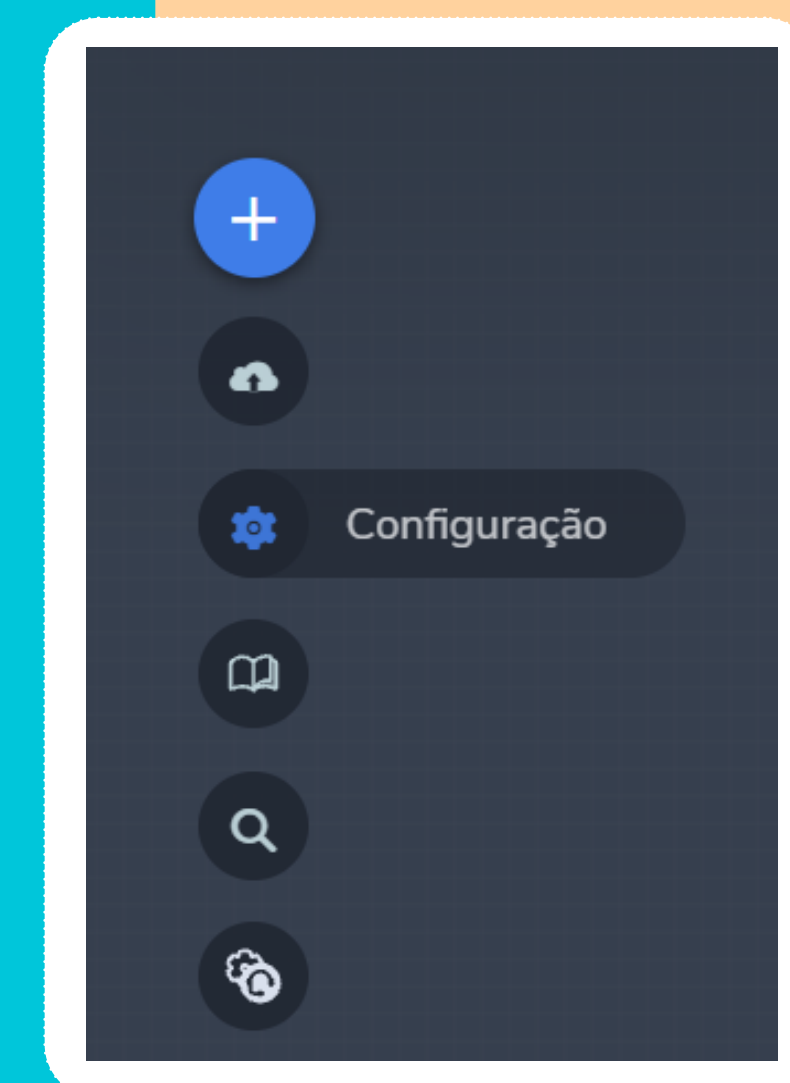
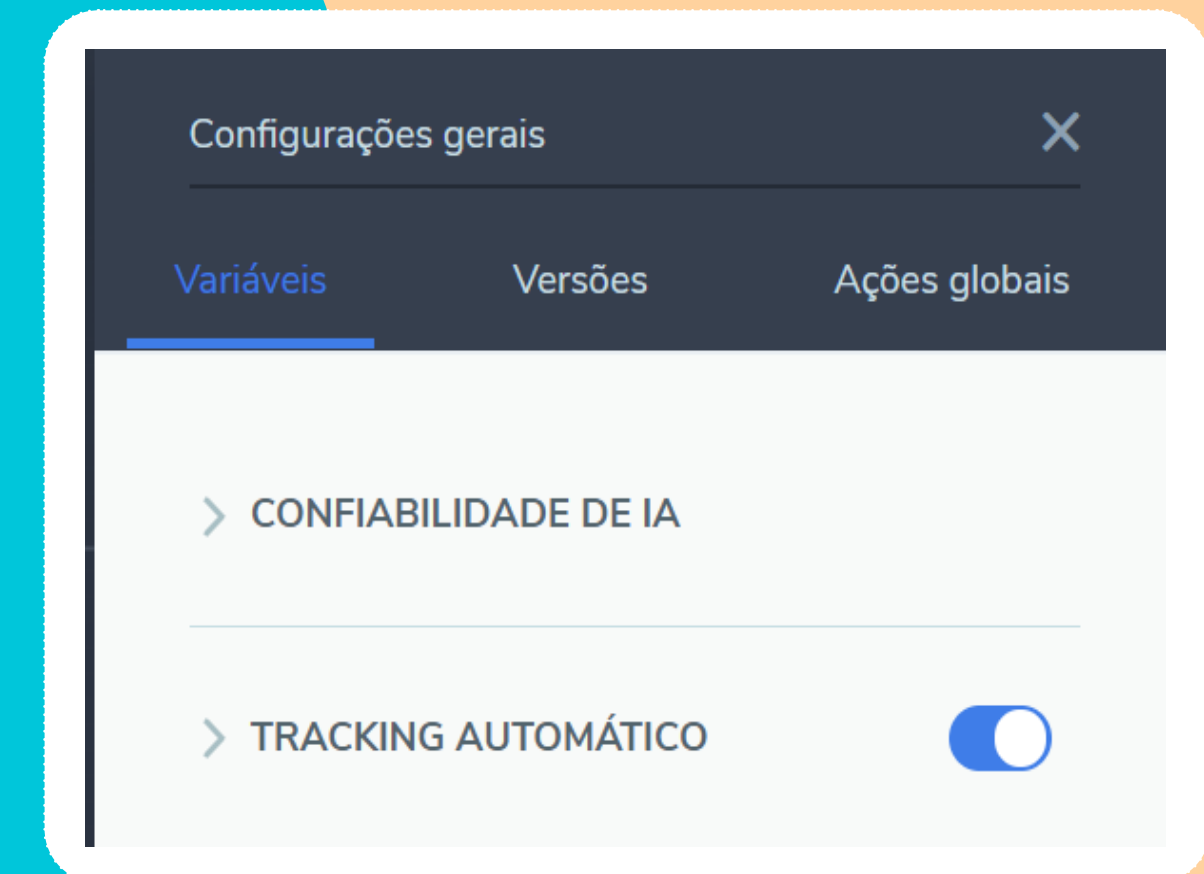
Os dados gerados poderão ser acessados através dos relatórios personalizados. Falaremos sobre isso logo abaixo. **[Neste treinamento](#)**, explicamos com detalhes tudo que você precisa saber para criar registros de eventos estratégicos.

