

**CENTRO UNIVERSITÁRIO INTERNACIONAL UNINTER
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

REGINALDO GALLI DOS SANTOS – RU 3573100

**DESENVOLVIMENTO DE PÁGINA WEB COM DASHBOARD PARA EXIBIÇÃO
DOS DADOS DA VACINAÇÃO CONTRA COVID-19 EM ATIBAIA-SP**

Atibaia - SP

2022

REGINALDO GALLI DOS SANTOS – RU 3573100

**DESENVOLVIMENTO DE PÁGINA WEB COM DASHBOARD PARA EXIBIÇÃO
DOS DADOS DA VACINAÇÃO CONTRA COVID-19 EM ATIBAIA-SP**

Pesquisa apresentada ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas apresentado na disciplina de Atividade Extensionista II do Centro Universitário Internacional UNINTER.

Atibaia - SP

2022

SUMÁRIO

1.INTRODUÇÃO.....	3
2.DESENVOLVIMENTO.....	4
2.1.OBJETIVOS E JUSTIFICATIVA PARA A AÇÃO	4
3.METODOLOGIA.....	5
4.RESULTADOS.....	7
5.CRONOGRAMA.....	8
6.CONCLUSÃO.....	8
REFERÊNCIAS.....	9
APÊNDICE A - APP.PY.....	10
APÊNDICE B - STYLE.CSS.....	16

1 INTRODUÇÃO

Em janeiro de 2021 iniciou-se a vacinação contra Covid-19 em todo o território brasileiro. Todos os dados referentes à vacinação no país estão sendo registrados, possibilitando o acompanhamento diário da vacinação no país.

A disponibilização dos dados oferece aos governantes a possibilidade de análises detalhadas que podem servir de base para tomadas de decisão.

Os dados, porém, existem em grande volume e são difíceis de serem analisados, sendo necessário convertê-los em informações. Conforme Baltzan e Philips (2012), os dados são fatos brutos que descrevem um evento, enquanto a informação é o produto da conversão dos dados em um contexto significativo e útil.

A presente atividade tem o propósito de desenvolver e documentar o desenvolvimento de um dashboard online para exibição dos dados da vacinação contra Covid-19 no município de Atibaia-SP.

2 DESENVOLVIMENTO

O objetivo a ser atingido nesta atividade é dar seguimento ao projeto iniciado na [Atividade Extensionista I](#), onde visando atender o objetivo de desenvolvimento sustentável da ONU "Saúde e Bem-Estar", foi realizado um estudo sobre análise de dados da vacinação contra covid-19 no município de Atibaia – SP:

Nesta primeira etapa, os gráficos foram gerados, possibilitando uma análise dos dados disponíveis, porém ainda dentro do IDE Python.

2.1 OBJETIVOS E JUSTIFICATIVA PARA A AÇÃO

Dando continuidade ao projeto já iniciado e visando atender os princípios do Desenho Universal, está sendo proposto o desenvolvimento de um dashboard com exibição em uma página web tornando a visualização dos gráficos da vacinação mais intuitivos e acessíveis a qualquer pessoa, independentemente do conhecimento do usuário em programação.

Para desenvolvimento da atividade será utilizada a linguagem de programação Python em conjunto com as bibliotecas Pandas e Dash.

3 METODOLOGIA

Dando seguimento ao trabalho desenvolvido na Atividade Extensionista I, iniciou-se o trabalho de obtenção dos dados atualizados da vacinação no site Open Data Sus (<https://opendatasus.saude.gov.br/dataset/covid-19-vacinacao>). No site são disponibilizados arquivos do tipo CSV, contendo os dados da vacinação de cada estado.

Optou-se então por continuar utilizando a plataforma de edição de códigos Colaboratory, ou “Colab” do Google nesta etapa de filtragem dos dados. A vantagem desta ferramenta é a integração ao Google Drive, permitindo assim, realizar a leitura de arquivos armazenados na nuvem, eliminando a necessidade do download dos arquivos CSV que contém em torno de 60 GB de dados.

