



# MANUAL DE INSTALAÇÃO E UTILIZAÇÃO

# ZURI – PYTHON

Este é um manual de instalação e utilização da linguagem **python** no Zuri.

Aqui você irá encontrar o passo a passo para a instalação, ativação bem como dicas e boas práticas para a utilização do mesmo.

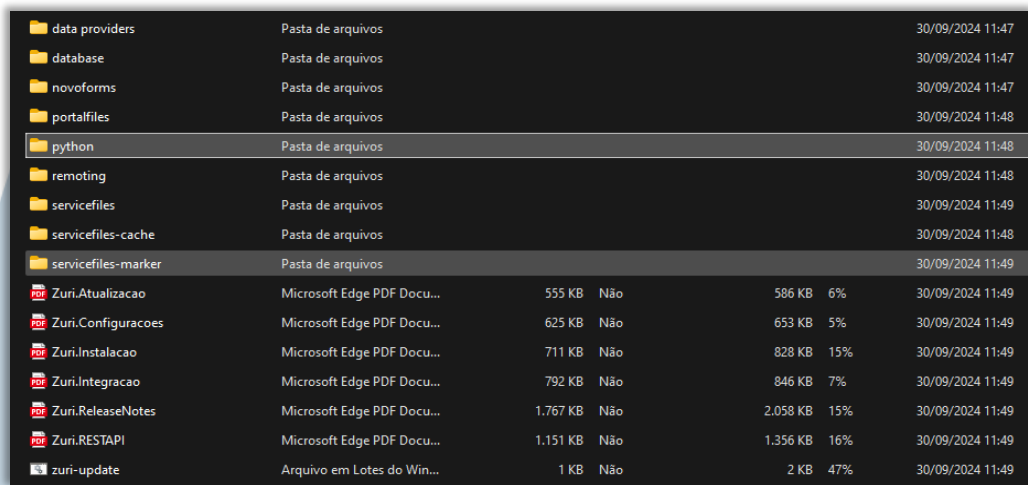


## SUMÁRIO

1. Instalação do python no Zuri
2. Instalação de pacotes no IronPython
3. Utilização no módulo de processo
4. Observações importantes

# 1. INSTALAÇÃO DO PYTHON NO ZURI

O arquivo compactado da instalação fica na pasta raiz do pacote, dentro da pasta python (Figura 1).



data providers	Pasta de arquivos	30/09/2024 11:47
database	Pasta de arquivos	30/09/2024 11:47
novofoms	Pasta de arquivos	30/09/2024 11:47
portalfiles	Pasta de arquivos	30/09/2024 11:48
python	Pasta de arquivos	30/09/2024 11:48
remoting	Pasta de arquivos	30/09/2024 11:48
servicefiles	Pasta de arquivos	30/09/2024 11:49
servicefiles-cache	Pasta de arquivos	30/09/2024 11:48
servicefiles-marker	Pasta de arquivos	30/09/2024 11:49
Zuri.Atualizacao	Microsoft Edge PDF Docu...	555 KB Não 586 KB 6% 30/09/2024 11:49
Zuri.Configuracoes	Microsoft Edge PDF Docu...	625 KB Não 653 KB 5% 30/09/2024 11:49
Zuri.Instalacao	Microsoft Edge PDF Docu...	711 KB Não 828 KB 15% 30/09/2024 11:49
Zuri.Integracao	Microsoft Edge PDF Docu...	792 KB Não 846 KB 7% 30/09/2024 11:49
Zuri.ReleaseNotes	Microsoft Edge PDF Docu...	1.767 KB Não 2.058 KB 15% 30/09/2024 11:49
Zuri.RESTAPI	Microsoft Edge PDF Docu...	1.151 KB Não 1.356 KB 16% 30/09/2024 11:49
zuri-update	Arquivo em Lotes do Win...	1 KB Não 2 KB 47% 30/09/2024 11:49

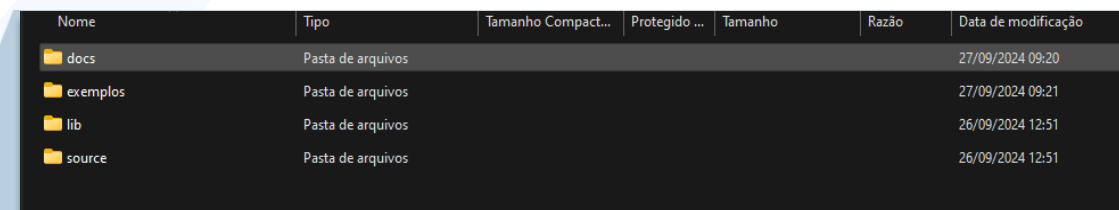
Figura 1 – Estrutura de pastas do pacote do Zuri

Dentro do arquivo compactado existem 4 pastas (Figura 2).

