Sumário

Apresentação 2

Sobre a extensão 3

Instalação 4

Manual de uso 5

Termos de Uso 11

Apresentação

A extensão **Emergencial Messages** foi desenvolvida por **Take Blip.** Atualmente, ela se encontra na versão **1.0.0** e está disponível nos idiomas **português, inglês e espanhol.**

O cliente está passando por alguma instabilidade? Precisa que uma mensagem seja passada para o usuário durante esse tempo? A extensão Emergencial Messages permite que você cadastre mensagens para que sejam mostradas para o usuário em qualquer ponto do fluxo! Nela você cria o título e o conteúdo da mensagem, assim como os dias e horários iniciais e finais em que a mensagem deverá aparecer!

Este manual tem como objetivo auxiliar os usuários da plataforma Blip na instalação, configuração e utilização da extensão **Emergencial Messages.**

Em adendo, qualquer dúvida de uso pode ser encaminhada para **support@take.net.**

Sobre a extensão

A extensão **Emergencial Messages** foi desenvolvida tendo como objetivo **facilitar o aviso e a notificação de usuários em casos de emergências como instabilidade, lentidão ou algum outro problema que possa estar ocorrendo.**

As seguintes funcionalidades são oferecidas:

- **Criação, edição e remoção de mensagens contendo: título, conteúdo, data e hora inicial e final**
- **Switch para ativação/desativação das mensagens**

Instalação

Ao ativar a extensão pela Blip Store, ela deverá ser instalada nos bots em que **deseja mostrar as mensagens** ou **router de escolha**, Após sua ativação, você já poderá utilizá-la.

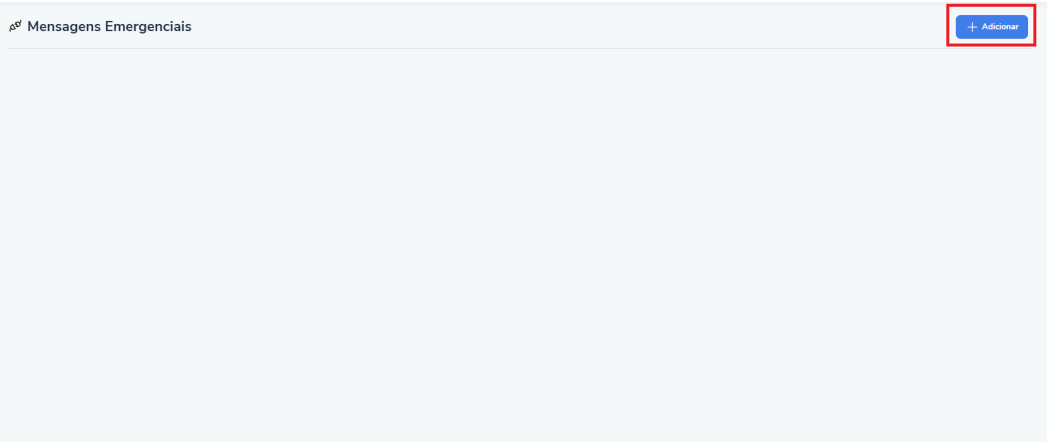
Obs: Caso a extensão seja instalada no router o contexto roteador deve ser compartilhado com os sub bots e os blocos onde essas mensagens serão mostradas podem ser criados em qualquer sub bot que esteja conectado como serviço do router.

Manual de uso

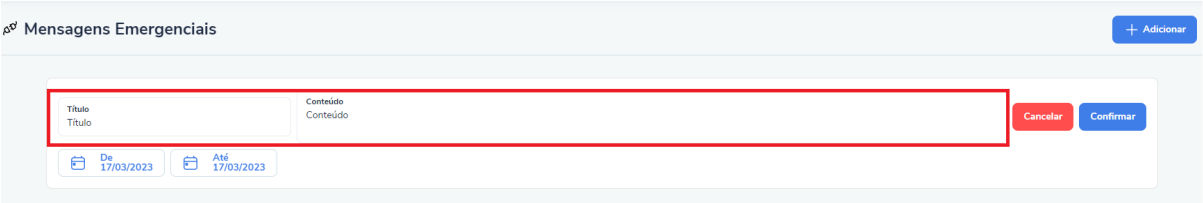
Apenas usuários com permissão de administrador conseguem utilizar o plugin.