Figura 1: Google Colaboratory

```

# Importa a biblioteca pandas como pd
import pandas as pd

# Importa a biblioteca matplotlib como plt
import matplotlib.pyplot as plt

from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).

# baixa baixa um arquivo csv da internet direto para a pasta no google drive
!wget -P "/content/drive/MyDrive/Colab Datasets/vacina_sp" https://s3.sa-east-1.amazonaws.com/ckan.saude.gov.br/SIPNI/COVID/uf/ufX3DSP/part-00000-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv

# baixa baixa um arquivo csv da internet direto para a pasta no google drive
!wget -P "/content/drive/MyDrive/Colab Datasets/vacina_sp2" https://s3.sa-east-1.amazonaws.com/ckan.saude.gov.br/SIPNI/COVID/uf/ufX3DSP/part-00001-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv

# baixa baixa um arquivo csv da internet direto para a pasta no google drive
!wget -P "/content/drive/MyDrive/Colab Datasets/vacina_sp3" https://s3.sa-east-1.amazonaws.com/ckan.saude.gov.br/SIPNI/COVID/uf/ufX3DSP/part-00002-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv

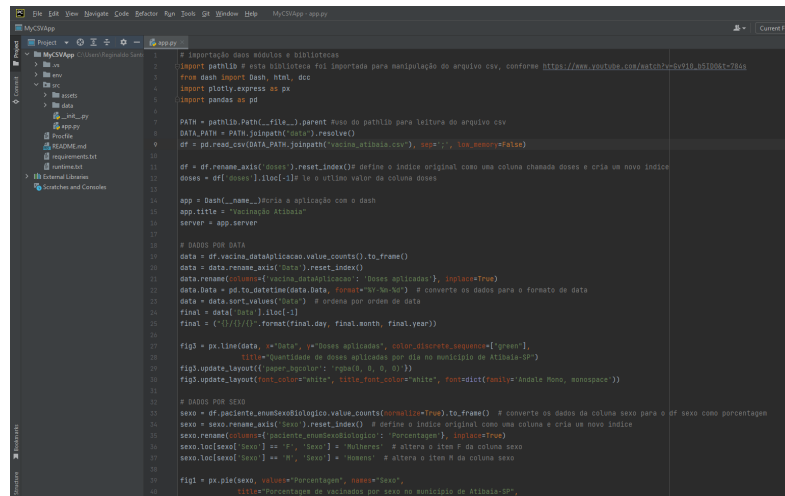
# este código lê o arquivo csv de lotes de 10000 linhas e vai salvando na lista temporária filtro_df. Ao final salva na variável df
filtro_df = []
for chunk in pd.read_csv('/content/drive/MyDrive/Colab Datasets/vacina_sp/part-00000-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv', sep=';', chunksize=10000):
    temp_df = chunk.loc[chunk['estabelecimento_municipio_nome'] == 'ATIBAIA']
    filtro_df.append(temp_df)
for chunk in pd.read_csv('/content/drive/MyDrive/Colab Datasets/vacina_sp/part-00001-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv', sep=';', chunksize=10000):
    temp_df = chunk.loc[chunk['estabelecimento_municipio_nome'] == 'ATIBAIA']
    filtro_df.append(temp_df)
for chunk in pd.read_csv('/content/drive/MyDrive/Colab Datasets/vacina_sp/part-00002-b8f5fec1-909d-4f70-a59f-b9343347286e.c000.csv', sep=';', chunksize=10000):
    temp_df = chunk.loc[chunk['estabelecimento_municipio_nome'] == 'ATIBAIA']
    filtro_df.append(temp_df)
df = pd.concat(filtro_df, ignore_index=True)
  
```

Fonte: O autor (2022)

Após realizar a filtragem dos dados que seriam realmente utilizados, obteve-se um arquivo CSV com aproximadamente 35MB. Neste ponto, realizou-se o download do arquivo filtrado e iniciou-se o desenvolvimento com a IDE Pycharm.