## Exemplos:

- **Exemplos de Docs** – Documentação de como utilizar o Zuri com o Python
- **Chamadas e retornos** dos arquivos do Python

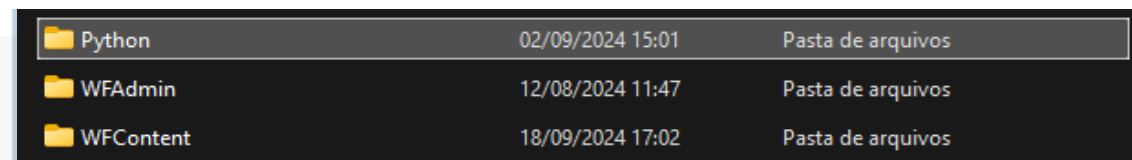
- **Libs** – Essa é a biblioteca padrão do Python (StdLib) para que possa ser executdo pelo IronPython no Zuri
- **Source** – A pasta em que deve ficar os arquivos a serem executados pelo Zuri.



Nome	Tipo	Tamanho Compact...	Protegido ...	Tamanho	Razão	Data de modificação
docs	Pasta de arquivos					27/09/2024 09:20
exemplos	Pasta de arquivos					27/09/2024 09:21
lib	Pasta de arquivos					26/09/2024 12:51
source	Pasta de arquivos					26/09/2024 12:51

Figura 2 – Estrutura de pastas do arquivo compactado

As pastas “**Python/lib** e **Python/source**” devem por padrão ficar na raiz da pasta Site (Figura 3).



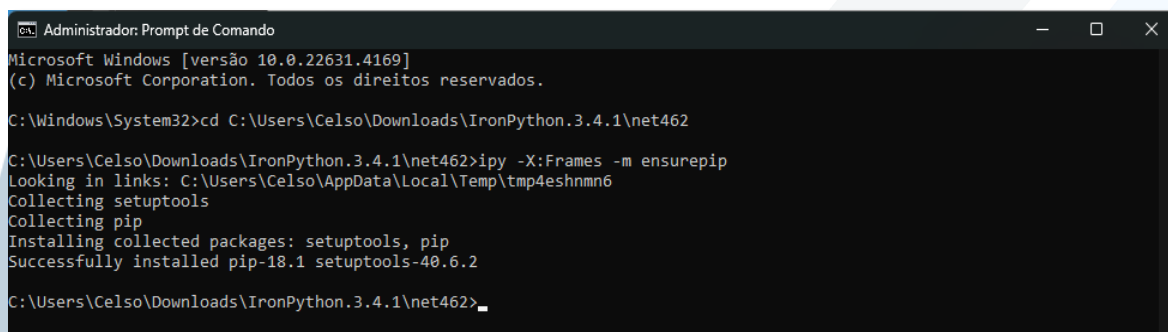
Python	02/09/2024 15:01	Pasta de arquivos
WFAAdmin	12/08/2024 11:47	Pasta de arquivos
WFContent	18/09/2024 17:02	Pasta de arquivos

Figura 3 – Pasta Python na raiz da pasta Site do Zuri

## 2. INSTALAÇÃO DE PACOTES DO IRON PYTHON

O Zuri utiliza uma versão embedded do IronPython, sendo assim a instalação dos pacotes precisa ser feita de forma manual.