- **Criação de mensagens**

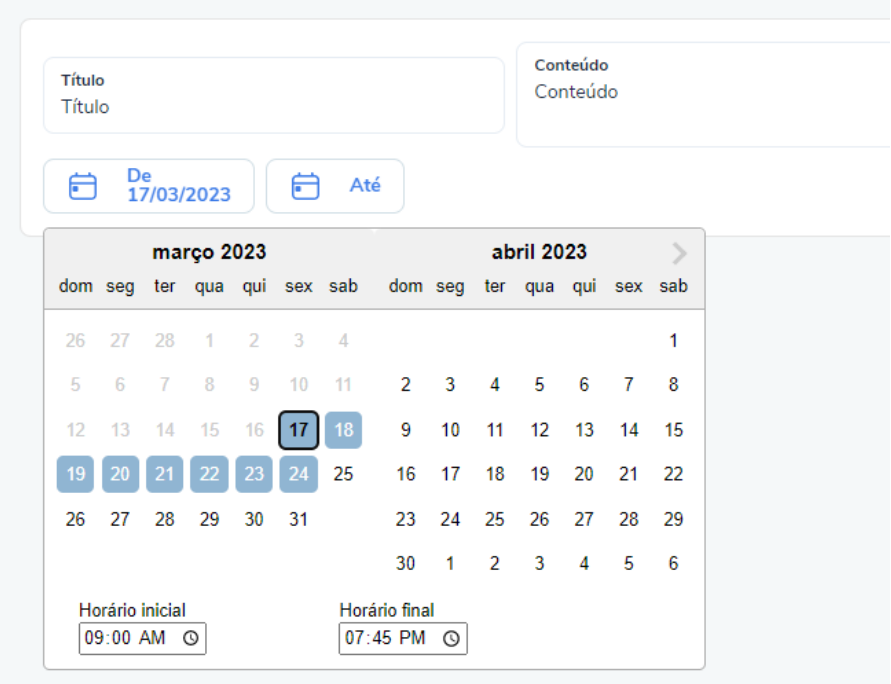
Para criar mensagens basta clicar no botão “Adicionar” no canto superior direito do plugin, um card com os campos a serem preenchidos irá aparecer para que você possa criar sua mensagem:



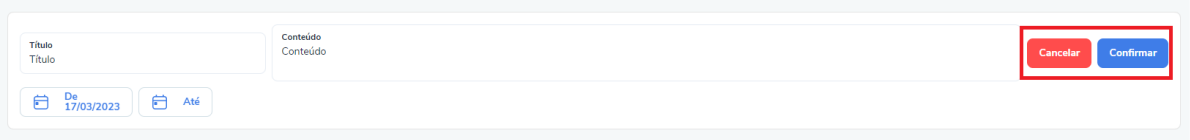
Você verá então campos para preencher o título, conteúdo e o período de tempo no qual deseja que a mensagem fique ativa:



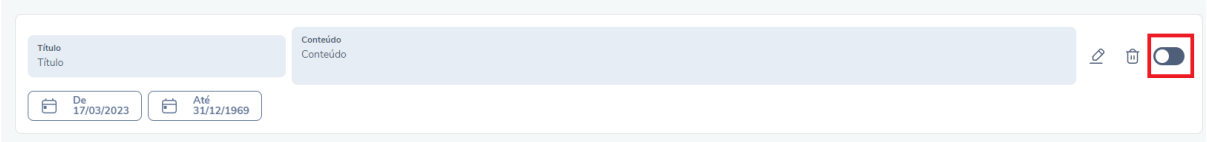
Ao clicar em um dos botões de data, você verá um calendário, onde poderá selecionar a data inicial e final assim como a hora inicial e final em que a mensagem deve ser mostrada:



Assim que preencher todos os campos, basta clicar em confirmar para salvar sua mensagem, ou cancelar caso não queira mais criá-la.

