

# Informe de Disponibilidade - GSE

Fluxo automatizado de informações

Autor: João Henrique Catarino

## Sumário

Objetivo.....	3
Resultado esperado .....	3
Introdução .....	3
Planejamento.....	4
Ferramentas utilizadas:.....	4
Microsoft Forms .....	4
Microsoft Power Automate .....	4
Microsoft Excel.....	4
Microsoft Teams .....	4
Desenvolvimento .....	5
Entrada de dados (Forms) .....	5
Escopo do formulário .....	5
Banco de dados .....	6
Fluxo para disparar informações .....	8
Estrutura.....	8
Compreendendo as expressões das variáveis.....	10
Função string() .....	10
Funções sub(), Int() e body() .....	10
Função int() .....	10
Possíveis erros.....	10
Função body().....	11
Expressão completa: .....	12
Cartões Adaptáveis.....	13
Referências Bibliográficas.....	14

## Objetivo

Melhorar a visibilidade quanto a disponibilidade de equipamentos e acompanhamento diário de evolução.

## Resultado esperado

- Melhor comunicação entre os setores GSE e Operação
- Atualizações em tempo real
- Contabilizar e mensurar a disponibilidade média

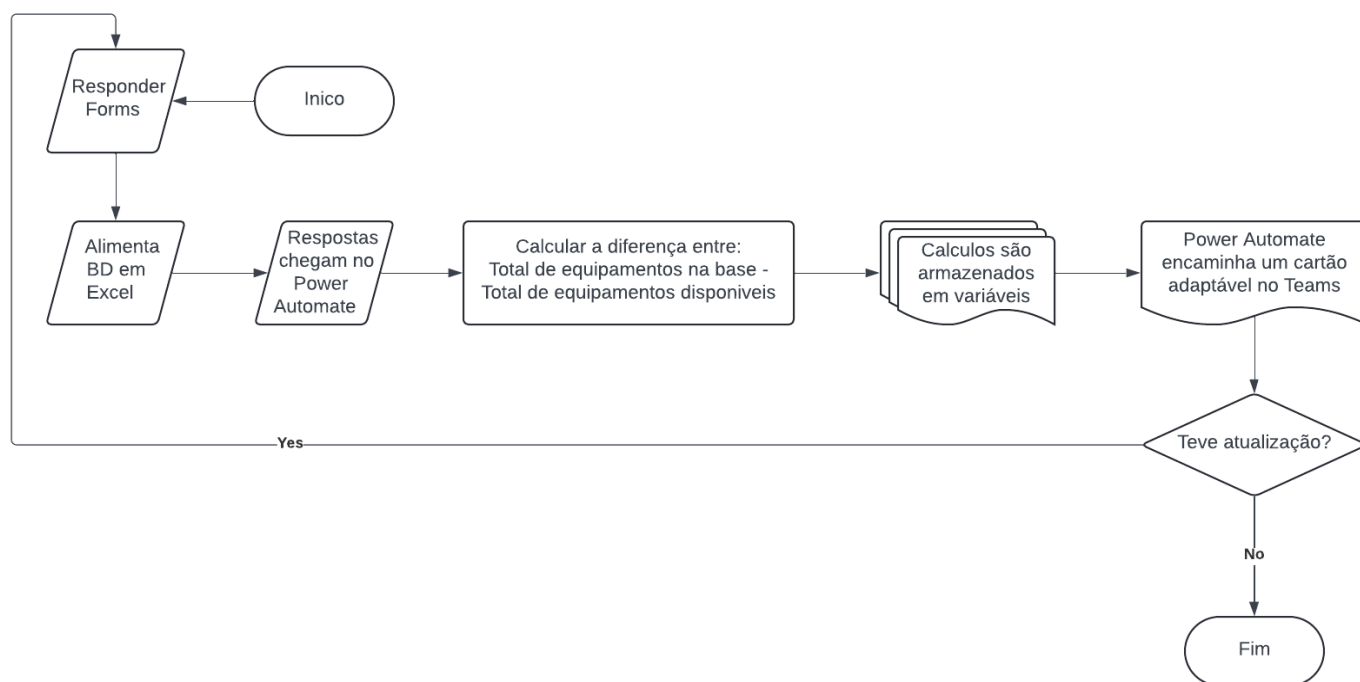
## Introdução

Com a mudança de localização do setor operacional (OCC) ficou difícil a comunicação entre os setores de manutenção e operação, onde o operacional necessita constantemente de atualizações quanto a disponibilidade de equipamentos. Dada a problemática, vislumbrei a oportunidade de utilizar-se o Aplicativo Power Automate juntamente com o MS Teams, disparando informes diariamente, três vezes ao dia.

