Coleta de Dados 1

Processamento dos dados 4

Atividade da Prática Integradora 2 6

Considerações Finais 7

Código de pré-processamento dos dados do INMET 8

Código com o processamento final 13

Coleta de Dados

Site: https://tempo.inmet.gov.br
Caminho: Produto -> Tabela de Dados das Estações (Imagem A1)
Parâmetros utilizados (Imagem A2):

- Estado: São Paulo
- Estação: SAO PAULO - INTERLAGOS (A771)
- Data Início: Primeiro dia do mês (para cada mês)
- Data Fim: Último dia do mês (para cada mês)
- Ano: 2021
- Meses: Jan a Dez (amostra parcial coletada na Imagem A3)

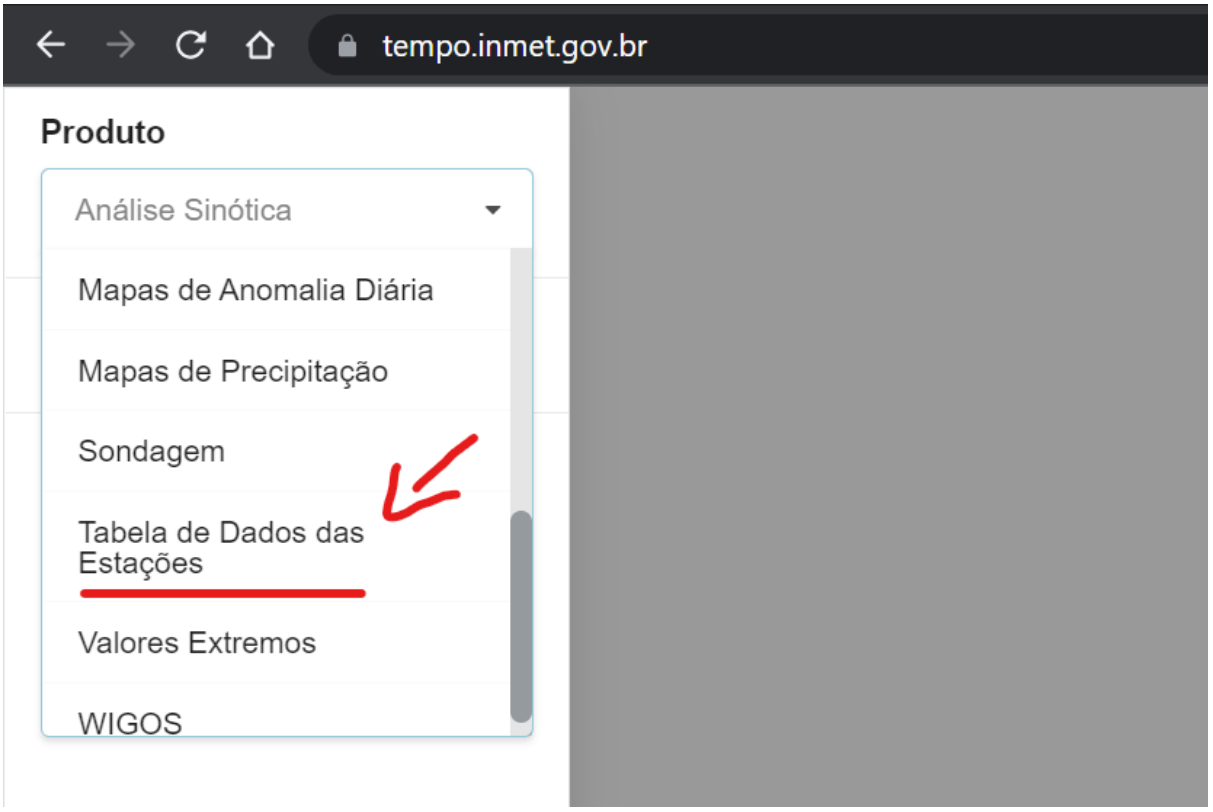


Imagem A1: Início da coleta de dados do site do Instituto Nacional de Meteorologia (INMET).