PASSO 18: ativar tracking automático

O tracking automático registra um evento para cada bloco do fluxo do chatbot, sempre que um usuário passa por ele. O evento é registrado na categoria **“flow”** e a ação é o nome de cada bloco. Essa funcionalidade pode ser usada para criar relatórios personalizados na tela de análise, permitindo, por exemplo, a avaliação dos fluxos e blocos mais utilizados. Nesse caso, não é necessário criar os event-tracks personalizados de bloco por bloco.

Para ativar essa funcionalidade, basta acessar o menu de configurações e habilitar a opção tracking automático.

Os dados gerados pelo tracking automático são disponibilizados no relatório de Jornada dos Contados, no menu de Análise.



PASSO 19: expirar sessão

Ainda no menu de configurações, é possível habilitar o recurso **expirar sessão**, que é calculado **em segundos**. Neste campo, você define **a partir de quanto tempo sem interagir com o bot o usuário voltará para o início do fluxo**. Se esse valor não estiver definido, a expiração da sessão não ocorre.

EXPIRAÇÃO DA SESSÃO

Tempo em segundos de expiração da sessão dos usuários em caso de inatividade. Em caso de expiração da sessão, o usuário volta para o estado inicial do fluxo. Se este valor não estiver definido, a expiração não ocorre.