Este arquivo tem como objetivo denotar as atualizações e etapas realizadas durante a execução e elaboração do fluxo.

## Planejamento

### Fluxo de ações Power Automate



## Ferramentas utilizadas:

### Microsoft Forms

Utilizado para receber as atualizações dos equipamentos disponíveis, onde o usuário deve inserir a quantidade de equipamentos em operação.

### Microsoft Power Automate

Ao receber a resposta do usuário no formulário (Forms), realizará o tratamento dos dados e informará os colaboradores da coordenação (OCC – Operação)

### Microsoft Excel

Utilizado para armazenar todas as respostas enviadas no formulário

### Microsoft Teams

Enviar cartão adaptável em um grupo



## Desenvolvimento

Processo de elaboração e acompanhamento das etapas realizadas, e resultados alcançados.

### Entrada de dados (Forms)

Antes de começar a trabalhar com o fluxo em si, precisa-se pensar em como serão inseridos os dados, para isso utilizei o Microsoft Forms. Porque utilizar? É uma ferramenta sem custos adicionais; possibilidade de utilização por diversos colaboradores.

### Escopo do formulário

**DISPONIBILIDADE DE EQUIPAMENTOS**

Olá, Joao. Quando você enviar este formulário, o proprietário verá seu nome e endereço de email.

\* Obrigatória

1. QUANTOS **ATC** INOPERANTES \*

- Quantidade na base **10**

O valor deve ser um número

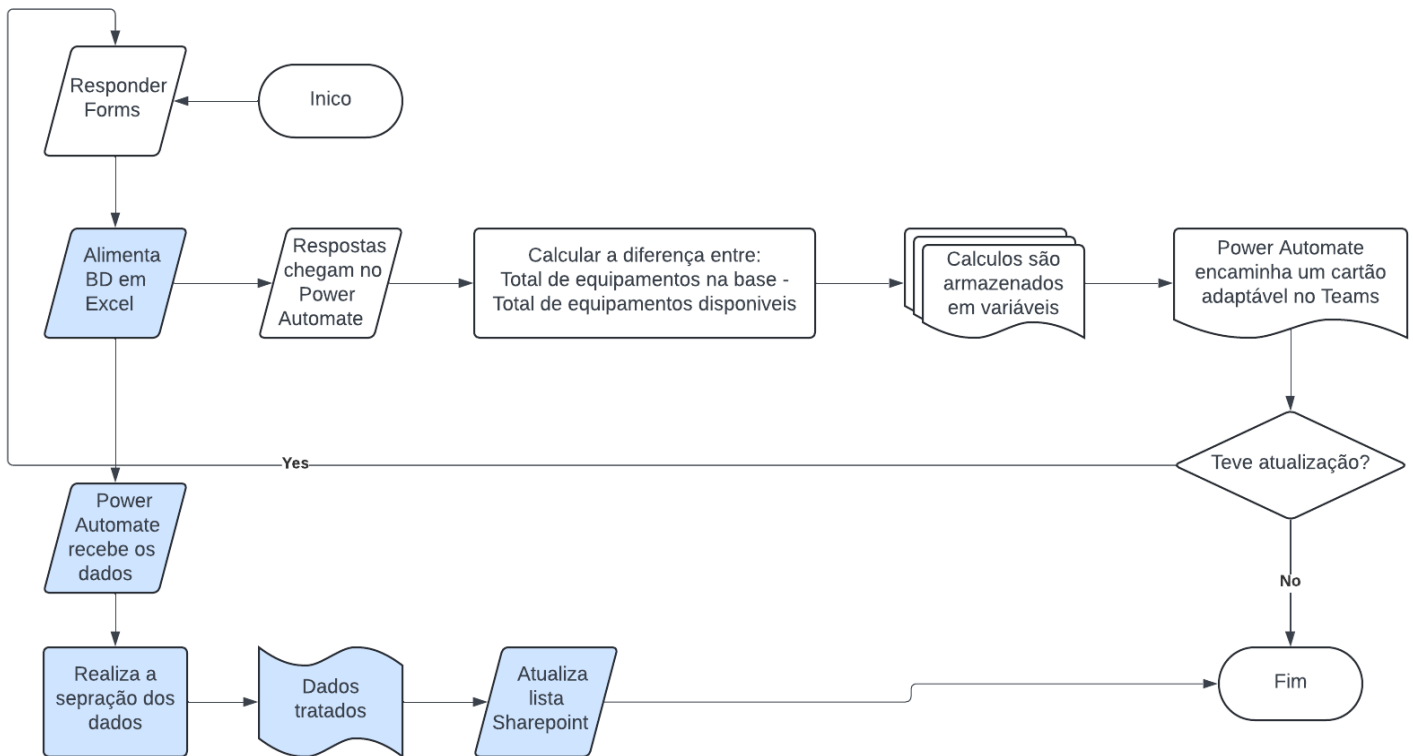
Inserção da quantidade de equipamentos parados e/ou em manutenção