Após os estudos iniciais definiu-se que para criar a aplicação web seria utilizada a biblioteca Dash que em conjunto com a biblioteca Pandas, é capaz de gerar os gráficos para análise de dados e carregá-los em uma página web, inicialmente hospedada em um servidor local (localhost).

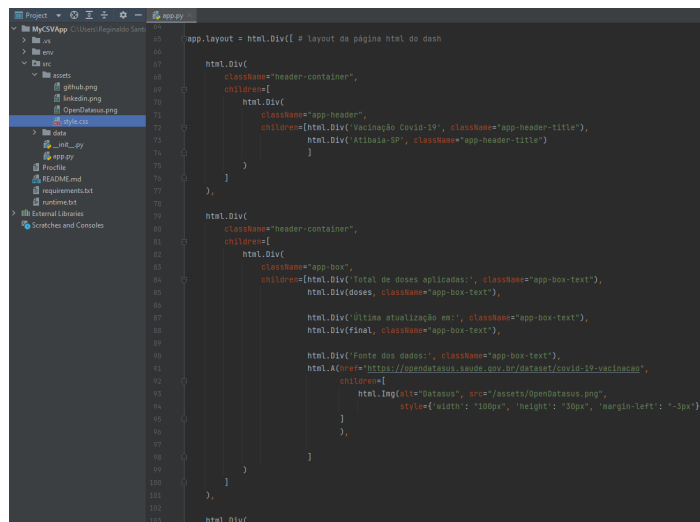
Figura 2: Desenvolvimento na IDE Pycharm



Fonte: O autor (2022)

Uma particularidade da biblioteca Dash é que ela utiliza as linguagens HTML e CSS com uma sintaxe própria para realizar a criação e layout da página web.

Figura 3: Utilização de HTML dentro do Python.



Fonte: O autor (2022)

4 RESULTADOS

Como citado anteriormente, durante o período de desenvolvimento a página web era executada em um servidor na máquina local.

Figura 4: Página web sendo carregada no servidor local.

```

C:\Users\Reginaldo Santos\AppData\Local\Programs\Python\Python310\python.exe "C:/Users/Reginaldo Santos/Meus Documentos/dashVacina/hyCSVApp/src/app.py"
Dash is running on http://127.0.0.1:8050/

* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8050
Press CTRL+C to quit

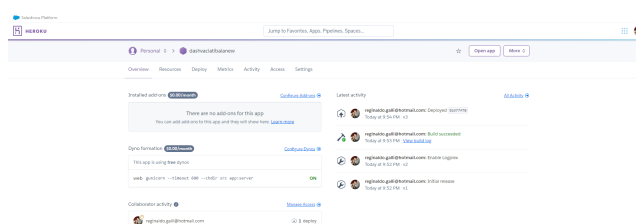
```

Fonte: O autor (2022)

Após a conclusão do desenvolvimento, iniciou-se a busca por uma solução online para hospedar a página desenvolvida.

Decidiu-se então hospedar a aplicação web na plataforma de hospedagem gratuita Heroku com o auxílio da biblioteca dashtools.

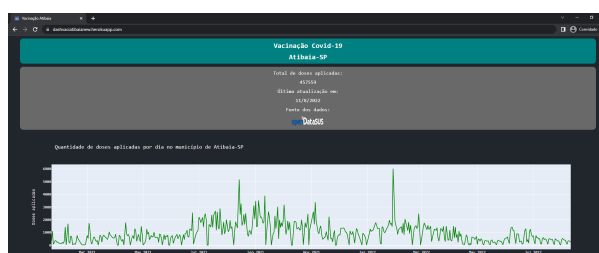
Figura 5: Página da plataforma Heroku com a aplicação web carregada.



Fonte: O autor (2022).