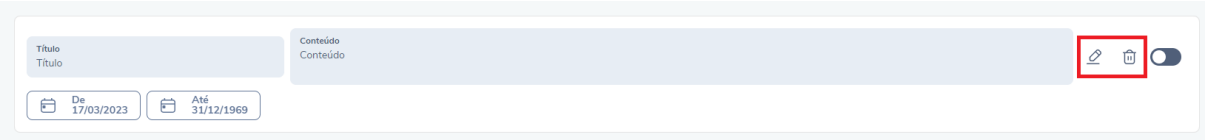


Assim que criada, para ativar a mensagem basta clicar no botão switch no lado direito do card. mensagens ativadas são mostradas para o usuário no fluxo se ele estiver dentro do período de tempo selecionado na criação. Se uma mensagem está desativada, ela não será mostrada.



- **Edição e Exclusão de mensagens**

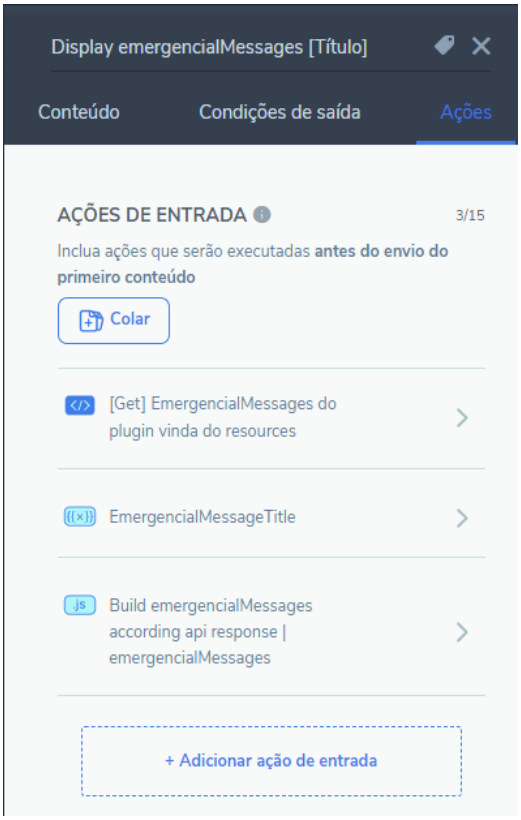
Para editar ou excluir uma mensagem criada, basta clicar nos ícones de **lápis** para editar, ou no ícone de **lixeira** para excluir a mensagem.



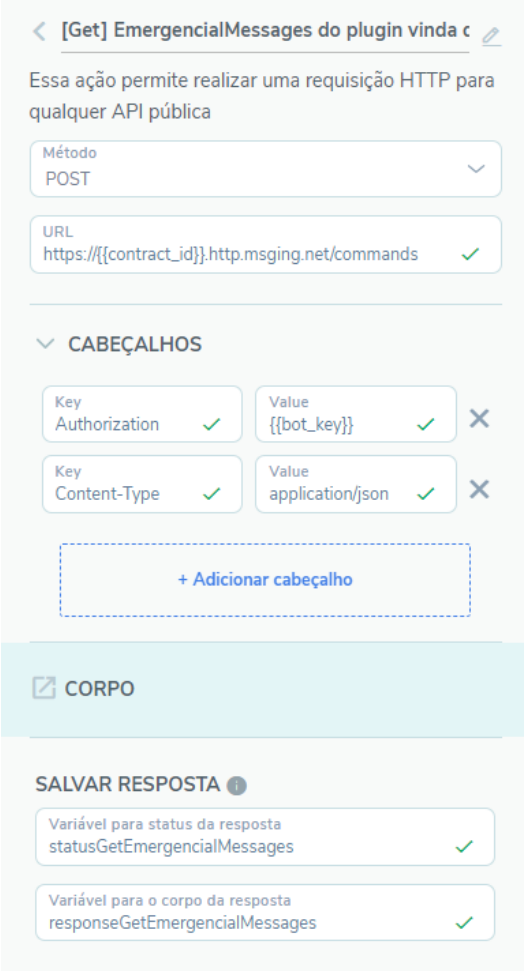
- **Bloco de mensagem emergencial no builder**