Qual equipamento se encontra inoperante, seguido de uma colinha da quantidade total de equipamentos na base

No total teremos 5 campos destinados a categorias de equipamentos diferentes

## Banco de dados

Para o banco de dados será utilizado o próprio Excel, que armazena todas as respostas enviadas no formulário de maneira automática. *Não é a melhor solução para um banco de dados, ideal seria armazenar estas informações em uma lista do Sharepoint, porém, caso seja*



*necessário pode-se ser elaborado um outro fluxo para atualizar uma lista.*

Mas o nosso foco aqui será o Excel, os dados serão estruturados da seguinte maneira:

Aqui será obtido informações valiosas como:

- E-mail e Nome do responsável por aferir a quantidade de equipamentos em manutenção;
- Hora de Início e Término;

- Respostas do formulário

ID	Hora de início	Hora de conclusão	Email	Nome	Hora da última modificação	QUANTOS ATC INOPERANTES	QUANTO
3	31/07/24	31/07/24				3	6
4	01/08/24	01/08/24				4	7
5	01/08/24	01/08/24				4	7
6	01/08/24	01/08/24				3	6
7	02/08/24	02/08/24				3	6
8	02/08/24	02/08/24				3	6
9	02/08/24	02/08/24				3	7
10	03/08/24	03/08/24				4	7
11	04/08/24	04/08/24				3	7
12	02/08/24	04/08/24				3	5
13	05/08/24	05/08/24				3	6
14	05/08/24	05/08/24				3	
15	06/08/24	06/08/24				3	
16	07/08/24	07/08/24				3	
17	07/08/24	07/08/24				3	
18	02/08/24	07/08/24				3	7
19	09/08/24	09/08/24				3	4
20	13/08/24	13/08/24				2	5
21	14/08/24	14/08/24				3	5
22	15/08/24	15/08/24				2	6
23	15/08/24	15/08/24				2	7

- Nota: caso queira trabalhar com números inteiros, lembre-se de configurar a entrada de dados no Forms para o tipo inteiro, caso contrário o Excel receberá as respostas como texto, acrescentando um Apostrofo 'na frente do número.

Essas informações são importantes caso precise lincar em um Power BI afim de obter insights como por exemplo: A média de equipamentos inoperantes por dia e/ou por horário do dia. Ou até mesmo qual foi o pico por hora e que se obteve mais equipamentos danificados.