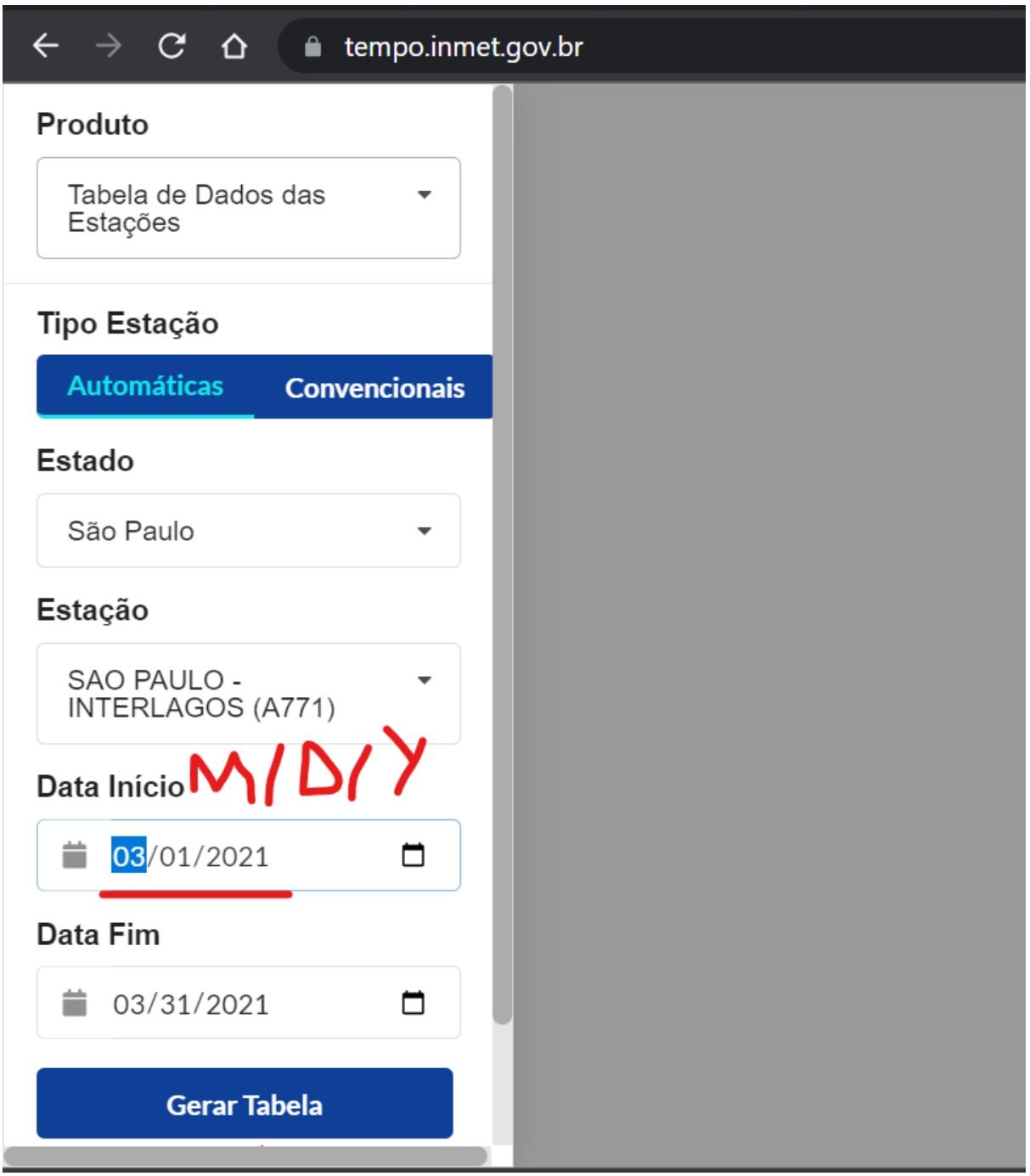
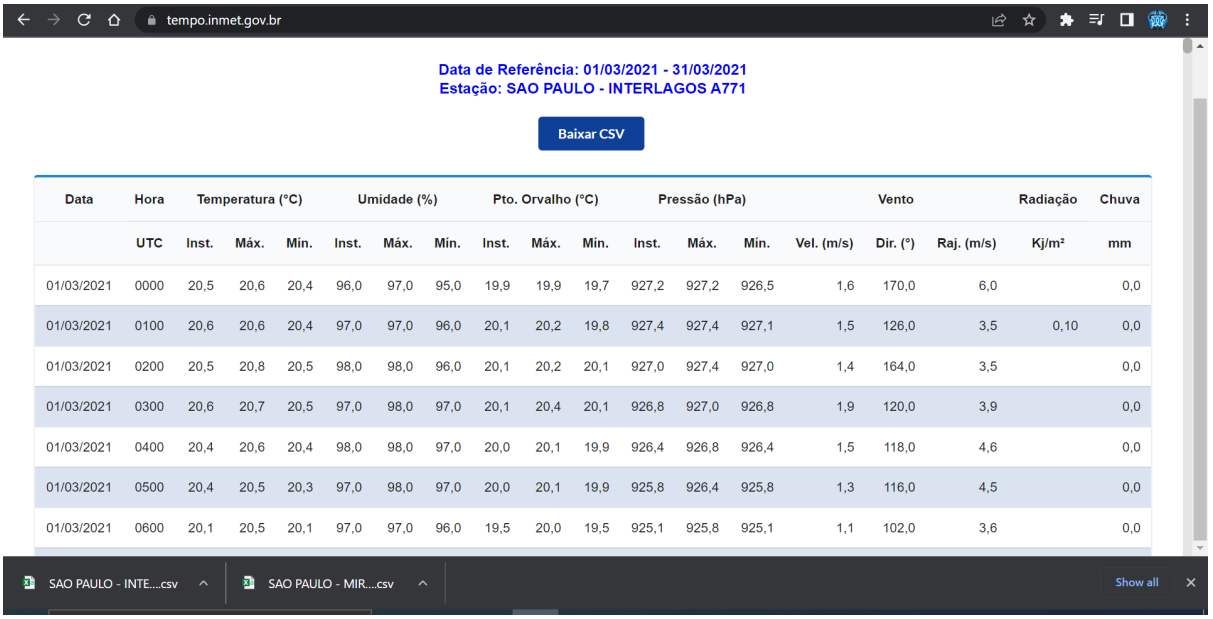