1. Baixar a ver 3.4 no site do IronPython (<https://ironpython.net>)
2. Executar o comando no cmd como admin **"ipy -X:Frames -m ensurepip"** para habilitar o pip. (conforme Figura 4.)
3. Executar o comando no cmd como admin **"ipy -X:Frames -m pip install Package"** para instalar a dependência desejada



```
Administrador: Prompt de Comando
Microsoft Windows [versão 10.0.22631.4169]
(c) Microsoft Corporation. Todos os direitos reservados.

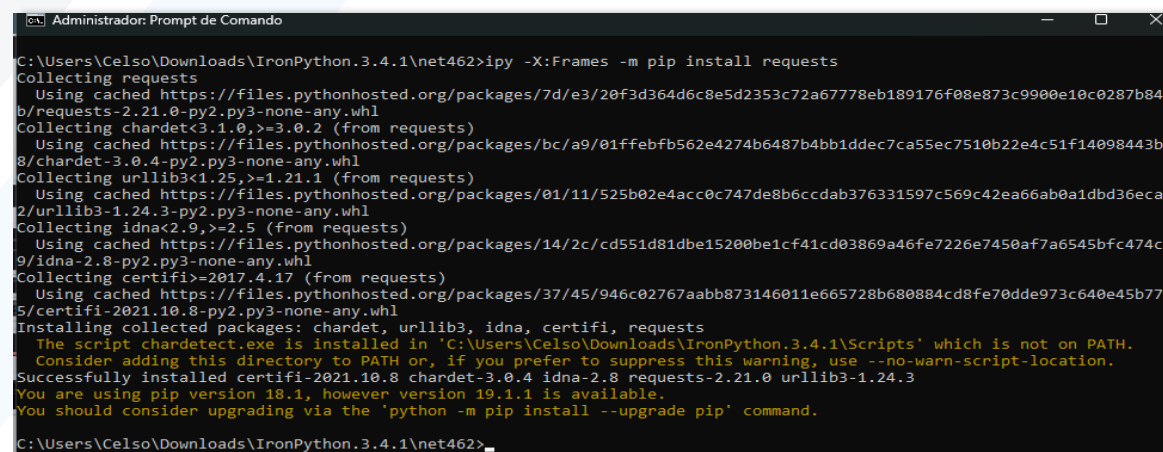
C:\Windows\System32>cd C:\Users\Celso\Downloads\IronPython.3.4.1\net462

C:\Users\Celso\Downloads\IronPython.3.4.1\net462>ipy -X:Frames -m ensurepip
Looking in links: C:\Users\Celso\AppData\Local\Temp\tmp4eshnmn6
Collecting setuptools
Collecting pip
Installing collected packages: setuptools, pip
Successfully installed pip-18.1 setuptools-40.6.2

C:\Users\Celso\Downloads\IronPython.3.4.1\net462>
```

Figura 4 – Habilitando o pip

4. Após a instalação ter sido realizada, copiar a pasta ***IronPython.3.4.1\lib\site-packages*** para a pasta do Zuri ***Site/Python/lib/site-package***



```
Administrador: Prompt de Comando

C:\Users\Celso\Downloads\IronPython.3.4.1\net462>ipy -X:Frames -m pip install requests
Collecting requests
  Using cached https://files.pythonhosted.org/packages/7d/e3/20f3d364d6c8e5d2353c72a67778eb189176f08e873c9900e10c0287b84b/requests-2.21.0-py2.py3-none-any.whl
Collecting chardet<3.1.0,>=3.0.2 (from requests)
  Using cached https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl
Collecting urllib3<1.25,>=1.21.1 (from requests)
  Using cached https://files.pythonhosted.org/packages/01/11/525b02e4acc0c747de8b6ccdb376331597c569c42ea66ab0a1dbd36eca2/urllib3-1.24.3-py2.py3-none-any.whl
Collecting idna<2.9,>=2.5 (from requests)
  Using cached https://files.pythonhosted.org/packages/14/2c/cd551d81dbe15200be1cf41cd03869a46fe7226e7450af7a6545bfc474c9/idna-2.8-py2.py3-none-any.whl
Collecting certifi<2017.4.17 (from requests)
  Using cached https://files.pythonhosted.org/packages/37/45/946c02767aabb873146011e665728b680884cd8fe70dde973c640e45b775/certifi-2021.10.8-py2.py3-none-any.whl
Installing collected packages: chardet, urllib3, idna, certifi, requests
The script chardet.exe is installed in 'C:\Users\Celso\Downloads\IronPython.3.4.1\Scripts' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed certifi-2021.10.8 chardet-3.0.4 idna-2.8 requests-2.21.0 urllib3-1.24.3
You are using pip version 18.1, however version 19.1.1 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

C:\Users\Celso\Downloads\IronPython.3.4.1\net462>
```

Figura 5– exemplos de instalação do pacote requests

**Importante!:** O IronPython é incompatível com bibliotecas do Python escritas em C como o NumPy, por exemplo.

### 3. UTILIZAÇÃO NO MÓDULO DE PROCESSO

O módulo de código Python fica disponível na aba Integração do desenho de processo. (conforme figura 6)



Figura 6 – Módulo de código Python no desenho de processo

O próximo passo é realizar o upload do arquivo **python** para a pasta **source** através dos seguintes passos:

1. Selecione o botão **“Escolher arquivo”** e selecione o arquivo pyton.
2. Após o arquivo estar selecionado, clique no botão de **“Upload”**.
3. Após o upload, o mesmo ficará disponível na combo **“Source”**.