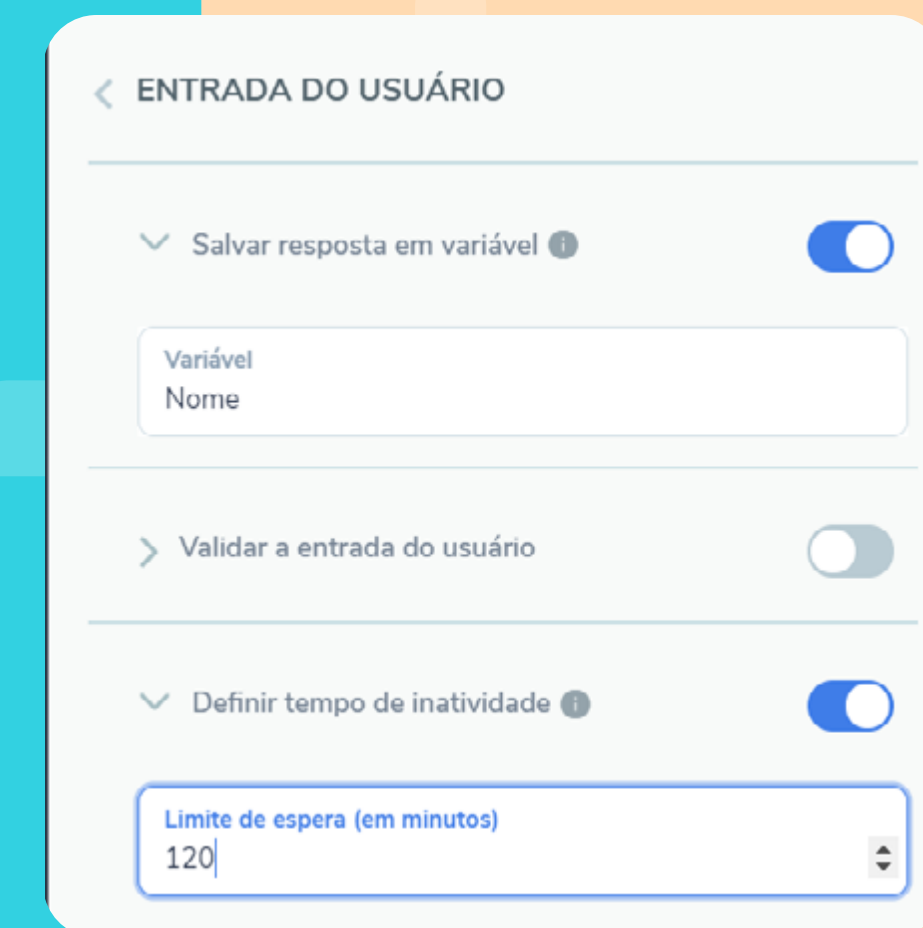
Expiração da sessão
3600

PASSO 20: usar o tempo de inatividade do usuário

É possível definir o **tempo máximo, em minutos, que o bot deverá esperar pela resposta do usuário** antes de encaminhá-lo para a próxima etapa do fluxo. Essa configuração está disponível em todos os blocos do seu fluxo, com exceção dos blocos de início e de atendimento humano!

Para ativar o tempo de inatividade, você deve **acessar o comando “Entrada do usuário”, ativar a opção “Definir tempo de inatividade” e inserir o tempo desejado**. O ideal é que o tempo de inatividade seja inferior ao tempo de expirar sessão.

Fazendo isso, você pode atribuir uma condição de saída que irá verificar que, se não houver resposta, o usuário deverá seguir por determinado caminho e enviar uma mensagem automática para reengajá-lo, por exemplo



< ENTRADA DO USUÁRIO

✓ Salvar resposta em variável

Variável
Nome

> Validar a entrada do usuário

✓ Definir tempo de inatividade

Limite de espera (em minutos)
120

Construindo um chatbot com **Take Blip**

GERENCIAMENTO E
ANÁLISE DE RESULTADOS



PASSO 01: gerenciar os horário de atendimento

Existem algumas formas de realizar a **verificação de horário de atendimento humano**: você pode utilizar o template de bot manual, criar blocos no seu fluxo e setar a configuração ali, executar script, dentre outras formas.