Após criada, para que a mensagem seja mostrada para o usuário é necessário a criação de um bloco para esta mensagem no builder, no ponto do fluxo que desejar mostrá-la, se a extensão está instalada em um router, os blocos de mensagem devem ser criados no sub bot desejado. (clicando aqui você pode baixar o bot de exemplo)

Neste bloco teremos três ações para que a mensagem seja construída e mostrada da maneira correta:

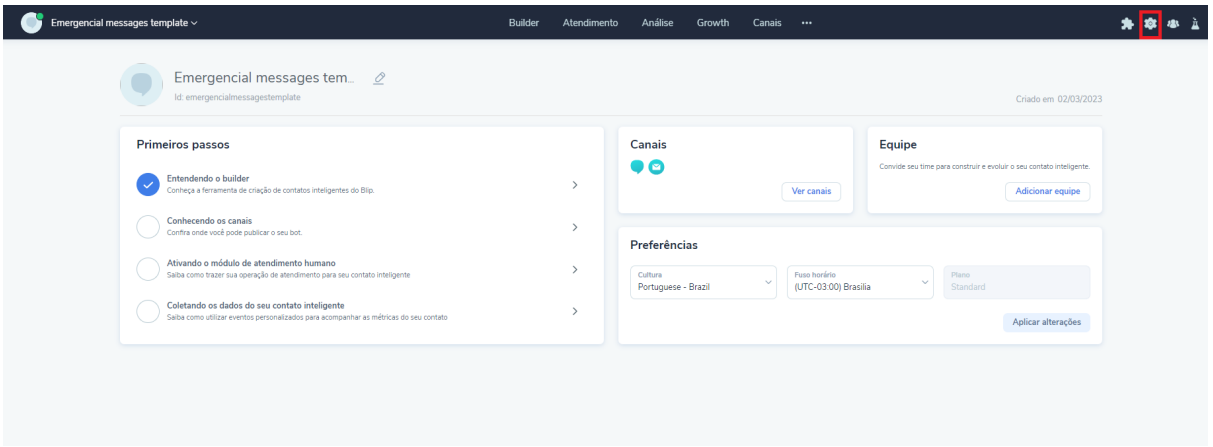


Primeiro realizamos a seguinte requisição http para pegar as informações que queremos do resources (se a extensão foi instalada em um router a chave deve ser do router, se não, deve ser do sub bot ou bots em que a extensão foi instalada):

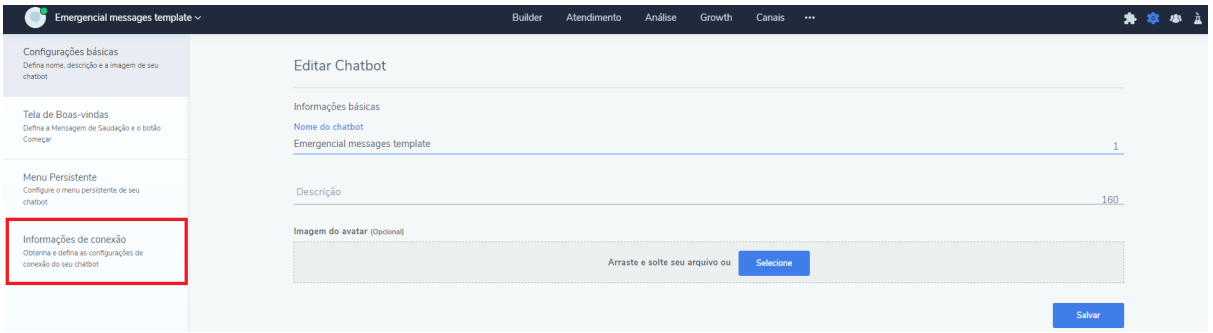


É importante lembrar de alterar a url da requisição para a url de comandos do seu contrato assim como trocar a “{{bot_key}}” no cabeçalho para a chave do bot ou router em que a extensão está instalada.