Imagem A2: Parâmetros utilizados na coleta de dados do site do Instituto Nacional de Meteorologia (INMET).



tempo.inmet.gov.br

Data de Referência: 01/03/2021 - 31/03/2021
Estação: SAO PAULO - INTERLAGOS A771

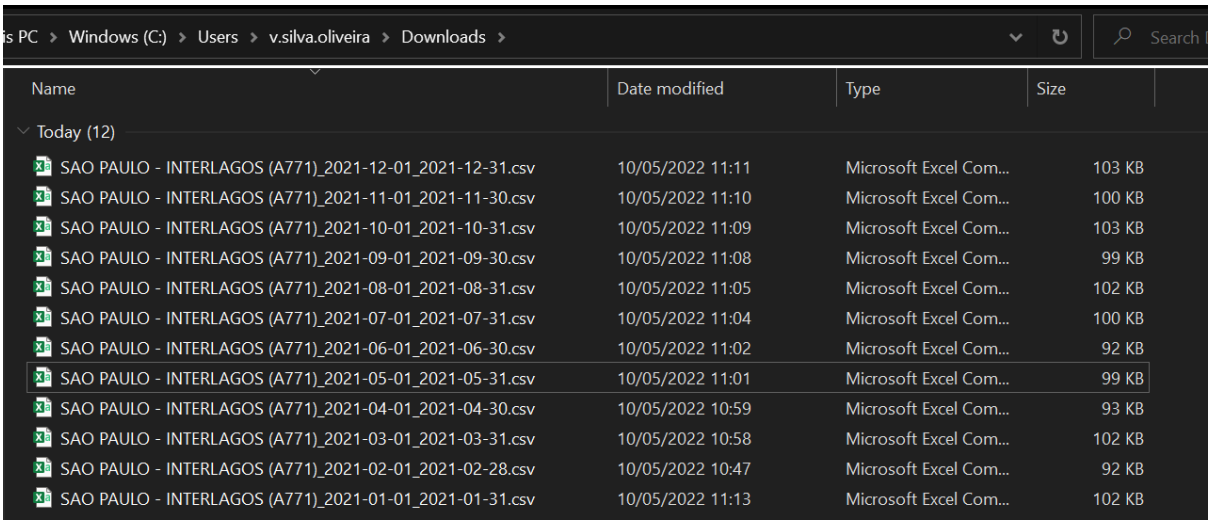
Baixar CSV

Data	Hora	Temperatura (°C)			Umidade (%)			Pto. Orvalho (°C)			Pressão (hPa)			Vento			Radiação	Chuva
	UTC	Inst.	Máx.	Mín.	Inst.	Máx.	Mín.	Inst.	Máx.	Mín.	Inst.	Máx.	Mín.	Vel. (m/s)	Dir. (°)	Raj. (m/s)	Kj/m²	mm
01/03/2021	0000	20,5	20,6	20,4	96,0	97,0	95,0	19,9	19,9	19,7	927,2	927,2	926,5	1,6	170,0	6,0		0,0
01/03/2021	0100	20,6	20,6	20,4	97,0	97,0	96,0	20,1	20,2	19,8	927,4	927,4	927,1	1,5	126,0	3,5	0,10	0,0
01/03/2021	0200	20,5	20,8	20,5	98,0	98,0	96,0	20,1	20,2	20,1	927,0	927,4	927,0	1,4	164,0	3,5		0,0
01/03/2021	0300	20,6	20,7	20,5	97,0	98,0	97,0	20,1	20,4	20,1	926,8	927,0	926,8	1,9	120,0	3,9		0,0
01/03/2021	0400	20,4	20,6	20,4	98,0	98,0	97,0	20,0	20,1	19,9	926,4	926,8	926,4	1,5	118,0	4,6		0,0
01/03/2021	0500	20,4	20,5	20,3	97,0	98,0	97,0	20,0	20,1	19,9	925,8	926,4	925,8	1,3	116,0	4,5		0,0
01/03/2021	0600	20,1	20,5	20,1	97,0	97,0	96,0	19,5	20,0	19,5	925,1	925,8	925,1	1,1	102,0	3,6		0,0

SAO PAULO - INTE....csv
SAO PAULO - MIR....csv
Show all

Imagem A3: Amostra parcial dos dados coletados.

Na Imagem 4 podemos verificar os arquivos baixados e seus metadados.



is PC > Windows (C:) > Users > v.silva.oliveira > Downloads >

Name	Date modified	Type	Size
Today (12)			
SAO PAULO - INTERLAGOS (A771)_2021-12-01_2021-12-31.csv	10/05/2022 11:11	Microsoft Excel Com...	103 KB
SAO PAULO - INTERLAGOS (A771)_2021-11-01_2021-11-30.csv	10/05/2022 11:10	Microsoft Excel Com...	100 KB