## Fluxo para disparar informações

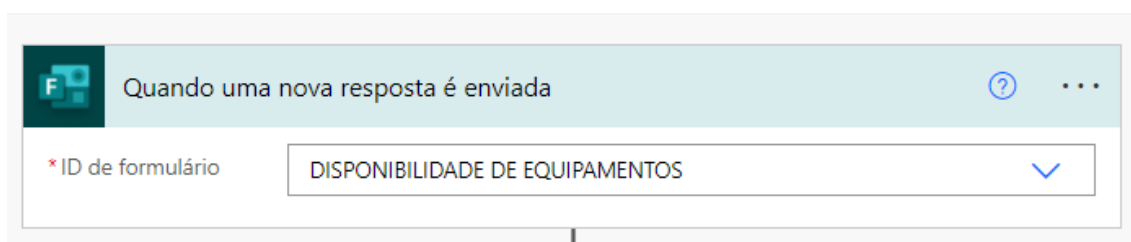
Toda a informatização da disponibilidade ocorrerá com a ferramenta Power Automate, onde, terá a entrada dos dados (Obtidos através do Forms) sendo nosso gatilho para iniciar todo o processo. O escopo geral do fluxo é, conforme descrito na secção

[planejamento](#) :

### Estrutura

A estrutura que utilizei foi a de um fluxo em nuvem automatizado que consistindo em:

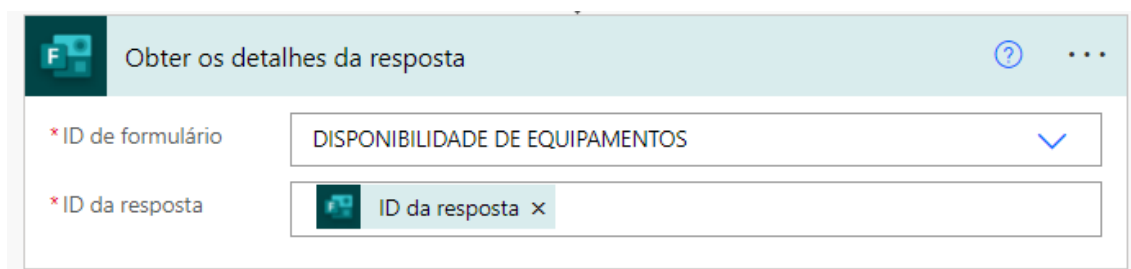
1. Gatilho, o app espera uma determinada ação ser iniciada para desencadear uma serie de cálculos e tratamento de dados, no nosso caso, o gatilho será o envio de uma nova resposta a um formulário:



Quando uma nova resposta é enviada

\* ID de formulário: DISPONIBILIDADE DE EQUIPAMENTOS

2. Após receber esse gatilho, ainda é necessário obter os detalhes da resposta enviada, por isso é adicionada mais uma ação:

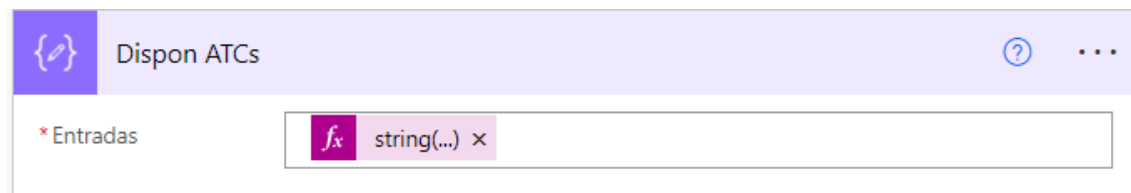


Obter os detalhes da resposta

\* ID de formulário: DISPONIBILIDADE DE EQUIPAMENTOS

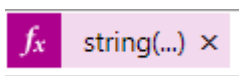
\* ID da resposta: ID da resposta

3. Neste momento, precisamos calcular a quantidade de equipamentos disponíveis, que é dado pela diferença: **Total de equipamentos na base - Total de equipamentos em manutenção**
4. Para isso deve-se criar uma Variável para cada tipo de equipamento e efetuar este cálculo, peguemos como exemplo a disponibilidade dos ATC's:



Dispon ATCs

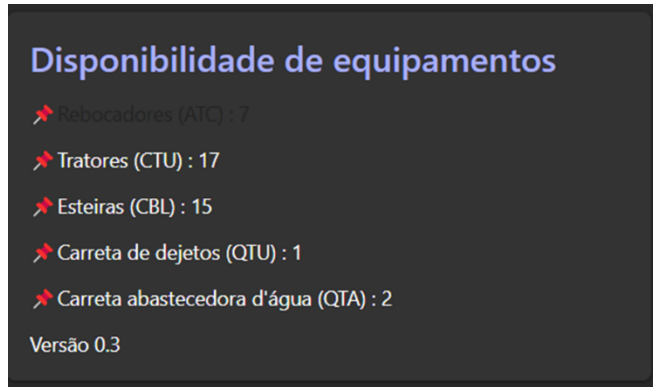
\* Entradas: string(...)

5. O Valor da variável ->  é (entenda a estrutura [clikando aqui](#)):



```
a. string (sub (10, int(body('Obter_os_detalhes_da_resposta')?
['ref4d235fea7a4b0588b84423a3b13175'])))
```

6. Após tratar cada uma das variáveis, será enviado um cartão adaptável no chat do Teams, mas específico em um grupo onde todos os colaboradores estão inclusos. De primeiro momento, o cartão estruturado da seguinte forma:



Ao decorrer das semanas, verificou-se a necessidade de modificar sua estrutura, levando ao seguinte design [clique aqui](#) para saber mais sobre cartões adaptáveis:



## Compreendendo as expressões das variáveis

Compreendendo melhor as funções usadas em cada uma das variáveis

### Função string()

Lembra quando disse que o precisamos configurar o tipo de dados que a resposta do formulário deve obter, caso não [clique aqui](#) para relembrar. Essa função obtém os dados do tipo numérico e os converte em texto, isso nos será importante mais pra frente quando entrar na parte de cartões adaptáveis.

Argumentos: **String(n)** – Converte qualquer valor ‘n’ para uma cadeia de caracteres

### Funções sub(), Int() e body()

**Função sub()** essa função realiza cálculos matemáticos de subtração, neste ponto obtém-se a quantidade total de equipamentos em operação dado pelo argumento

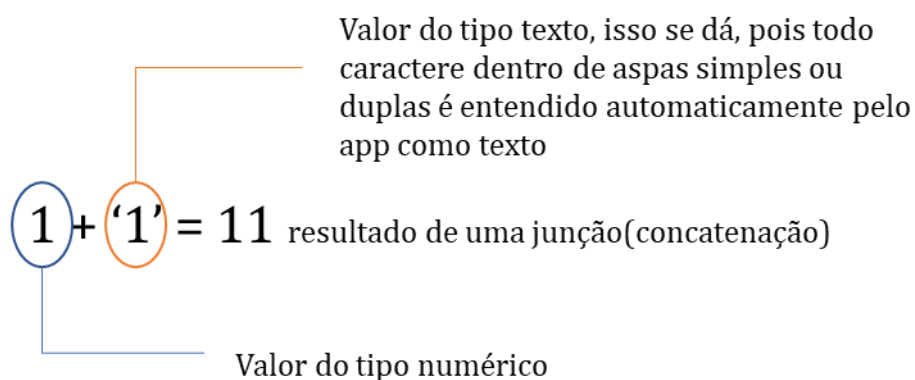
**Sub(Equipamentos na base -*neste caso 1*” - Equipamentos inoperantes)**

### Função int()

Ainda falando sobre conversões, notei que mesmo padronizando a entrada de dados no meu Forms para o tipo numérico ao chegar no Power Automate os dados estão vindo como **String**, para isso utilizo a função **int()** para converter a resposta para o tipo **Numero** – se não o fizer, ao realizar qualquer cálculo matemático, teremos um erro, pois o app tentara substituir ou somar uma cadeia de caracteres a um número.