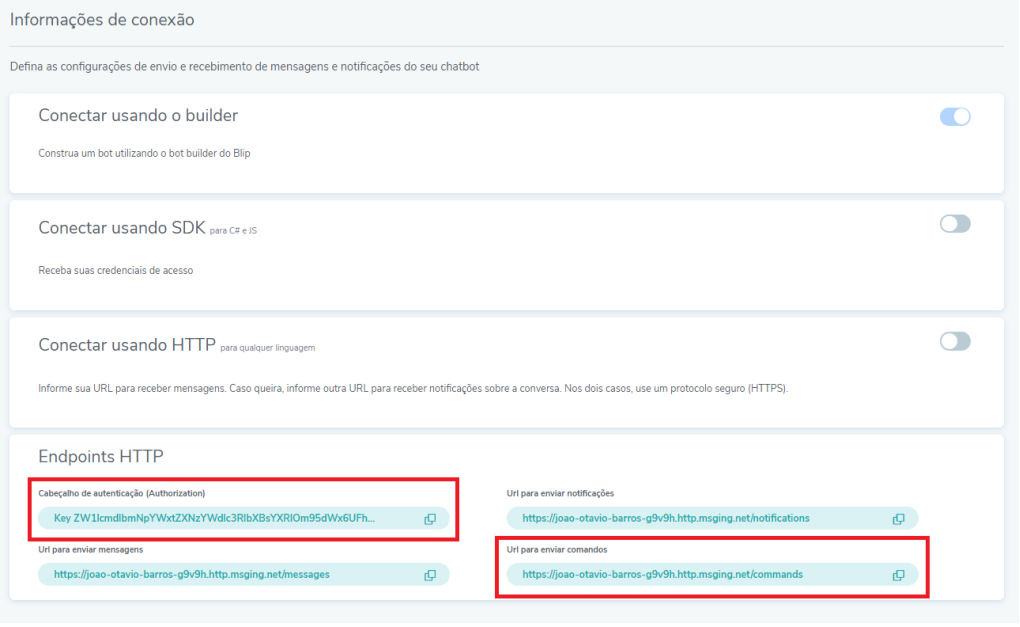
Para encontrar a url de comandos e a bot key de seu bot, primeiro basta entrar nas configurações do bot em que a extensão foi instalada e clicando no ícone de engrenagem no canto superior direito:



Ao entrar nas configurações, vá para Informações de conexão:

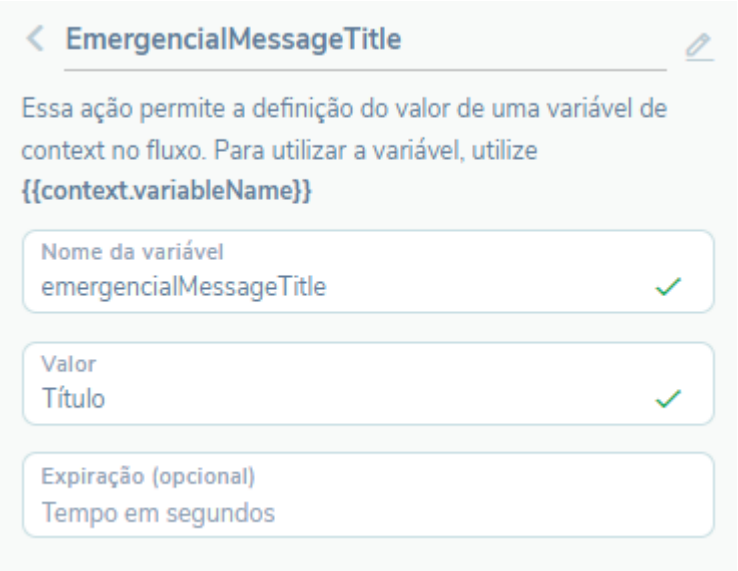


Você verá a seguinte tela. No card “Endpoints HTTP” você encontrará o "Cabeçalho de autenticação (Authorization)” que é a sua api key e também a “Url para enviar comandos” que é a sua url para fazer a requisição.