SAO PAULO - INTERLAGOS (A771)_2021-10-01_2021-10-31.csv	10/05/2022 11:09	Microsoft Excel Com...	103 KB
SAO PAULO - INTERLAGOS (A771)_2021-09-01_2021-09-30.csv	10/05/2022 11:08	Microsoft Excel Com...	99 KB
SAO PAULO - INTERLAGOS (A771)_2021-08-01_2021-08-31.csv	10/05/2022 11:05	Microsoft Excel Com...	102 KB
SAO PAULO - INTERLAGOS (A771)_2021-07-01_2021-07-31.csv	10/05/2022 11:04	Microsoft Excel Com...	100 KB
SAO PAULO - INTERLAGOS (A771)_2021-06-01_2021-06-30.csv	10/05/2022 11:02	Microsoft Excel Com...	92 KB
SAO PAULO - INTERLAGOS (A771)_2021-05-01_2021-05-31.csv	10/05/2022 11:01	Microsoft Excel Com...	99 KB
SAO PAULO - INTERLAGOS (A771)_2021-04-01_2021-04-30.csv	10/05/2022 10:59	Microsoft Excel Com...	93 KB
SAO PAULO - INTERLAGOS (A771)_2021-03-01_2021-03-31.csv	10/05/2022 10:58	Microsoft Excel Com...	102 KB
SAO PAULO - INTERLAGOS (A771)_2021-02-01_2021-02-28.csv	10/05/2022 10:47	Microsoft Excel Com...	92 KB
SAO PAULO - INTERLAGOS (A771)_2021-01-01_2021-01-31.csv	10/05/2022 11:13	Microsoft Excel Com...	102 KB

Imagem A4: Arquivos baixados com metadados.

Processamento dos dados

O programa “processar_dados_inmet.js” tem como entrada os arquivos csv do diretório “dados_mensais_inmet” e uma lista em formato txt contendo o nome dos arquivos do inmet. Abaixo estão a lista dos arquivos csv do inmet utilizados, evidenciado na Imagem A5:

- SAO PAULO - INTERLAGOS (A771)_2021-01-01_2021-01-31.csv
- SAO PAULO - INTERLAGOS (A771)_2021-02-01_2021-02-28.csv
- SAO PAULO - INTERLAGOS (A771)_2021-03-01_2021-03-31.csv
- SAO PAULO - INTERLAGOS (A771)_2021-04-01_2021-04-30.csv
- SAO PAULO - INTERLAGOS (A771)_2021-05-01_2021-05-31.csv

