

## Teste para Programador Pleno de RPA com Python

### Avisos antes de começar:

Crie um repositório no seu GitHub e, após finalizar, faça o upload dos códigos. Inclua instruções detalhadas no README.md sobre como rodar o projeto e suas dependências.

### Objetivo:

Automatizar o download e instalação de um executável Python, além de realizar algumas operações adicionais para demonstrar habilidades avançadas de automação e manipulação de dados.

### Requisitos:

#### 1. Automatização do Download

- Criar uma automação para buscar no Google pela palavra "**baixar python**" usando o browser do Google Chrome.
- Com o retorno da consulta, acessar o link do [www.python.org](http://www.python.org).
- Procurar e encontrar a página de Downloads.
- Pesquisar pela versão 3.12.7 e baixar o executável.

#### 2. Instalação Automatizada

- Realizar a instalação automatizada da versão baixada.
- Validar se a instalação foi realizada com sucesso através de um script Python que confirme a versão instalada.
- Criar um Log no MongoDB com os logs de Baixa do executável, execução da instalação e verificação da versão instalada.

#### 3. Automatização de Tarefas Extras

- Criar uma automação que extraia dados de uma página web e salve esses dados em um arquivo **CSV**.
- Implementar um script que manipule os dados extraídos: faça filtros, agregações e gere um relatório resumido em **PDF** utilizando alguma biblioteca Python.

Link sugerido para capturas dos dados <https://books.toscrape.com>

### O que será avaliado e valorizamos:

- **Documentação Completa:** Instruções claras, detalhadas e bem organizadas.
- **Código Limpo e Estruturado:** Uso de boas práticas, PEP8, organização em funções ou classes.
- **Lógica e Solução dos Problemas:** Criatividade e eficiência nas soluções.
- **Uso de Ferramentas:**
  - Python
  - Selenium para automação de navegador.
  - Git para versionamento.
  - Pandas para manipulação de dados.
  - Biblioteca para geração de PDF (ex: ReportLab ou FPDF).

**Diferenciais (Opcional):**

- Utilização de testes automatizados (ex: **pytest**).
- Uso de **Docker** para criar um ambiente isolado de execução.