*caso precise atualizar um arquivo é só repetir o processo de upload conforme figura 7.*

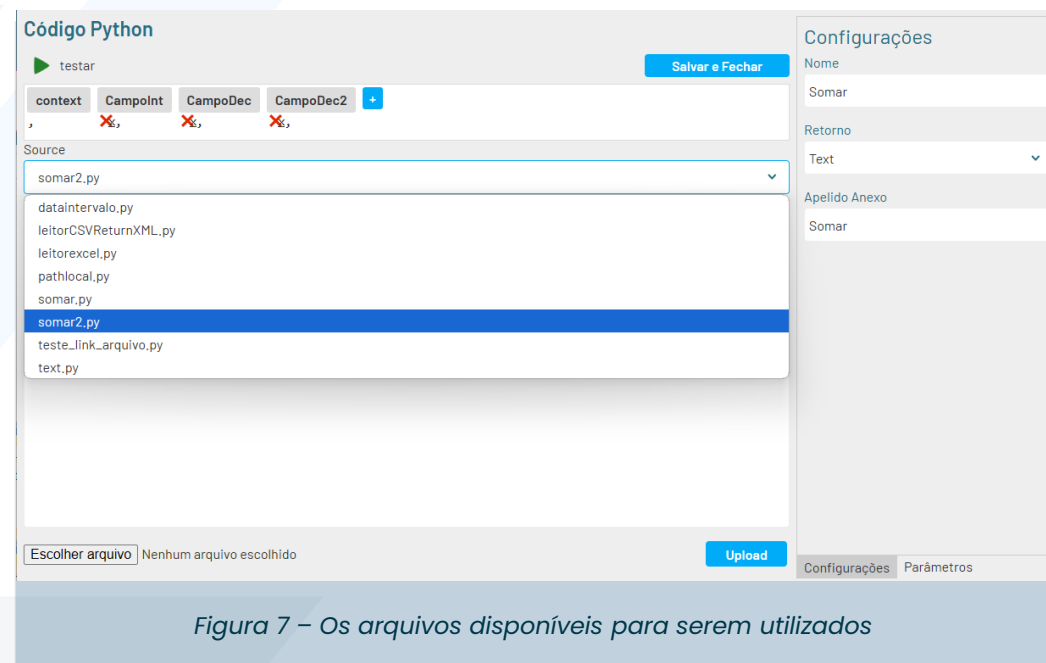


Figura 7 – Os arquivos disponíveis para serem utilizados

### 3. UTILIZAÇÃO NO MÓDULO DE PROCESSO

4. Todos os retornos devem vir como um texto do python para o Zuri, respeitando as opções da combo **"Retorno"**. (conforme figura 8)

