



COMO CRIAR LOGS NO AUTOMATION ANYWHERE?

Guia prático sobre como utilizar os logs
dentro das suas automações.

POR MARTHA BEATRIZ



SEUS BOTS ESTÃO SOBRE CONTROLE?

Sem logs, sua automação pode estar executando no escuro. Aprenda de forma simples a registrar e acompanhar cada passo do seu RPA.

INTRODUÇÃO

Automação sem logs é como dirigir sem farol numa estrada escura. Você pode até seguir em frente, mas não sabe se está na pista certa ou se está vindo um carro na sua direção.

Se você já enfrentou problemas de **rastreabilidade, segurança ou auditoria**, como por exemplo registrar manualmente no bloco de notas tudo o que acontece, esse conteúdo é para você!

O QUE SÃO LOGS?

Os logs são registros de eventos gerados por sistemas, aplicativos ou dispositivos, permitindo o monitoramento, análise e auditoria das atividades.

Por que dedicar **tempo** para aplicar os logs?

Existem 3 principais tipos de aplicação de logs que são podem ser utilizados:

MONITORAMENTO

1

Detecção de anomalias

Permite identificar comportamentos inesperados nos bots, possibilitando ações proativas por parte dos devs para evitar falhas maiores.

Auditoria e conformidade

Mantém um registro detalhado das operações como data, hora, máquina de execução entre outras informações. Cumpre um papel essencial para auditorias e para garantir a conformidade dos RPAs com as políticas corporativas.

Otimização de performance

A análise cuidadosa dos logs ajuda na identificação de gargalos e oportunidades de melhoria nos processos automatizados.

Rastreabilidade

Permite identificar ações suspeitas e prevenir fraudes, registrando quem fez determinada atividade, quando e como. Facilita a investigação de erros e falhas.

Conformidade e regulamentações

O objetivo é evitar penalidades e multas por não conformidade de regulamentações como ISO 27001 (Segurança da informação). Além disso, ajuda a manter a credibilidade da organização.

Resolução de problemas

Redução no tempo de inatividade dos sistemas envolvidos, otimização do desempenho dos bots e melhorias significativas na experiência do usuário ao melhorar a eficiência do processo automatizado.

Controle de acesso

O controle de acesso define quem pode visualizar, modificar e excluir registros, evitando o vazamento de dados sensíveis e evitando a manipulação dos logs.

Criptografia

A criptografia evita que informações importantes do negócio sejam expostas em ataques cibernéticos, garantindo a proteção aos dados mesmo em casos de interceptação.

Monitoramento contínuo

Existem ferramentas de monitoramento que analisam os logs em tempo real para identificar atividades suspeitas, tentativas de invasão ou acessos não autorizados.

Imagine que você está no supermercado passando as suas compras no caixa. Cada produto registrado é adicionado à sua nota fiscal.



Ao final, a nota fiscal deve conter **todos os itens comprados**, permitindo que você confira se tudo está correto antes de ir para casa. Os logs funcionam da mesma forma, criando um “histórico” do que aconteceu.

PONTOS IMPORTANTES

Assim como na nota fiscal, precisamos definir alguns pontos para os logs antes de criá-los:

1

Local de armazenamento - Onde os registros serão salvos? Banco de dados? Arquivo local? Plataforma de monitoramento?

2

Formato esperado - Qual formato será utilizado? JSON, TXT, CSV? Cada formato possui as suas vantagens na hora de analisar os dados.

3

Informações - O que deseja registrar? Data e hora, usuário responsável e status da execução? Quais erros foram encontrados? Defina o que deseja obter de informação.

SOBRE A SOLUÇÃO

A solução foi criada com foco no **reaproveitamento de código** do nosso task bot. Com isso, um **arquivo XML** simples foi empregado como arquivo principal de configuração.