- SAO PAULO - INTERLAGOS (A771)_2021-06-01_2021-06-30.csv
- SAO PAULO - INTERLAGOS (A771)_2021-07-01_2021-07-31.csv
- SAO PAULO - INTERLAGOS (A771)_2021-08-01_2021-08-31.csv
- SAO PAULO - INTERLAGOS (A771)_2021-09-01_2021-09-30.csv
- SAO PAULO - INTERLAGOS (A771)_2021-10-01_2021-10-31.csv
- SAO PAULO - INTERLAGOS (A771)_2021-11-01_2021-11-30.csv
- SAO PAULO - INTERLAGOS (A771)_2021-12-01_2021-12-31.csv

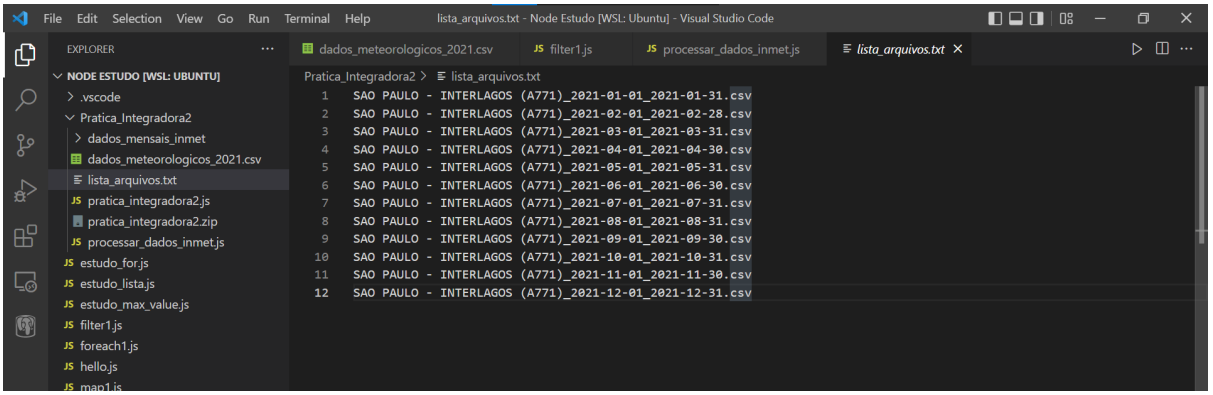


Imagem A5: Lista de arquivos separados por linha.

A saída do processamento anterior é um arquivo csv com 365 linhas, uma para cada dia do ano de 2021, contendo os seguintes dados por linha: data (ordem crescente), média diária das temperaturas máximas, média diária das temperaturas mínimas, índice pluviométrico (mm).
Importante ressaltar que o programa verifica se as linhas dos registros são nulas, e após fazer a média, mantém os dados com 1 casa decimal. A imagem A6 exibe o arquivo de saída resultante.