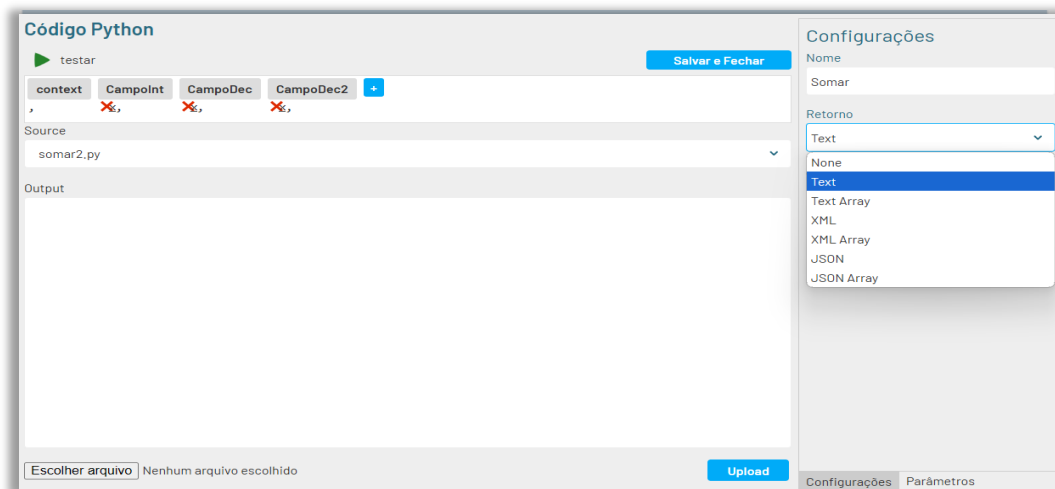


Figura 8 – Tipo de retorno disponíveis.

5. Os parâmetros são configurados como o módulo de código original do Zuri, tendo seu **Type**, **Value** e **Test Value**. (conforme figura 9)

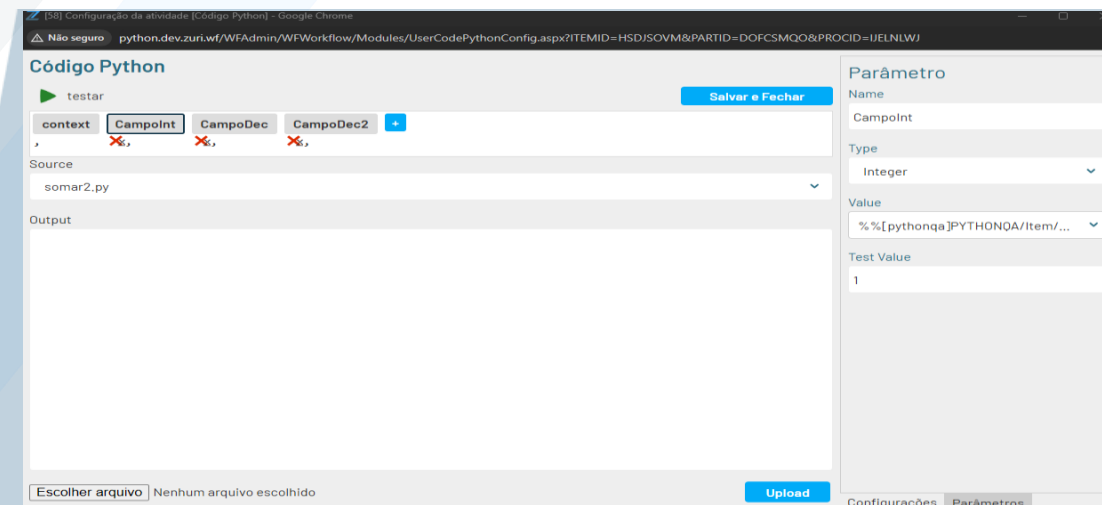


Figura 9 – Aba de parâmetros.

Qualquer dúvida consultar a pasta de exemplos, dentro do pacote.

### 3. UTILIZAÇÃO NO MÓDULO DE PROCESSO

6. Ao clicar **em testar** o valor de retorno do Python assim como os erros são exibidos na caixa de "Output. (conforme figura 10)