Para facilitar controle e gerenciamento de horários de atendimento do seu chatbot, disponibilizamos um template de atendimento escalável que você pode adaptar para o seu negócio. [Neste link](#), você acessa as informações sobre como baixar e configurar o template no seu portal.

Criamos vídeos tutoriais que explicam detalhadamente sobre este modelo:

[Configuração de atendimento simples](#)

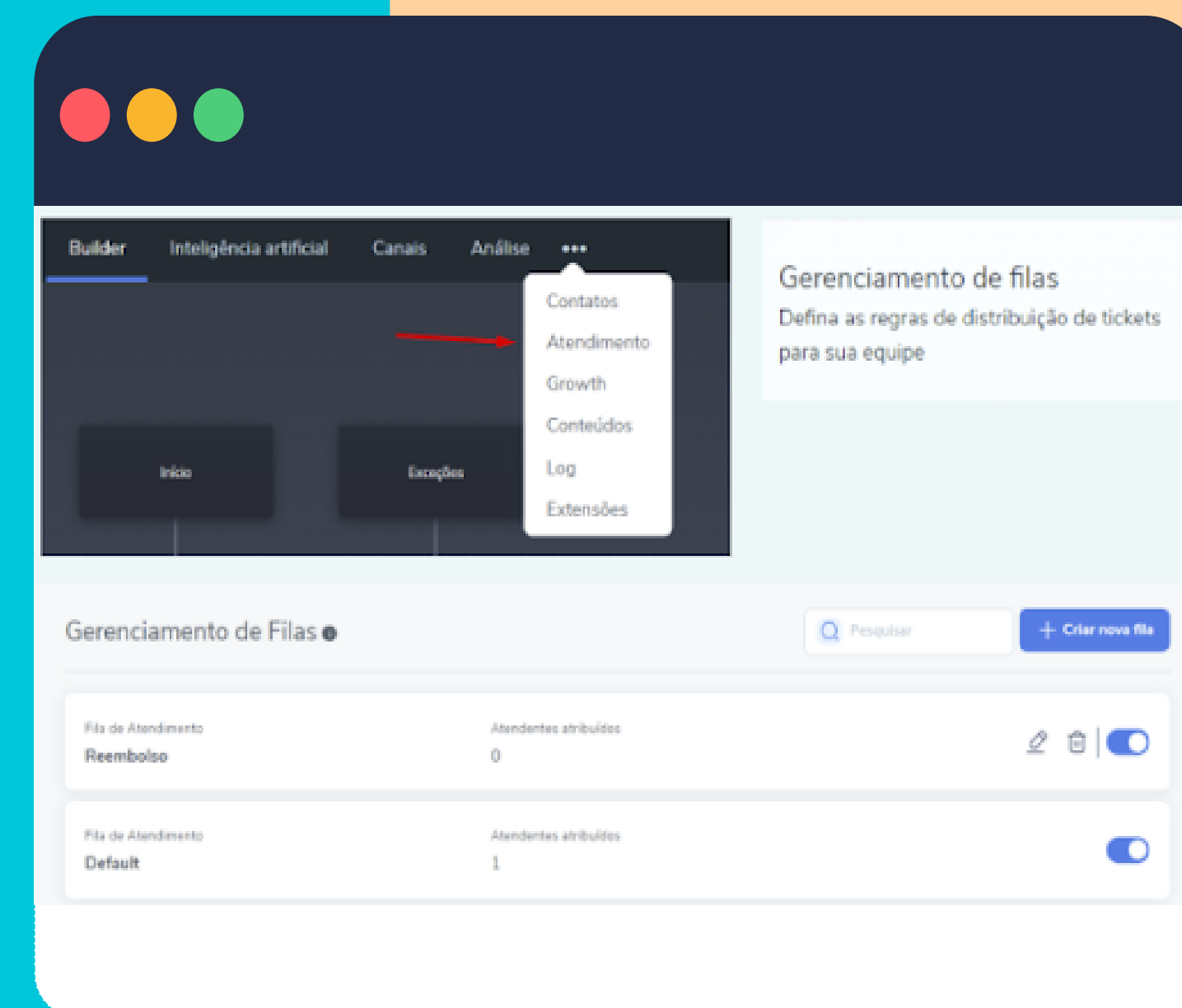
[Configuração de atendimento por fila e dia da semana](#)

Se preferir executar diretamente no seu fluxo, você pode replicar as configurações ensinadas [neste artigo](#).