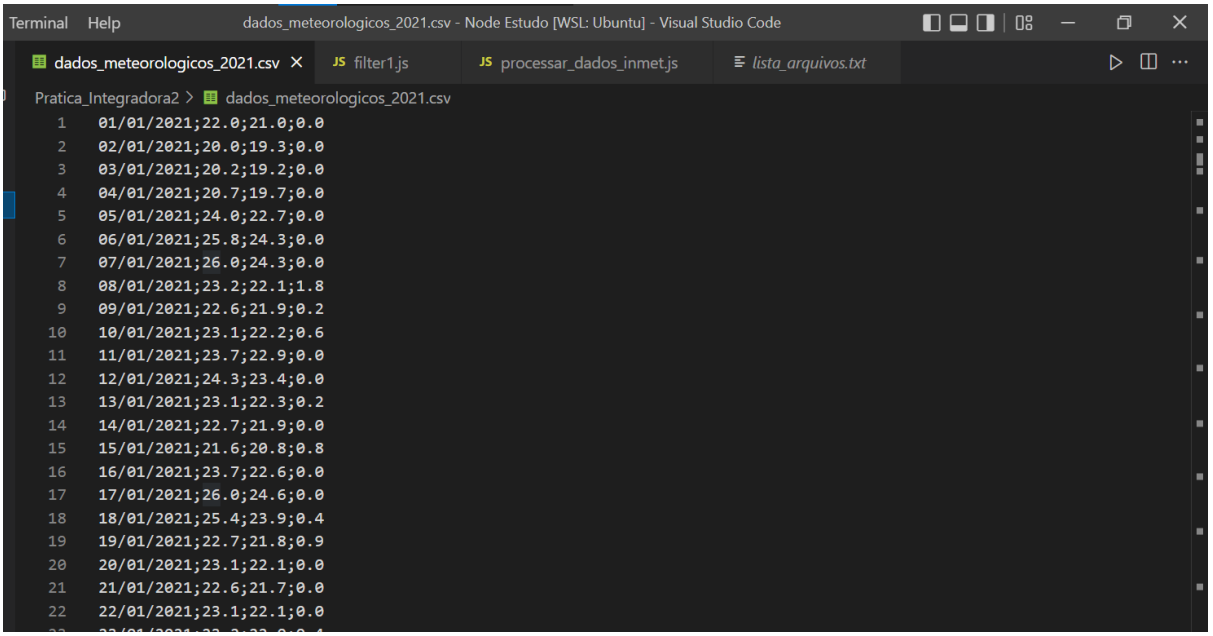


Imagem A6: Saída gerada pelo processamento anterior

Tendo uma lista com 365 temperaturas foi iniciada a atividade do projeto.

Atividade da Prática Integradora 2

Objetivos principais:

- Obter a quantidade de dias que tivemos temperatura maior ou igual à 30°C;
- Obter a quantidade de dias que choveu na região selecionada.

Foi utilizado a média das médias diárias de temperatura mínima e máxima, sendo assim, não tivemos valores superiores à 30°C. Poderíamos também mudar nosso programa para pegar a máxima e mínima diária. Sem perda de generalidade criamos uma função que que dado um vetor de features, estabelecemos qual o valor alvo e filtramos apenas por valores maiores que o alvo, desse modo conseguimos filtrar temperaturas maiores que qualquer 30°C, 25°C ou qualquer outro. Para o nosso caso específico, utilizamos temperatura maior que 20°C.

A função anterior também foi utilizada para saber quais dias choveram em São Paulo, tomando o índice pluviométrico como sendo maior que zero. A Imagem A7 mostra a saída do arquivo.

```
victor@CPX-PZMSKGQ47FH ~/Node Estudo/Pratica_Integradora2$ nodejs pratica_integradora2.js
O número de dias com temperatura maior que 20 °C foi: 140 dias.
Choveu em 107 dias.
```

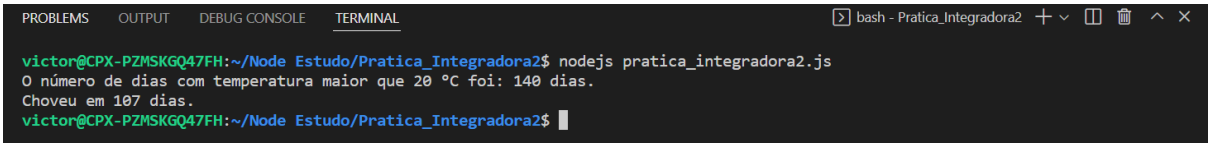


Imagem A7: Resultado do processamento da Prática Integradora 2.

Através do processamento anterior descobrimos que no ano de 2021 choveu em 107 dias no estado de São Paulo e que em 140 dias desse mesmo ano, tivemos temperatura maior que 20°C.

Considerações Finais

A realização dessa prática foi um desafio intenso, pois foi necessário achar o local correto para obtenção dos dados, coletá-los, fazer uma análise exploratória e processá-los, isso tudo com apenas 6h disponíveis (5 se desconsiderarmos 1h de almoço). Além das próprias tarefas da atividade, realizá-la sem bibliotecas externas para manipulação de csv e utilizando a linguagem JavaScript ao invés de Python tornou a dificuldade ainda maior, sendo necessário um bom planejamento para a sua finalização.

Essa Prática Integradora contribuiu bastante no meu desenvolvimento com a linguagem JavaScript, Gestão de Tempo e Programação em geral, me permitindo crescer como desenvolvedor.