### Possíveis erros

- Tentar somar uma cadeia de caracteres a um número, para muitos a soma de: **1+1'** será igual a 2. Entretanto não é bem assim que acontece, o resultado disso será uma concatenação, resultando em **11** vamos entender melhor o porquê:

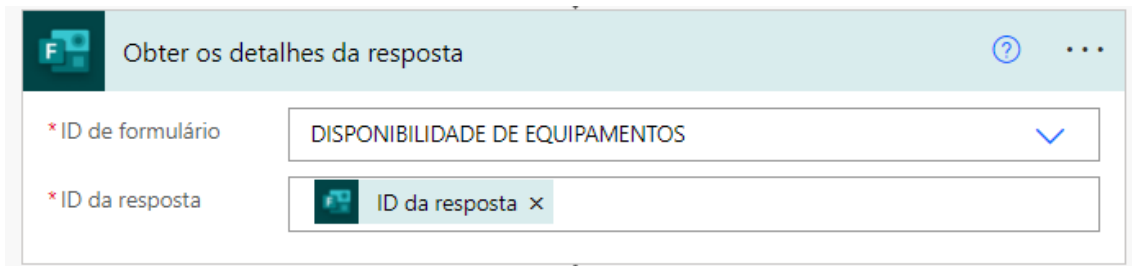


- Nos casos de subtração, obtém-se um erro, pois estamos tentando subtrair um caractere de um número.

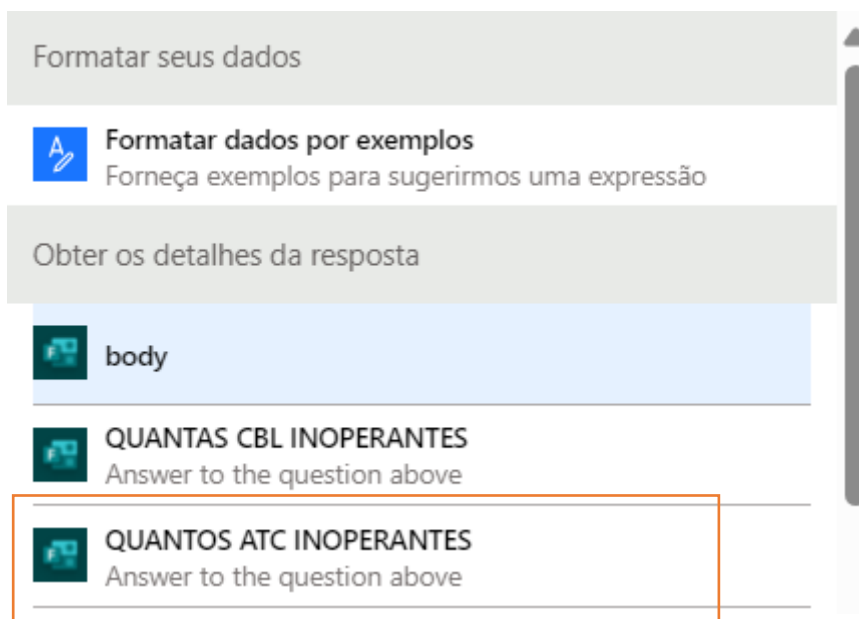
Por isso a importância da conversão dos tipos de dados.