PASSO 02: gerenciar filas

O Gerenciamento de filas, disponível no módulo de Atendimento, é um recurso de gestão de atendimento onde devem ser criadas as filas (antes identificadas como “equipes”) e definidas as Regras de atendimento para direcionamento dos tickets.

Aqui, também é possível acompanhar os atendentes atribuídos a cada uma das filas ao clicar no ícone de edição.

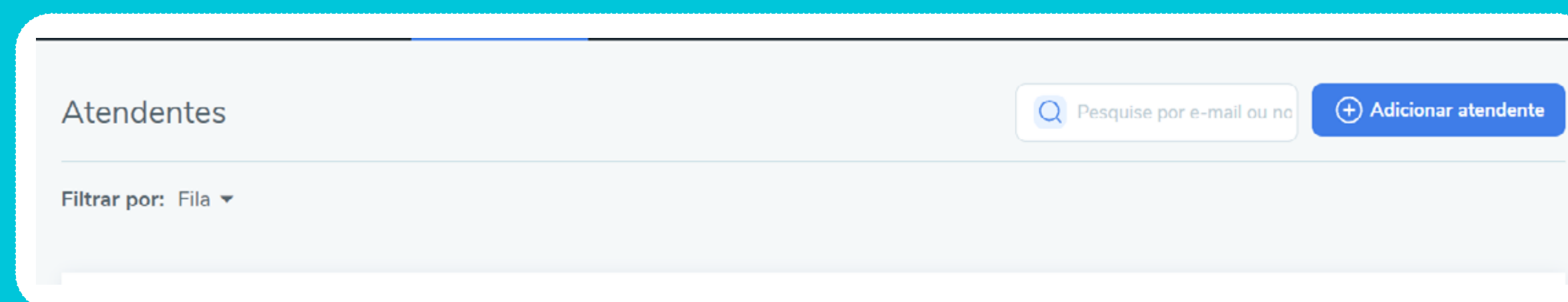


PASSO 03: adicionar atendentes

Para gerenciar as equipes de atendimento, vá até o **módulo de Atendimento** e **selecione a opção Atendente**. Clique no botão **+ Adicionar atendente** e informe:

Email do atendente (esta pessoa deve possuir uma conta Blip válida).

Nome da fila a qual o atendente fará parte.



PASSO 04: definir regras de atendimento

Com o Blip Desk, você pode **definir regras de atendimento** que visam **direcionar o usuário** para os atendimentos humanos entre as equipes cadastradas. Isso é útil, principalmente, se você contar com várias equipes disponíveis. [Neste artigo, explicamos:](#)

Onde encontrar a funcionalidade Regras de atendimento na plataforma Take Blip;

Permissões de acesso da equipe do chatbot à Regras de atendimento;