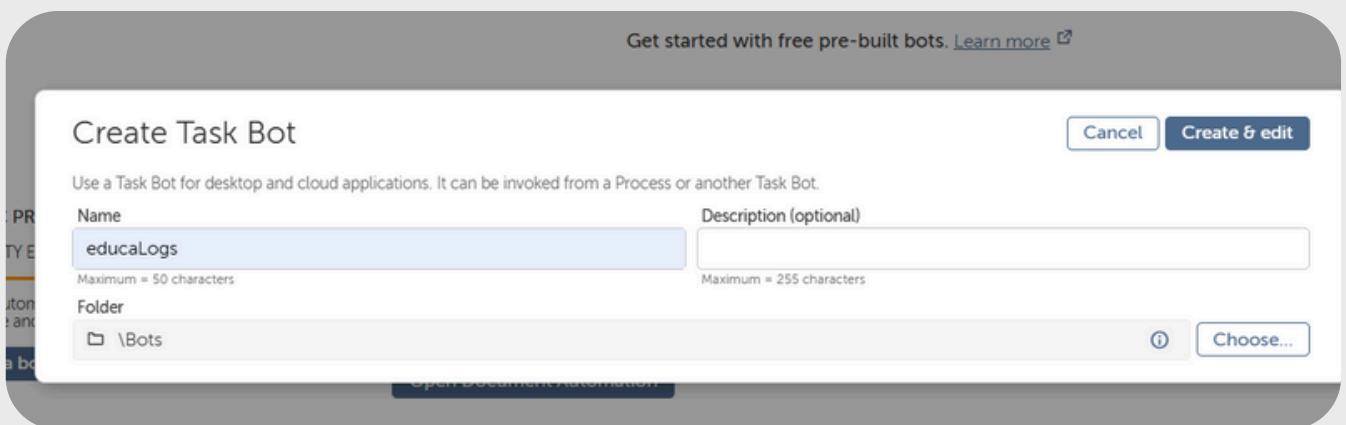
Dentro do XML, temos os **nós** que representam as informações específicas que queremos registrar.

Como o arquivo pode ser amplamente utilizado por outras automações, cada nó pode receber valores que são únicos para cada bot.

Assim, as informações armazenadas podem ser **facilmente recuperadas** dentro do Automation Anywhere! Vamos explorar cada uma das ações a seguir.

CRIANDO SEU TASK BOT DE LOG

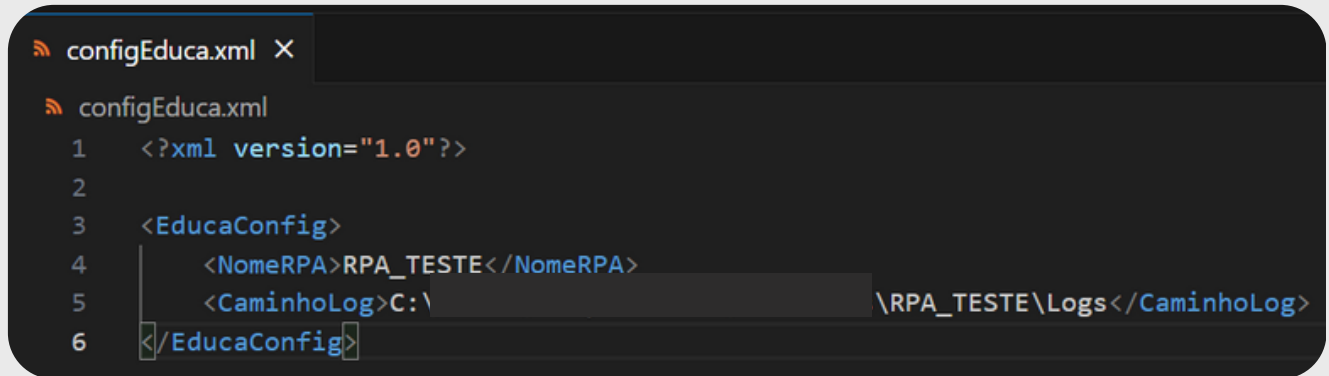
1. Comece criando o seu task bot clicando em **Create a bot**, no Automation Anywhere.