E por fim temos o corpo da requisição para buscar as mensagens:

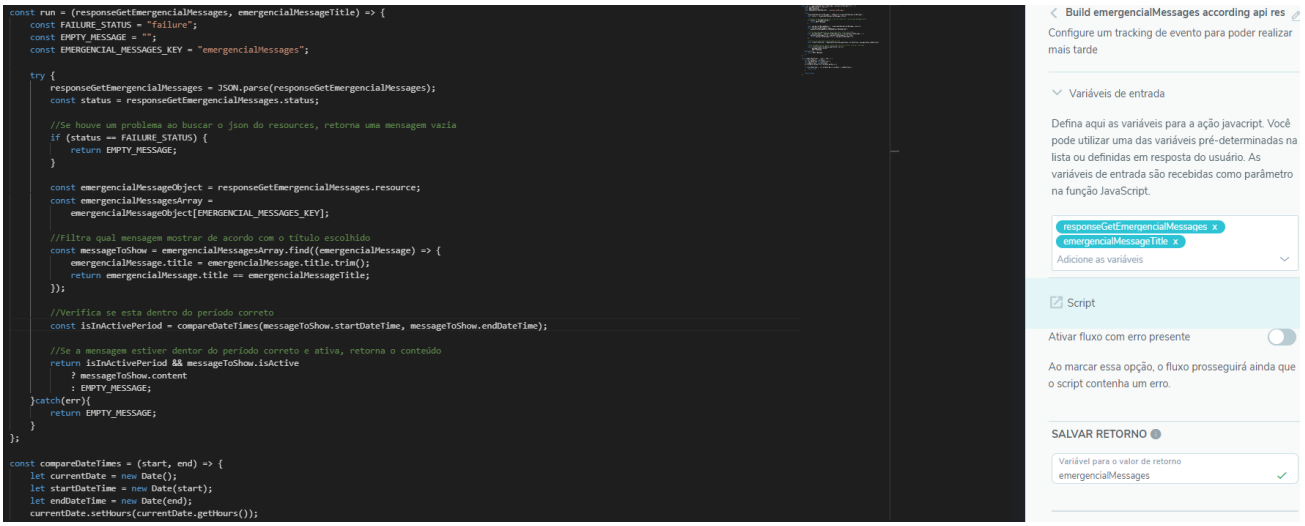
A segunda ação é definir uma variável que receberá o título da mensagem que deseja mostrar, esse título deve ser exatamente igual ao título da mensagem criada no plugin que você deseja que seja mostrada e ele será usado para filtrá-la.



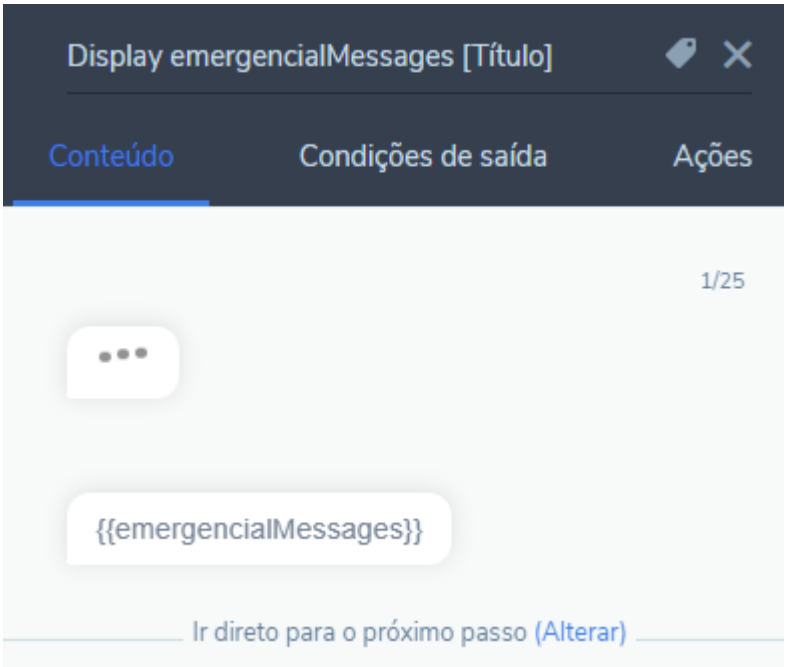
Por fim, criamos uma ação de script onde será filtrada e construída a mensagem que queira mostrar neste bloco, o script tem como entradas o conteúdo vindo do resources e o título da mensagem definido em uma variável anteriormente.