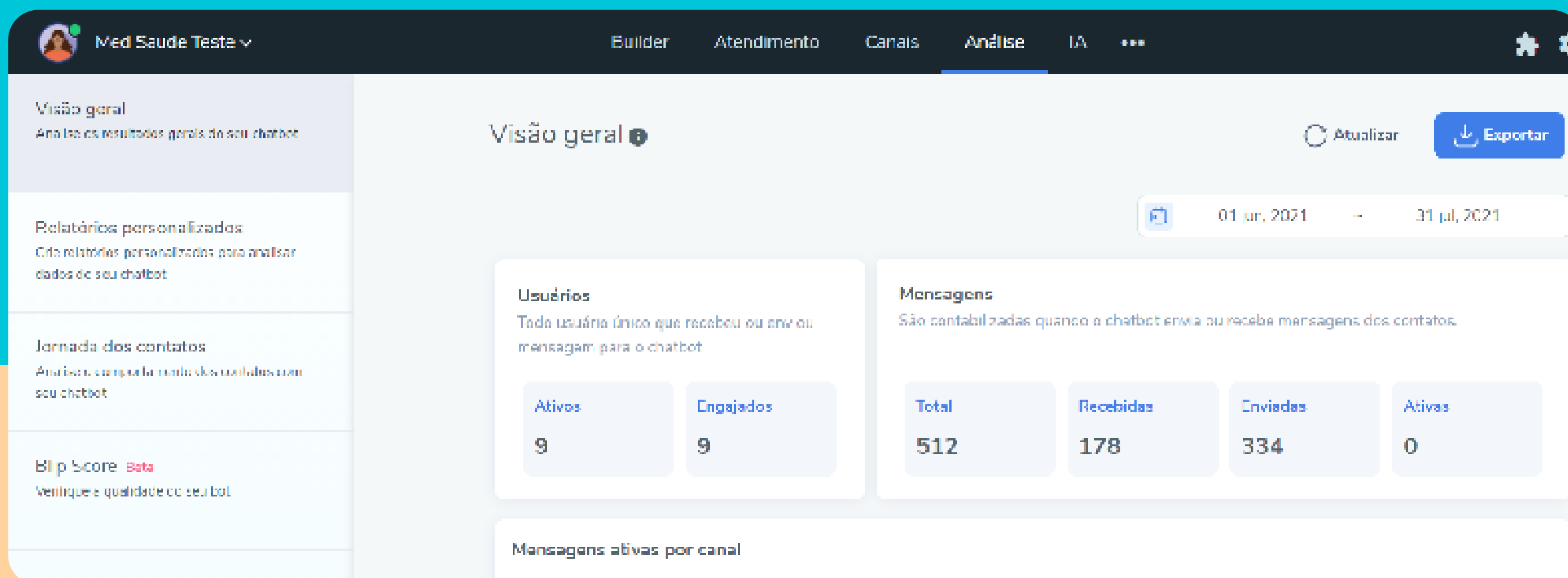
Como distribuir tickets através de uma ou múltiplas condicionais para realizar atendimentos mais assertivos e estratégicos.



PASSO 05: acompanhar relatórios

O que não é mensurado, não é melhorado. Por isso, é tão importante ativar o tracking, acompanhar a jornada de contatos e criar registro de eventos. Confira os relatórios disponibilizados no módulo de Análise.

Visão geral: este é o primeiro relatório disponibilizado no módulo. Aqui, são contabilizados todos os contatos e mensagens trafegadas, incluindo interações durante o atendimento humano ou envio de mensagens ativas. A mensuração deste dados é feita diariamente, sendo que podem haver alterações em um período de até dois dias.



Monitoramento: neste dashboard, temos as principais métricas do seu time de atendimento. É possível visualizar a quantidade de tickets em tempo real, o status dos atendentes, resumo facilitado do dia do atendimento e filtrar por equipe. Para acessar esse relatório, basta clicar em “Análise” e selecionar “Monitoramento”.

A partir dos resultados que estão disponibilizados neste dash, você pode ter alguns questionamentos, como: por que tem muitos tickets em espera na fila? Ou, por que o tempo de atendimento está tão longo? Por que estamos fechando poucos tickets?



Relatório de atendimento: aqui você acessa informações gerais do seu atendimento através do Blip Desk, de acordo com a data estipulada. Com os dados fornecidos neste relatório é possível ver o andamento do seu time de atendimento, assim como a eficiência dos atendentes.

Uma visualização muito interessante é o desempenho dos tickets por dia, em uma comparação de tickets abertos e fechados. É possível ter também os resultados dos atendentes, por equipe e por tags, conforme período selecionado. Para acessar este dash, basta clicar em **“Análise”** e selecionar **“Relatórios de Atendimento”**.



Relatórios personalizados: a partir da ação de registro de eventos, é possível criar relatórios personalizados, com gráficos ou contadores, que exibem os dados registrados em uma categoria. Assim, você consegue ter uma análise assertiva dessas informações e controle na gestão do seu chatbot.

