

## ESPECIFICAÇÃO DO PROCESSO

Área: Recursos Humanos

Processo: Extração de vagas e envio por e-mail

Data: 28/01/2022

Histórico de alterações:

Data	Versão	Descrição	Autor
28/01/2022	1.0	Processo de automatização extração de vagas	Igor Damasceno

### Descrição Geral

#### Objetivo:

Automatizar o fluxo de extração de vagas do portal da CadMus, gerar um relatório e encaminhar por e-mail para área responsável.

Detalhamento do Processo:

#### Macro processo:

- 1- Acessar a página de vagas da CadMus;
- 2- Extrair as vagas existentes;
- 3- Gerar um relatório Excel;
- 4- Encaminhar por e-mail para área responsável analisar;

#### Detalhes do Processo:

- Passo-a-Passo para a **Extração de vagas e envio:**
  - Acessar a página de vagas da Cadmus
  - Aguardar os elementos exibirem
  - Obter as oportunidades armazenadas em LocalStorage (Json)
  - Tratar os dados obtidos e retorná-los
  - Transformar os dados retornados em um DataFrame (pandas)
  - Gerar um excel a partir deste Dataframe (Pandas)
  - Disparar e-mail SMTP para área responsável com anexo

## ESPECIFICAÇÃO DO PROCESSO

Método responsável por abrir a página da CadMus e aguardar os elementos existirem

```
1 def OPEN_PAGE()->webdriver:
2     driver = webdriver.Chrome()
3     driver.get("https://cadmus.com.br/vagas-tecnologia/")
4
5     #WAITING FOR THE PAGE LOAD ELEMENTS
6     wdw = WebDriverWait(driver, 30)
7     locator = (By.CSS_SELECTOR, '#pfolio')
8     wdw.until(wdw.until(ExpectedConditions.presence_of_element_located(locator)))
9     return driver
10
```

Método responsável por pegar as oportunidades armazenadas no LOCALSTORAGE, tratá-las e retornar JSON.

```
def GET_OPORTUNITIES(driver:webdriver)->List:
    storage = LocalStorage(driver) #LOCAL STORAGE
    STORAGE_DETAILS = json.loads(storage.get("vagas")) # GET ALL OPORTUNITIES STORE
    # GET ONLY NECESSARY FIELDS FROM EACH OPORTUNITY
    DATA = [
        {
            "Oportunity": box['name'],
            "Locale": box['cidade_Regi_o_c'],
            "Detail": box['descricao_da_vaga_c'].replace("<br>", "")
        }
        for box in STORAGE_DETAILS
    ]
    return DATA
```

Método responsável por transformar JSON em um DATAFRAME e gerar excel a partir dele

```
def EXPORT_TO_EXCEL(DATA:List):
    DF = pd.DataFrame(DATA)
    DF.to_excel("Output/RelatorioVagas.xlsx",index=False)
```

Método responsável por enviar e-mail com anexo para área responsável.

Neste método são definidos: caminho do excel gerado, assunto, corpo html do e-mail.

## ESPECIFICAÇÃO DO PROCESSO

```
def SEND_EMAIL_TO_AREA(driver:webdriver):  
    # SUBJECT FROM THE EMAIL  
    Subject = "Relatório de Vagas Disponíveis CadMus"  
    SetSubject(Subject)  
  
    #BODY HTML  
    Body = "<h2>Relatório de vagas disponíveis Cadmus</h2><p>Segue Excel com vagas disponíveis do portal da CadMus para análise<p>"  
    SetBody(Body)  
  
    # WHO MUST RECIEVE E-MAIL  
    To = "igortech.pb@gmail.com"  
    SetTo(To)  
  
    #PATH FROM EXCEL FILE GENERATE  
    Excel = "Output/RelatorioVagas.xlsx"  
    SetPdf(Excel)  
  
    #SENDING E-MAIL WITH THE  
    Send()  
  
    #WHEN COMPLETED CLOSES  
    driver.quit()
```

### Requisitos

() Python instalado na máquina e instalação do requirements.txt

()Servidor de e-mail

**Declaro este documento de requisitos em acordo com escopo do projeto.**

Aprovação:

Aprovador por:	Assinatura	Data