Código de pré-processamento dos dados do INMET

```
// execucao:
// nodejs processar_dados_inmet.js
// requer a pasta "dados_mensais_inmet" no mesmo diretorio do programa
// a pasta "dados_mensais_inmet" contem os arquivos csv do inmet

/* referencias
leitura de arquivo: https://nodejs.dev/learn/reading-files-with-nodejs
print sem quebra de linha:
https://stackoverflow.com/questions/6157497/node-js-printing-to-console-
without-a-trailing-newline
concatenar dicionarios js:
https://stackoverflow.com/questions/43449788/how-do-i-merge-two-dictiona
ries-in-javascript
*/
const fs = require('fs')
const lista_arquivos = "lista_arquivos.txt" // txt contendo todos os
nomes dos arquivos do INMET coletados
const path_dados = "dados_mensais_inmet/" // path da pasta com os
arquivos do INMET
const ARQUIVO_SAIDA = "dados_meteorologicos_2021.csv"

// parametros de leitura do arquivo
DELIMITADOR_REGISTRO = '\n'
DELIMITADOR_COLUNA = ';'
CODIFICACAO = 'utf-8'

// cabecalho
DATA = 0; TEMP_MAX = 3; TEMP_MIN = 4; CHUVA = 18;
ROTULO = [DATA, TEMP_MAX, TEMP_MIN, CHUVA]

const nomes_arquivos_raw = fs.readFileSync(lista_arquivos, CODIFICACAO)
// leitura da lista de arquivos do INMET

const nomes_arquivos = nomes_arquivos_raw.split(DELIMITADOR_REGISTRO) //
obtem os nomes dos arquivos do INMET e coloca em um vetor

const dados_mensal = nomes_arquivos.map( mes => path_dados+mes) // path
+ nome_arquivo

// permite exibir um registro selecionando previamente os rotulos
function print_registro(registro, rotulo){
    let i = 0;
    for(i = 0; i< rotulo.length-1; i++){
```

```
        process.stdout.write(registro[rotulo[i]] + DELIMITADOR_COLUNA +
" ")
    }
    console.log(registro[rotulo[i]])
}

// dado uma lista de registros, exibe esses registros com a formatacao
do print_registro
function print_lista_registro(lista_registros, pos_a, pos_b, rotulo){
    for(i=pos_a; i<=pos_b; i++){
        print_registro(lista_registros[i], rotulo)
    }
}
// funcao para processar os dados mensais do INMET
function processar_dados_inmet(path_arquivo_mensal_inmet){
    arquivo_mes = fs.readFileSync(path_arquivo_mensal_inmet,
CODIFICACAO) // leitura do dado mensal do INMET

    registros_mes = arquivo_mes.split('\n') // arquivo mensal separado
em registros por linhas
    registros_mes = registros_mes.map( registro =>
registro.split(DELIMITADOR_COLUNA)) // registros em vetores

    return registros_mes
}

// remove as aspas duplas de cada coluna do registro oriundas de rotulo
function limpar_registro(registro, rotulo){
    for(i = 0; i < rotulo.length; i++){
        registro[rotulo[i]] = registro[rotulo[i]].replace(/"/g,"") //
remove aspas
        registro[rotulo[i]] = registro[rotulo[i]].replace(/,/g,".") //
troca o separador decimal
    }
}

// (desativado) agrupa registros em um dicionario por data
function agrupar_registros_por_data(registros_mes){

    dicionario_registros = {}

    registros_mes.forEach( registro => {
        limpar_registro(registro, ROTULO)
        if(dicionario_registros[registro[DATA]] === undefined){
            dicionario_registros[registro[DATA]] = [];
        }
```

```
        dicionario_registros[registro[DATA]].push(registro);
    })
    return dicionario_registros
}

function sumarizar_dicionario_registros(dicionario_registros){
    dicionario_sumarizado = {}

    for(chave in dicionario_registros){
        lista_registros = dicionario_registros[chave]
        registro_simplificado = {}
        registro_simplificado['TEMP_MAX'] = 0
        registro_simplificado['TEMP_MIN'] = 0
        registro_simplificado['CHUVA'] = 0
        num_registros = 0

        lista_registros.forEach( registro => {
            reg_temp_max = parseFloat(registro[TEMP_MAX]);
            reg_temp_min = parseFloat(registro[TEMP_MIN]);
            reg_chuva = parseFloat(registro[CHUVA]);

            if(isNaN(reg_temp_max) || isNaN(reg_temp_min) ||
isNaN(reg_chuva)){
                return
            }

            num_registros += 1
            registro_simplificado['TEMP_MAX'] += reg_temp_max;
            registro_simplificado['TEMP_MIN'] += reg_temp_min;
            registro_simplificado['CHUVA'] += reg_chuva;
        })

        registro_simplificado['TEMP_MAX'] /= num_registros;
        registro_simplificado['TEMP_MIN'] /= num_registros;
        registro_simplificado['CHUVA'] /= num_registros;
        dicionario_sumarizado[chave] = registro_simplificado
    }

    return dicionario_sumarizado
}

function salvar_dicionario_datas(dicionario_datas){
    dados_saida = ""
    count = 0
    for(chave in dicionario_datas){
        if(count == 0){
```

```
            count += 1
            continue
        }

        registro = dicionario_datas[chave]
        linha_saida = chave
        linha_saida += DELIMITADOR_COLUNA +
registro['TEMP_MAX'].toFixed(1)
        linha_saida += DELIMITADOR_COLUNA +
registro['TEMP_MIN'].toFixed(1)
        linha_saida += DELIMITADOR_COLUNA + registro['CHUVA'].toFixed(1)
        if(count != 365){
            linha_saida += "\n"
        }
        dados_saida += linha_saida
        count += 1
    }
    fs.writeFile(ARQUIVO_SAIDA, dados_saida, err => {
        if (err) {
          console.error(err);
        }
        // file written successfully
      });
}

// funcao principal do programa
function main(){
    // le todos os arquivos do inmet e os coloca em um array
    registros_mensais = []
    for(i = 0; i < dados_mensal.length; i++){
        registro_mes = processar_dados_inmet(dados_mensal[i])
        registros_mensais.push(registro_mes) // append
    }

    // transforma listas de registros em dicionarios de registros
agrupados por data, chave = data
    dicionario_geral = {}
    for(k = 0; k < registros_mensais.length; k++){
        dicionario_agrupado =
agrupar_registros_por_data(registros_mensais[k])
        dicionario_geral = Object.assign({}, dicionario_geral,
dicionario_agrupado)
    }

    dicionario_datas = sumarizar_dicionario_registros(dicionario_geral)
```

```
    salvar_dicionario_datas(dicionario_datas)

    console.log("Arquivo salvo com sucesso")
}

// ===== INICIO DA FUNCAO PRINCIPAL ===== //
main()
```