O script verifica se a resposta vinda do resources possui algum conteúdo, filtra qual mensagem mostrar de acordo com o título definido no bloco, verifica se está dentro do período correto selecionado e se a mensagem está

ativada, se tudo estiver certo retorna o conteúdo da mensagem, senão retorna uma mensagem vazia para que passe direto pelo bloco sem mostrá-la. (você pode encontrar esse script no final desta seção)



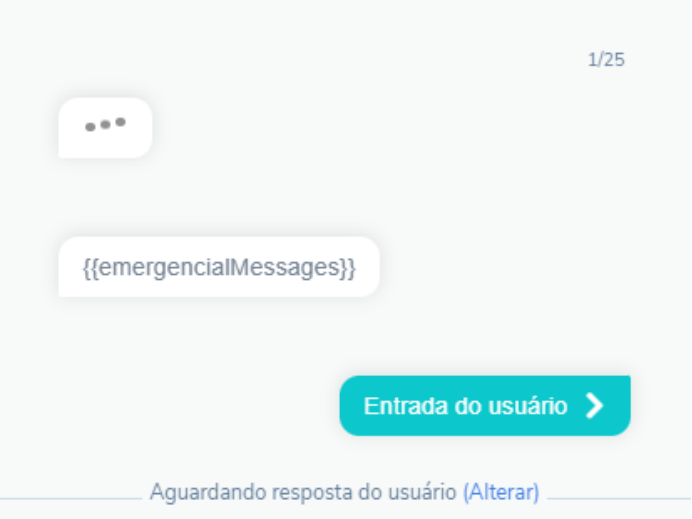
Com isso temos a mensagem pronta para ser mostrada, basta utilizar a variável de retorno do script na mensagem do bloco.



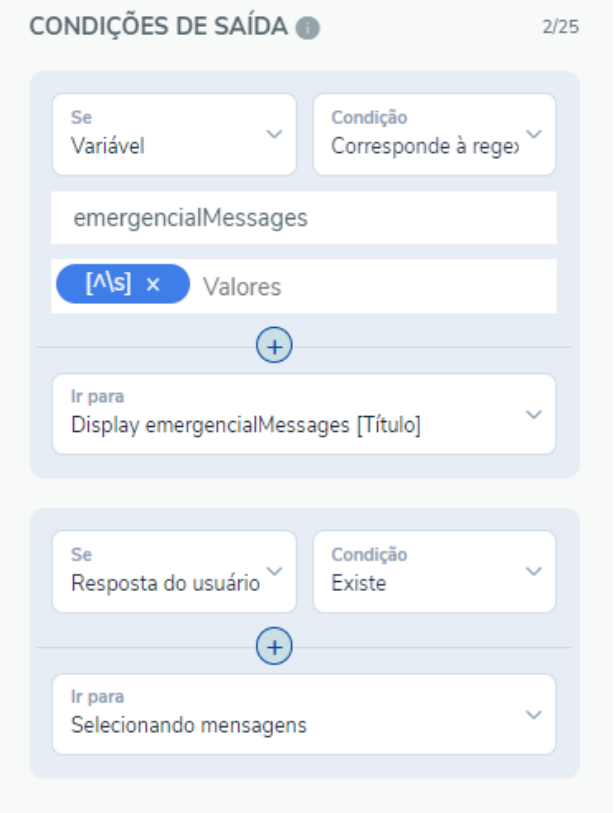
Este bloco de exemplo não possui nenhuma condição de saída, ao mostrar a mensagem o fluxo de mensagens segue normalmente. Caso queira impedir o usuário de seguir no fluxo após mostrar esta mensagem basta fazer o seguinte:

Primeiro basta fazer com que o bloco espere por um input do usuário, clicando em alterar e então em aguardar resposta, como mostra na imagem abaixo.