The screenshot shows the 'Create Task Bot' dialog box in Automation Anywhere. At the top, there is a header bar with the text 'Get started with free pre-built bots. [Learn more](#)'. Below this, the dialog box has a title 'Create Task Bot' and two buttons: 'Cancel' and 'Create & edit'. A subtitle reads: 'Use a Task Bot for desktop and cloud applications. It can be invoked from a Process or another Task Bot.' The form contains three fields: 'Name' with the value 'educaLogs' and a note 'Maximum = 50 characters'; 'Description (optional)' with a note 'Maximum = 255 characters'; and 'Folder' with the value '\Bots' and a 'Choose...' button. There is also an information icon (i) next to the folder field.

Esse task bot será importante para utilizar dentro de outras automações. Dê um nome ao seu bot e clique sobre **Create & edit**.

2. Vamos utilizar como exemplo o seguinte arquivo XML:

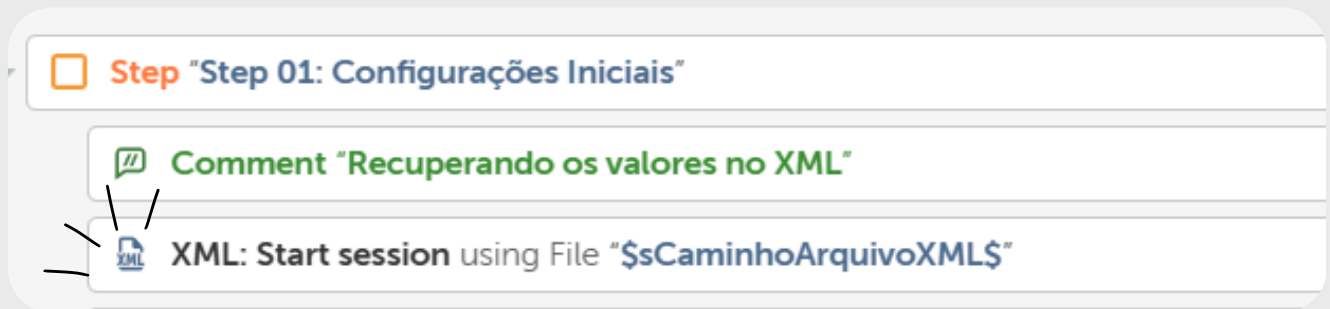
A screenshot of a code editor window titled 'configEduca.xml'. The editor shows the following XML code:

```
1 <?xml version="1.0"?>
2
3 <EducaConfig>
4   <NomeRPA>RPA_TESTE</NomeRPA>
5   <CaminhoLog>C:\[redacted]\RPA_TESTE\Logs</CaminhoLog>
6 </EducaConfig>
```

Dentro dele, observamos apenas dois nós que serão utilizados: **NomeRPA** e **CaminhoLog**.

Para o CaminhoLog, vou utilizar uma pasta local para facilitar a solução.

3. Dentro da Step 01, vamos definir o **Start Session** como **Default**, usando uma variável contendo o caminho do arquivo XML, que será fornecido pelo bot principal.



XML: Start session

Creates a new XML session based on an XML file or specified text
Required bot agent version: 20.11 or above

Session name
" Default (x)

Data Source
☒ File

Variable Control Room file Desktop file

Specific path Regex

" \$sCaminhoArquivoXML\$ (x) Browse...

Atenção!

As **variáveis do task bot** que vão receber dados de outras automações precisam estar marcadas como **Input**.



Use as input

4. A ação **Get Single Node** retorna o valor de cada nó. Como temos apenas dois nós, usamos duas ações, com variáveis a serem preenchidas.

