

# Cálculo de Torque

---

## REGISTRO DE MODELOS

Todas as informações dos modelos estudados ficam no arquivo *inputs.json*. Para configurá-lo, basta abrir o arquivo com qualquer editor de texto e informar os novos valores no seguinte formato:

```
"aileron202X": {  
  "area_asa": valor,  
  "fator_carga_maxima": valor,  
  "massa": valor,  
  "base": valor,  
  "altura": valor  
},  
"profundor202X": {  
  "maior_carga_emp_hor": valor,  
  "corda_emp_hor": valor,  
  "corda": valor,  
  "base": valor,  
  "altura": valor  
},  
"leme202X": {  
  "area_emp_vert": valor,  
  "area": valor,  
  "carga_rajada_emp_vert": valor,  
  "base": valor,  
  "altura": valor  
}  
}
```

Obs.: Atenção às vírgulas, aspas e chaves.

Para registrar um novo modelo não há necessidade de excluir algum outro que já esteja registrado. O programa é capaz de filtrar os dados necessários para o cálculo de cada modelo individualmente.

**Atenção:** O arquivo *inputs.json* deve estar na mesma pasta que o arquivo de comandos “*Cargas - TorqueS.py*”.

# CÁLCULOS E EXIBIÇÃO

## Funções de torque

Todos os cálculos estão dentro de funções específicas para cada superfície de comando. Dessa forma, o processo matemático já está automatizado para qualquer modelo que esteja dentro das formas geométricas já estudadas. Para obter os valores resultantes, basta usar os comandos

```
torque_aileron("aileron202X")  
  
torque_profundor("profundor202X")  
  
torque_leme("leme202X")
```

Obs.: Os comandos podem ser executados individualmente.

Perceba que o comando não exibe nenhum resultado na tela. Isso acontece porque as funções foram construídas para retornar uma lista de valores ao invés de exibi-los diretamente. Com essa estrutura, o programa será capaz de reaproveitar os resultados caso seja necessário em futuras implementações.

## Função de exibição

Visto que as funções de cálculo não exibem os resultados no terminal, foi criada uma função própria para isso, capaz de organizar as informações na tela de forma legível. Para executar a exibição, basta inserir o seguinte comando:

```
exibir(torque_aileron("aileron202X"))  
  
exibir(torque_profundor("profundor202X"))  
  
exibir(torque_leme("leme202X"))
```

Obs.: Uma função dentro de outra pode causar estranheza, mas é completamente possível.

Caso haja a necessidade, também é possível salvar os resultados em uma variável do tipo biblioteca e posteriormente usá-la como argumento da função `exibir()`. Porém, essa seria uma aplicação um pouco mais avançada e, por enquanto, desnecessária.

## ACESSO AO PROJETO E ATUALIZAÇÕES

Para acessar qualquer alteração feita pelo setor de desenvolvimento, é possível acessar [nosso repositório do Github](#). Por ser o ambiente de gestão de código da equipe, terá sempre o código disponível e atualizado.