## Função body()

Nada mais é que um argumento utilizado para obter detalhes de alguma ação do fluxo, neste caso está obtendo os detalhes da ação “Obter detalhes da resposta”:



E a resposta que nós queremos aqui é exatamente a quantidade ATC's em manutenção que também é dado pelo conteúdo dinâmico:



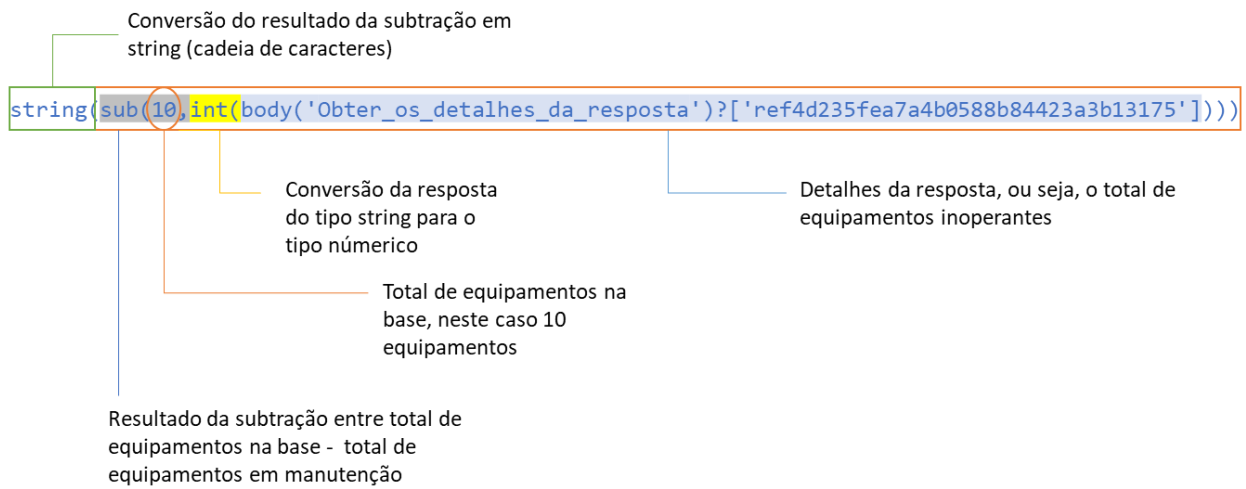
A função **body()** em linhas gerais é a mais tranquila de se obter, pois ao selecionar o item “Quanto ATC inoperantes” automaticamente montará a estrutura:

```
body('Obter os detalhes da resposta')?['ref4d235fea7a4b0588b84423a3b13175']
```

Argumento que faz referência a qual etapa do fluxo desejamos obter maiores detalhes, neste caso os detalhes das respostas enviadas

Não precisamos nos ater a este enorme código, ele faz referência a uma identificação do item “Quanto ATC inoperantes”

## Expressão completa:



Para retornar a leitura do escopo [clique aqui](#).

## Cartões Adaptáveis

São uma maneira flexível e padronizada de exibir informações em interfaces de usuário, como aplicativos, chatbots e plataformas de colaboração (neste caso o Teams). Vamos imaginar uma estrutura de blocos de construção visuais que podem ser personalizadas para exibir diferentes tipos de conteúdo.

Para criar este cartão utilizei um criador de cartões online (Confira na secção [referências bibliográficas](#)), é bem fácil, tem vários modelos, basicamente precisei clicar e arrastar e logicamente pesquisar muito sobre a estrutura de dados JSON.

### Estrutura JSON utilizada para criar este cartão

Todo o código seguirá publicado em meu GitHub, por tanto visite a área de referências para acessá-lo e baixa-lo.

## Referências Bibliográficas

Referência	Link
Adaptive Cards – Designer de cartões gratuito	<a href="#">Designer   Adaptive Cards</a>
Verificador de JSON	<a href="#">JSON Formatter &amp; Validator (curiousconcept.com)</a>
Documentação Microsoft Power Automate – Funções e expressões	<a href="#">Guia de referência para funções de expressão - Azure Logic Apps   Microsoft Learn</a>
Documentação Microsoft Power Automate – Completa	<a href="#">Visão geral dos tipos de fluxos no Power Automate - Power Automate   Microsoft Learn</a>
Código utilizado cartão adaptável	<a href="#">JoaoHCSilva/Disponibilidade GSE: Fluxo automatizado em nuvem para disparo de iformacoes (github.com)</a>