Step "Step 01: Configurações Iniciais"

Comment "Recuperando os valores no XML"

XML: Start session using File "\$sCaminhoArquivoXML\$"

XML: Get single node and assign value to variable \$sNomeRPA\$

XML: Get single node and assign value to variable \$sCaminhoLogsNode\$

XML: Get single node

Retrieves the value of a single node

Required bot agent version: 20.11 or above

Session name

” Default

XPath expression

” \$sNomeRPANode\$

e.g. /bookstore/book/title

Attribute (optional)

”

Assign the output to variable

” sNomeRPA

XML: Get single node

Retrieves the value of a single node

Required bot agent version: 20.11 or above

Session name

” Default

XPath expression

” \$sCaminhoLogsNode\$

e.g. /bookstore/book/title

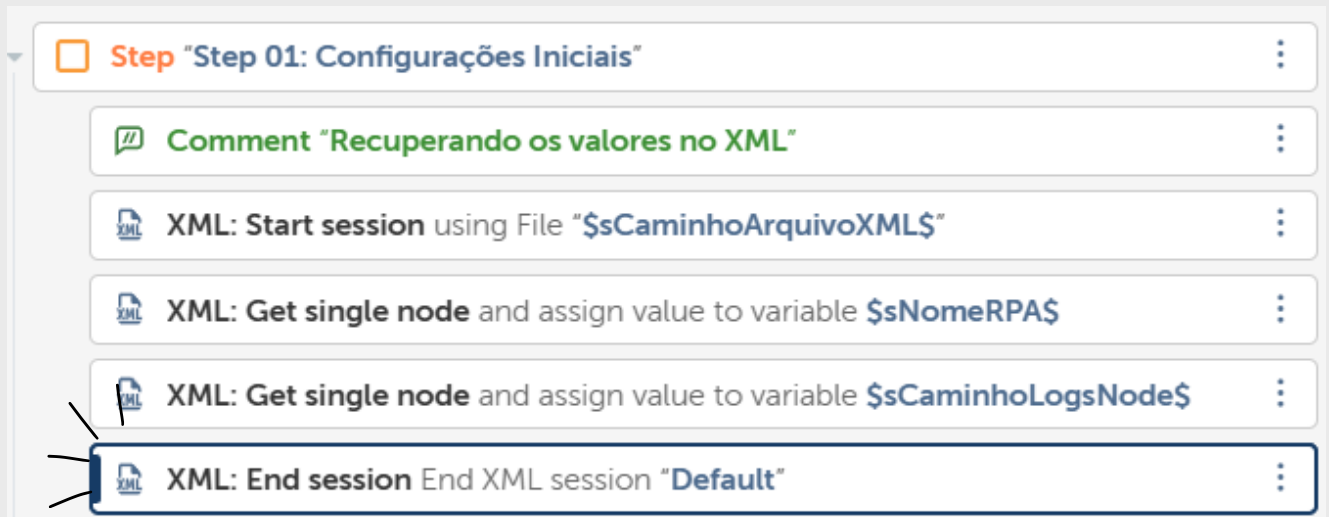
Attribute (optional)

”

Assign the output to variable

” sCaminhoLogsNode

5. Por último, a ação **End Session** finaliza a sessão atual, definida no início como **Default**.



Step "Step 01: Configurações Iniciais"

- Comment "Recuperando os valores no XML"
- XML: Start session using File "\$sCaminhoArquivoXML\$"
- XML: Get single node and assign value to variable \$sNomeRPA\$
- XML: Get single node and assign value to variable \$sCaminhoLogsNode\$
- XML: End session End XML session "Default"**