Feito isso, você deverá ver um indicador de mensagem verde para a entrada do usuário:



Por fim, basta adicionar a condição de saída para manter o usuário em um loop nesta mensagem:



A primeira condição verifica se a variável "emergencialMessages" corresponde ao regex "[^\s]", isso significa que se houver qualquer conteúdo na variável, o usuário deverá ir para o próprio bloco em que ele se encontra, ficando assim em um loop até que o período ativo da mensagem seja finalizado e ela não retorne mais nenhum conteúdo.

A segunda condição apenas manda o usuário para o bloco seguinte no fluxo.

Então, como as condições são verificadas de cima para baixo, primeiro é verificado se há conteúdo na mensagem emergencial, se houver o usuário fica em um loop, se não, o fluxo continua normalmente. (aqui você pode baixar um bot com esse exemplo)

- **Script de tratamento da mensagem emergencial**

Abaixo está o script para ser adicionado ao bloco de mensagem emergencial:

```
const run = (responseGetEmergencialMessages, emergencialMessageTitle)
=> {

   const FAILURE_STATUS = "failure";

   const EMPTY_MESSAGE = "";

   const EMERGENCIAL_MESSAGES_KEY = "emergencialMessages";


   try {

       responseGetEmergencialMessages =
JSON.parse(responseGetEmergencialMessages);

       const status = responseGetEmergencialMessages.status;


       //Se houve um problema ao buscar o json do resources, retorna
uma mensagem vazia

       if (status == FAILURE_STATUS) {

           return EMPTY_MESSAGE;

       }


       const emergencialMessageObject =
responseGetEmergencialMessages.resource;

       const emergencialMessagesArray =

           emergencialMessageObject[EMERGENCIAL_MESSAGES_KEY];


       //Filtra qual mensagem mostrar de acordo com o título escolhido
```

```
        const messageToShow =
emergencialMessagesArray.find((emergencialMessage) => {
            emergencialMessage.title = emergencialMessage.title.trim();
            return emergencialMessage.title == emergencialMessageTitle;
        });


        //Verifica se esta dentro do período correto
        const isInActivePeriod =
compareDateTimes(messageToShow.startDateTime,
messageToShow.endDateTime);


        //Se a mensagem estiver dentor do período correto e ativa,
retorna o conteúdo
        return isInActivePeriod && messageToShow.isActive
            ? messageToShow.content
            : EMPTY_MESSAGE;
    }catch(err){
        return EMPTY_MESSAGE;
    }
};


const compareDateTimes = (start, end) => {
    let currentDate = new Date();
    let startDateTime = new Date(start);
    let endDateTime = new Date(end);
```

```
    currentDate.setHours(currentDate.getHours());


    if (startDateTime <= currentDate && currentDate <= endDateTime) {
        return true;
    }


    return false;
};
```

Termos de Uso

Ao instalar esta extensão você concorda com os termos de uso e política de privacidade do Blip.

Os dados coletados nesta extensão serão utilizados para possibilitar melhorias e evoluções nas extensões e plataforma, e não são compartilhados com terceiros. A utilização das extensões representa uma interação com a Plataforma Blip.

O Cliente (Controlador dos dados pessoais), ao fazer uso desta interação, declara ciência e concordância de que se, neste ato, der causa a qualquer incidente de vazamento de dados, será exclusivamente responsável nos termos da legislação vigente, não havendo qualquer responsabilidade da Curupira S.A. (Take.Blip).