Código com o processamento final

```
const fs = require('fs')
const arquivo_entrada = "dados_meteorologicos_2021.csv"

DELIMITADOR_REGISTRO = '\n'
DELIMITADOR_COLUNA = ';'
CODIFICACAO = 'utf-8'

DATA = 0; TEMP_MAX = 1; TEMP_MIN = 2; CHUVA = 3;
ROTULO = [DATA, TEMP_MAX, TEMP_MIN, CHUVA]

const dados_raw = fs.readFileSync(arquivo_entrada, CODIFICACAO) //
leitura da lista de arquivos do INMET
lista_registros = dados_raw.split(DELIMITADOR_REGISTRO)
lista_registros = lista_registros.map( registro =>
registro.split(DELIMITADOR_COLUNA)) // registros em vetores

function filtrar_por_valor_maior(vetor_feature, valor_feature){
    vetor_filtrado = vetor_feature.filter(feature => feature >
valor_feature)
    return vetor_filtrado
}

// funcao principal
function main(){

    vetor_temperaturas = []
    vetor_chuva = []
    temperatura_filtro = 20 // no exemplo pediasse 30 graus, mas
utilizei temp_media diaria de SP como entrada
    filtro_chuva_mm = 0 // verifica quando choveu, indice (mm) > 0

    for(i = 0 ; i < lista_registros.length; i++){
        registro = lista_registros[i]
        reg_temp_max = parseFloat(registro[TEMP_MAX])
        reg_temp_min = parseFloat(registro[TEMP_MIN])
        temperatura_media =
parseFloat(((reg_temp_max+reg_temp_min)/2).toFixed(1))

        reg_chuva = parseFloat(parseFloat(registro[CHUVA]).toFixed(1))

        vetor_temperaturas.push(temperatura_media)
        vetor_chuva.push(reg_chuva)
    }
```

```
    vetor_temperaturas_filtradas =
filtrar_por_valor_maior(vetor_temperaturas, temperatura_filtro)
    vetor_chuva_filtrado = filtrar_por_valor_maior(vetor_chuva,
filtro_chuva_mm)

    console.log("O número de dias com temperatura maior que " +
                temperatura_filtro + " °C foi: " +
vetor_temperaturas_filtradas.length + " dias.")
    console.log("Choveu em " + vetor_chuva_filtrado.length + " dias.")

}

main()
```