XML: End session

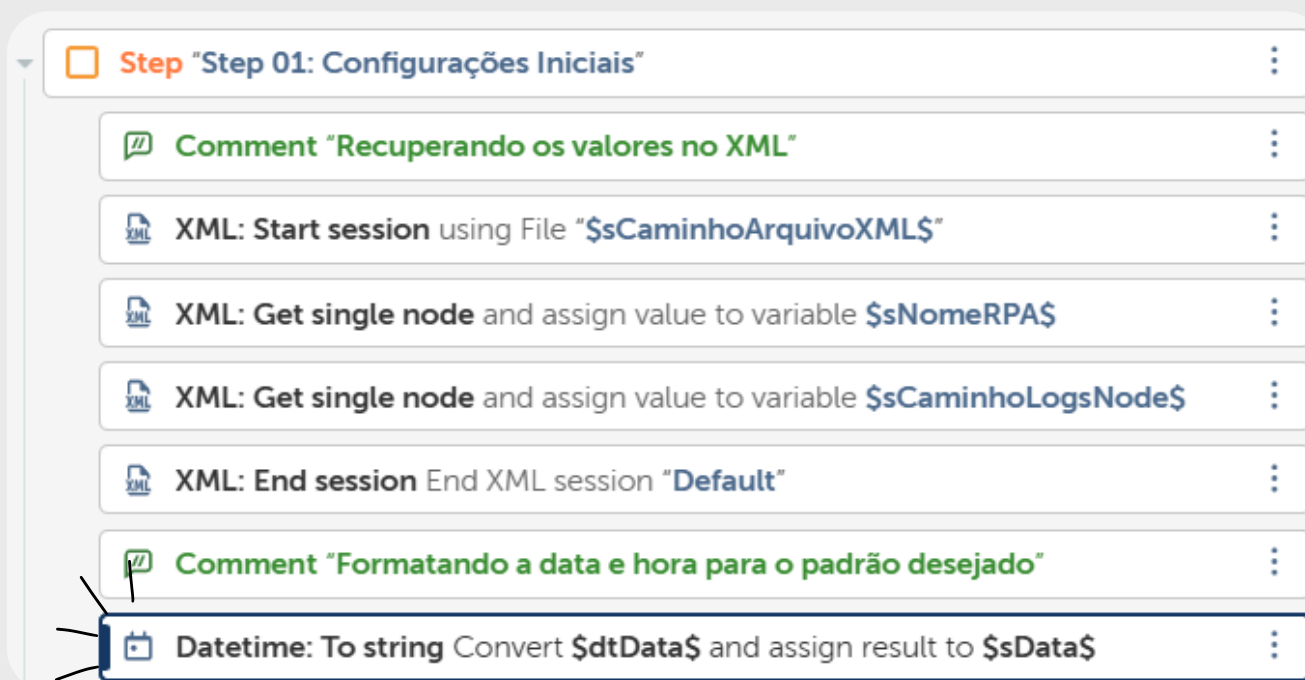
Closes an XML session

Required bot agent version: 20.11 or above

Session name

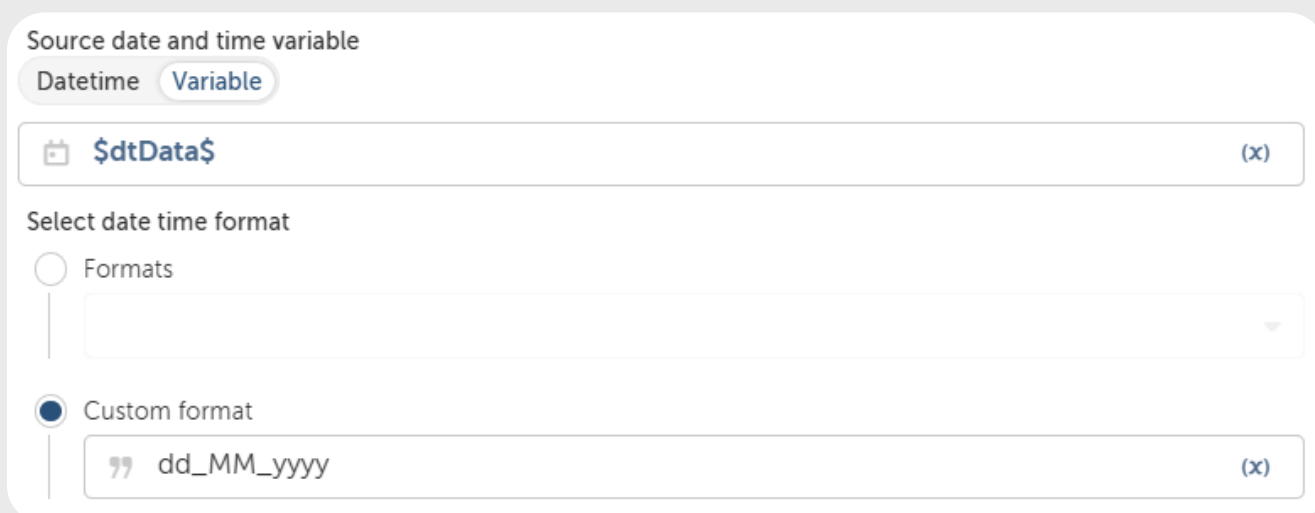
” Default

6. Como o formato do nosso log de exemplo será **NomeRPA_Data**, vou converter o **datetime** atual para apenas **data** no formato **dd_MM_yyyy**.