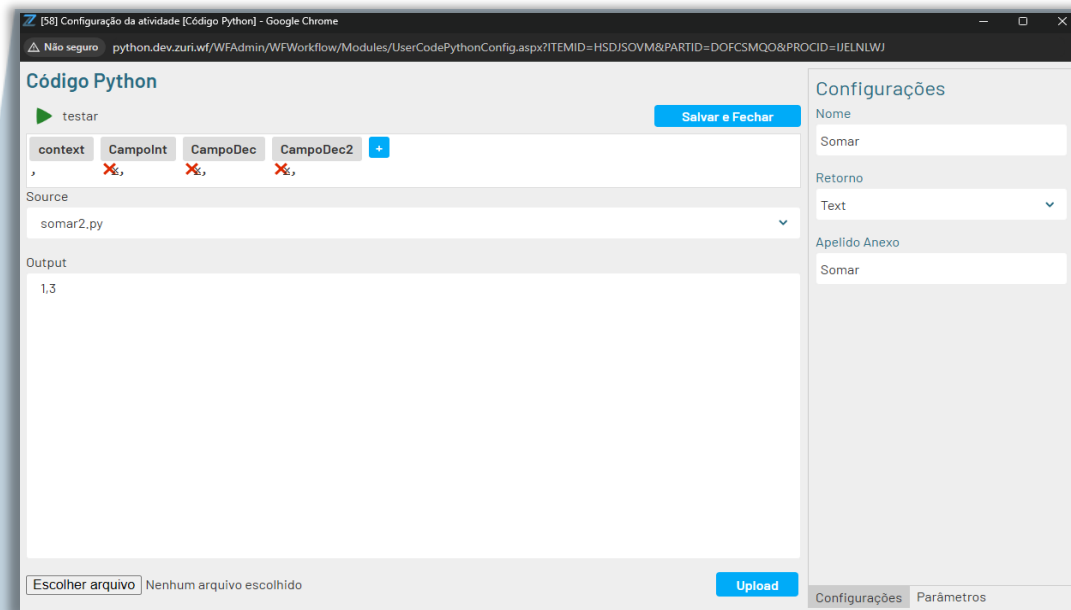


Figura 10 – Exemplo de retorno do arquivo somar2.py

7. Clicar em salvar e fechar para as variáveis de contexto estarem disponíveis para utilizar no processo. (conforme figura 11)

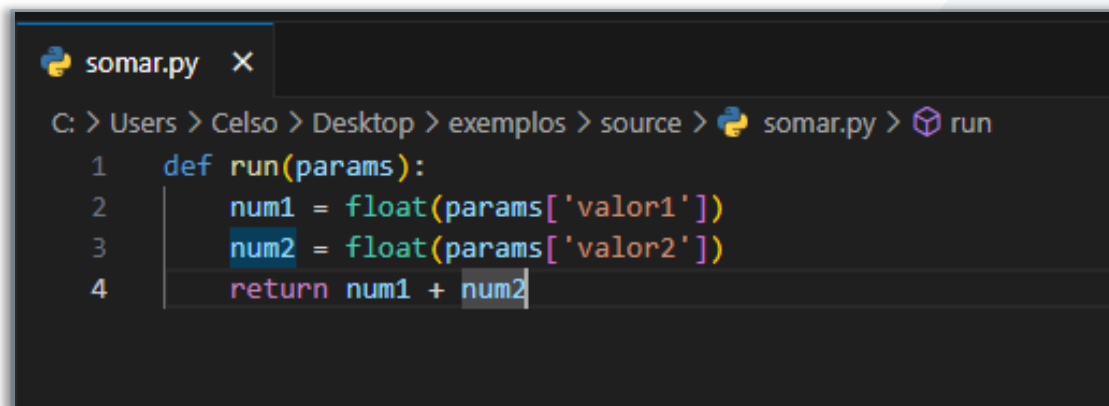


Figura 11 – Variáveis de contexto disponíveis para serem utilizados no desenho de processo.

### 3. UTILIZAÇÃO NO MÓDULO DE PROCESSO

Após a seleção de variáveis e a configuração, o Zuri vai fazer uma chamada da função **def run(params)**, ou seja irá passar um dicionário com o parâmetros informados.

**Essa função é obrigatória dentro do arquivo**, caso contrário o mesmo não será executado, conforme figura 12.



```
somar.py X
C: > Users > Celso > Desktop > exemplos > source > somar.py > run
1 def run(params):
2     num1 = float(params['valor1'])
3     num2 = float(params['valor2'])
4     return num1 + num2
```

Figura 12 – Exemplo de código python a ser chamado pelo Zuri.

### 4. OBSERVAÇÕES IMPORTANTES

- Quando for trabalhar como arquivos **CSV** certifique que o mesmo não está como o **encoding UTF8-BOM**, pois o mesmo dá erro na hora de transformar para o retorno desejado no Zuri.
- Os valores são enviados como texto para o python e os mesmos precisam ser tratados no python antes de sua utilização, no caso de **inteiro, float ou data**.
- Os objetos de data no python precisam ser enviados como texto para o Zuri pois o mesmo não consegue transformar os **objectos DeltaData e DeltaTime do python**.
- O Zuri utiliza a vírgula como separador de decimal, enquanto o python trabalha com o ponto por padrão. **Os dados devem ser tratados para que os mesmos possam ser utilizados de forma correta pelo python.**





**BOA UTILIZAÇÃO!**



Em caso de dúvidas, entre em contato:

**[suporte@gozuri.com](mailto:suporte@gozuri.com)**