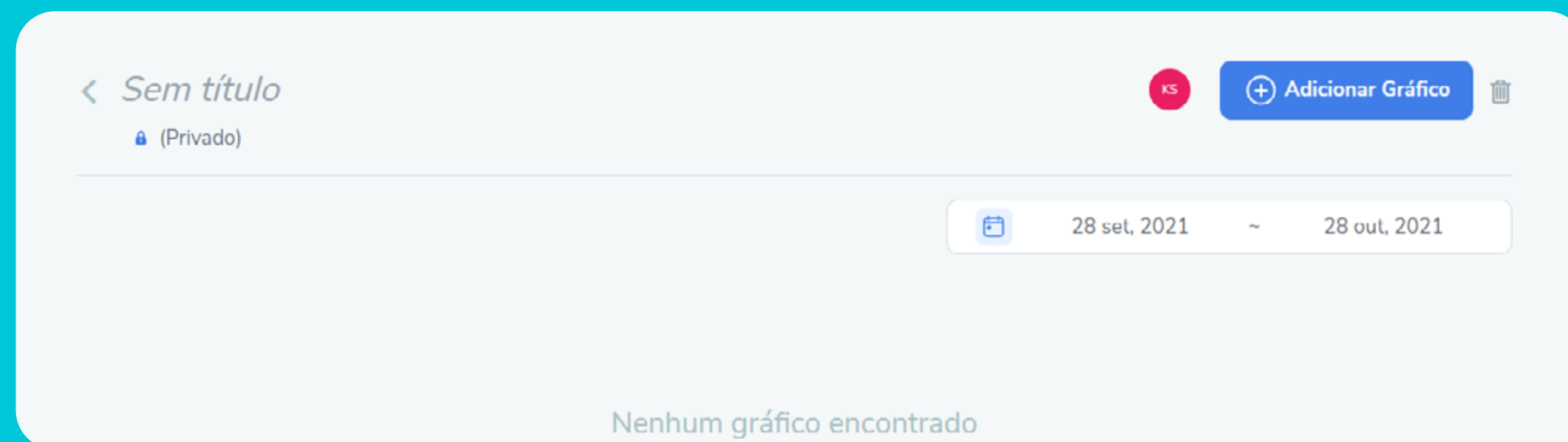
Para criar estes relatórios, acesse o menu de Análise e selecione **Relatórios personalizados**.

Relatórios personalizados



Criar relatório

Nomeie o seu relatório e clique em **Adicionar Gráfico**. Neste momento, você deverá escolher o formato que considera mais interessante para a sua análise. Você pode adicionar



Em seguida, dê um título para o seu gráfico, escolha a dimensão — para registro de eventos escolha Eventos Personalizados —, por fim, defina a categoria correspondente. É a mesma categoria que você atribui ao criar um registro de evento.

Conclusão

Como vimos, construir um chatbot com Take Blip não tem segredo. O primeiro passo é desenhar o fluxo. Você pode fazer isso em ferramentas como Figma e Miro, ou mesmo num papel. A ideia aqui é você ter um ponto de partida, saber aonde quer chegar e quais os passos serão necessários. Dessa forma, fica muito mais fácil criar o fluxo posteriormente.

Já na plataforma Take Blip, você irá replicar o fluxo planejado. É importante prestar atenção nos input inesperados, nas condições de saída de cada bloco, na criação de registro de eventos em blocos estratégico e no tracking automático. Também indicamos o uso dos recursos “expirar sessão” e “tempo de inatividade do usuário”.

Por fim, mas não menos importante, sempre metrifique e analise a performance do bot e das suas equipes de atendimento. Assim, você irá entender os pontos que precisam ser melhorados e quais estratégias estão dando certo.

Ufa! Bastante coisa, não é mesmo? Mas, calma, que seguindo os passos acima e indicações de materiais de apoio, seu contato inteligente será um sucesso!

Quer alavancar ainda mais seu negócio? Vá além com Take Blip.
Clique no botão abaixo e fale com especialistas!

Fale com especialistas!