Step "Step 01: Configurações Iniciais"

- Comment "Recuperando os valores no XML"
- XML: Start session using File "\$sCaminhoArquivoXML\$"
- XML: Get single node and assign value to variable \$sNomeRPA\$
- XML: Get single node and assign value to variable \$sCaminhoLogsNode\$
- XML: End session End XML session "Default"
- Comment "Formatando a data e hora para o padrão desejado"
- Datetime: To string** Convert \$dtData\$ and assign result to \$sData\$



Source date and time variable

Datetime Variable

\$dtData\$ (x)

Select date time format

☐ Formats

☒ Custom format

dd_MM_yyyy (x)

Para o outro datetime, que será usado no log, vamos formatar a **data e hora** no padrão **dd/MM/yyyy HH:mm:ss**.

Step "Step 01: Configurações Iniciais"

- Comment "Recuperando os valores no XML"**
- XML: Start session** using File "\$sCaminhoArquivoXML\$"
- XML: Get single node** and assign value to variable \$sNomeRPA\$
- XML: Get single node** and assign value to variable \$sCaminhoLogsNode\$
- XML: End session** End XML session "Default"
- Comment "Formatando a data e hora para o padrão desejado"**
- Datetime: To string** Convert \$dtData\$ and assign result to \$sData\$
- Datetime: To string** Convert \$dtData\$ and assign result to \$sDataHora\$

Source date and time variable

Datetime **Variable**

\$dtData\$

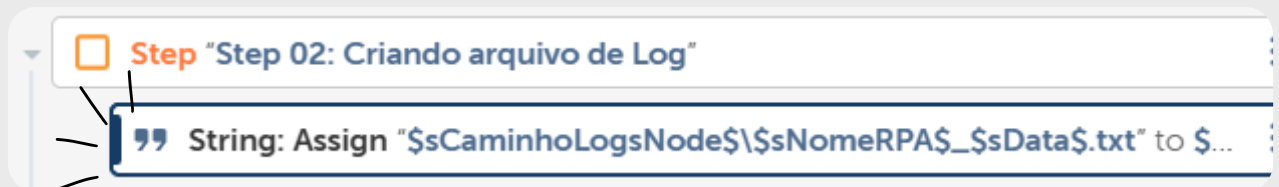
Select date time format

☐ Formats

☒ Custom format

dd/MM/yyyy HH:mm:ss

7. Na Step 02, vamos montar o **caminho completo** para o nosso arquivo de log, utilizando a ação **String: Assign**, no formato TXT.



String: Assign

Assigns or concatenates strings.

Required bot agent version: 20.11 or above

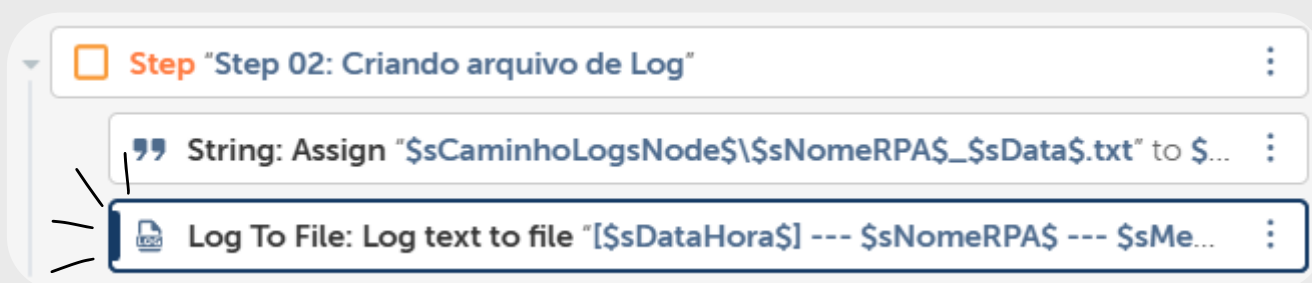
Select the source string variable(s)/ value (optional)

” \$sCaminhoLogsNode\$\sNomeRPA\$_sData\$.txt

Select the destination string variable

” sArquivoLog

8. Por último, na ação **Log to File**, vamos informar o caminho do arquivo de log para gravarmos os registros. Além disso, devemos manter habilitada a opção **Append to Existing Log File** para adicionar novos registros dentro do mesmo arquivo.



Log To File: Log text to file

Creates a log file with data about events that occur when a TaskBot runs
Required bot agent version: 20.11 or above

File path

” \$sArquivoLog\$

Enter text to log

” [\$sDataHora\$] --- \$sNomeRPA\$ --- \$sMensagemRPA\$

☐ Append timestamp

When logging

☒ Append to existing log file

☐ Overwrite existing log file

Encoding

UTF8

EXEMPLO DE RPA

Agora, vamos usar um RPA de teste que apenas abre uma planilha Excel para verificar se o task bot educaLogs **cria o arquivo de log** corretamente.

No RPA, utilizaremos a ação **String: Assign** para definir as mensagens que serão registradas no log.


” String: Assign "Abrindo planilha Excel" to \$\$MensagemRetorno\$

” String: Assign "Planilha aberta com êxito!" to \$\$MensagemRetorno\$

Na ação **Task Bot Run**, vamos clicar em **Choose** para selecionarmos o task bot **educaLogs**.

Runs the selected task bot.

Required bot agent version: 20.11 or above

 Task Bot to run

Current Task Bot **Control Room file** Variable

Required **Choose...**

Em seguida, clicaremos em **Browse** e vamos selecionar o nosso task bot.

Find a Task Bot



Search **Browse**

Folders

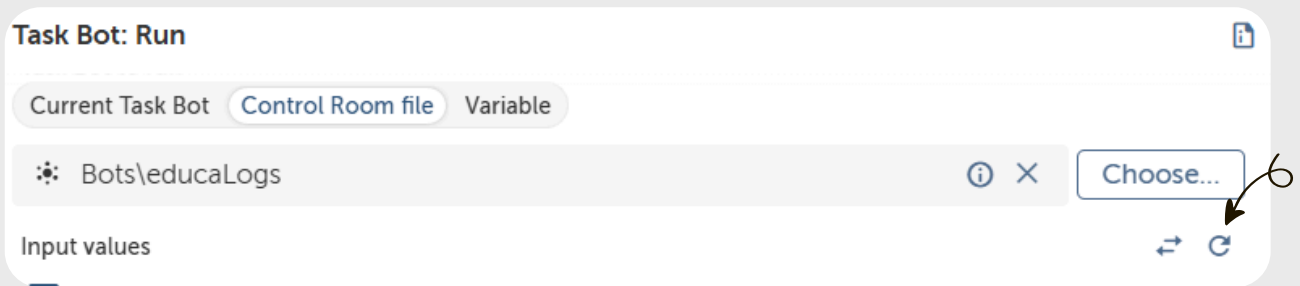
- ▼ Bots
 - > Document Workspace P...
 - Sample bots
 - Bot Store

Name Search

Task Bots (5 of 8) (1 checked)

T. 11	Name 12	Status	Size	Last modified	Mo
	 educaLogs	New	5.84 KB	34 minutes ago	MC

Na ação **Task Bot Run**, vamos clicar sobre a **seta** para exibir todas as variáveis de entrada que receberão valores.



Para cada variável, atribuímos os seguintes valores:

- **Datetime (dtData):** Valor de data e hora não formatado.
 - **Caminho XML (sCaminhoArquivoXML):** Localização do arquivo de configuração.
 - **Nó de caminho do log (sCaminhoLogsNode):** Caminho do nó no XML para a pasta de logs.
 - **Nó de nome do RPA (sNomeRPANode):** Caminho do nó no XML para o nome da automação.
 - **Mensagem de Retorno (sMensagemRPA):** Texto a ser gravado no log.
-

Ação Task Bot Run preenchida com todas as variáveis mencionadas anteriormente.

Task Bot: Run

Task Bot: Run

Required bot agent version: 20.11 or above

Task Bot to run

Current Task BotControl Room fileVariable

Bots\educaLogs

Choose...

Input values

☒ Datetime "dtData"

DatetimeVariable

\$System:Date\$

(x)

☒ String "sCaminhoArquivoXML"

\$sCaminhoArquivoXML\$

(x)

☒ String "sCaminhoLogsNode"

//EducaConfig/CaminhoLog

(x)

☒ String "sMensagemRPA"

\$sMensagemRetorno\$

(x)

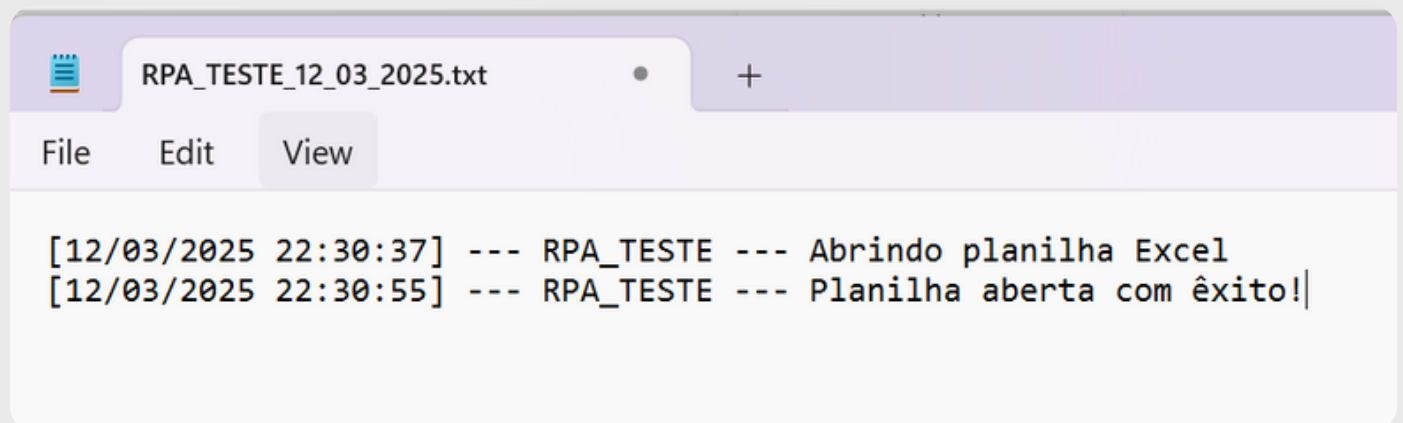
☒ String "sNomeRPANode"

//EducaConfig/NomeRPA

(x)

Chegamos ao fim!

Aqui está o nosso log gerado após finalizarmos a construção de todo o código.



The image shows a screenshot of a code editor window. The title bar at the top is light purple and contains a file icon, the filename 'RPA_TESTE_12_03_2025.txt', a close button, and a plus sign for additional tabs. Below the title bar is a menu bar with three items: 'File', 'Edit', and 'View'. The main area of the editor is white and displays a log file with two lines of text in a monospaced font. The first line is '[12/03/2025 22:30:37] --- RPA_TESTE --- Abrindo planilha Excel' and the second line is '[12/03/2025 22:30:55] --- RPA_TESTE --- Planilha aberta com êxito!'.

```
[12/03/2025 22:30:37] --- RPA_TESTE --- Abrindo planilha Excel  
[12/03/2025 22:30:55] --- RPA_TESTE --- Planilha aberta com êxito!
```

GOSTOU DO PROJETO?

Apoie curtindo o repositório no
Github 🌟 e este post aqui no
LinkedIn! Compartilhe com a sua
rede!



@marthabea



Martha Beatriz