Através do trabalho desenvolvido o objetivo inicial foi atingido, podendo ser observado na imagem abaixo um dashboard online com os dados da vacinação contra Covid-19 no município de Atibaia-SP. O dashboard pode ser acessado através do endereço: <https://dashvaciatibaianew.herokuapp.com/>

Figura 6: Dashboard online - parte superior



Fonte: O autor (2022).

5 CRONOGRAMA

Cronograma 1

Descrição	Data Inicial	Data Final
Escolha do tema e elaboração da proposta de atividade	06/06/2022	10/07/2022
Entrega e espera da validação da proposta	11/07/2022	27/07/2022
Pesquisa e planejamento para execução	28/07/2022	08/08/2022
Desenvolvimento da aplicação prática conforme proposta	09/08/2022	20/08/2022
Elaboração do relatório	15/08/2022	22/08/2022
Revisão e finalização da atividade extensionista II	23/08/2022	24/08/2022
Entrega da atividade extensionista II	25/08/2022	

Fonte: O autor (2022)

6 CONCLUSÃO

Dando continuidade ao projeto desenvolvido na Atividade Extensionista I, onde foram gerados gráficos referentes à vacinação contra Covid-19 no município de Atibaia-SP, utilizando a biblioteca Pandas.

Nesta segunda etapa foi desenvolvida uma aplicação web possibilitando o acesso aos mesmos gráficos por parte de qualquer pessoa com acesso à internet, contribuindo assim com os interesses da sociedade sobre este assunto.

O desenvolvimento da Atividade Extensionista II utilizou os recursos da biblioteca Dash aliados a técnicas de desenvolvimento de página web.

Os códigos desenvolvidos foram disponibilizados no endereço: <https://github.com/reginaldogalli/dashvacinacao>

Concluindo, pode-se dizer que o estudo realizado reforçou os conhecimentos adquiridos durante o curso possibilitando aplicar na prática diversos conceitos adquiridos durante o conteúdo teórico do curso.

REFERÊNCIAS

CORRÊA, Eduardo. **Pandas Python**: Data Wrangling para Ciência de Dados. Rio de Janeiro: Casa do Código, 2020.

BANIN, Sérgio Luiz. **Python 3**: conceitos e aplicações; uma abordagem didática. São Paulo: Érica, 2018.

COLABORATORY. **Perguntas frequentes**. Disponível em: (<https://research.google.com/colaboratory/intl/pt-BR/faq.html>). Acesso em: 11 jul. 2022.

PANDAS. **About Pandas**. Disponível em: (<https://pandas.pydata.org/about/index.html>). Acesso em: 23 ago. 2022.

DASH PYTHON, **User guide**. Disponível em: (<https://dash.plotly.com/>). Acesso em: 11 jul. 2022.

HASHTAG TREINAMENTOS, **Dashboards em Python com Dash e Plotly**. Disponível em: (<https://www.hashtagtreinamentos.com/dashboards-em-python>). Acesso em: 11 jul. 2022.

Hosaack Andrew, **DashTools Documentation - v1.8.2**. Disponível em: (<https://dash-tools.readthedocs.io/en/latest/index.html>). Acesso em: 23/08/2022

Charming Data, **Easiest Way to Deploy a Dash App to the Web**. Disponível em: (https://www.youtube.com/watch?v=Gv910_b5ID0&t=346s). Acesso em: 23/08/2022

BALTZAN, P; PHILLIPS, A. **Sistemas de informação**. Porto Alegre: AMGH, 2012.

APÊNDICE A - app.py

```
# importação dos módulos e bibliotecas

import pathlib # esta biblioteca foi importada para
manipulação do arquivo csv, conforme
https://www.youtube.com/watch?v=Gv910\_b5ID0&t=784s

from dash import Dash, html, dcc

import plotly.express as px

import pandas as pd

PATH = pathlib.Path(__file__).parent #uso do pathlib para
leitura do arquivo csv

DATA_PATH = PATH.joinpath("data").resolve()

df = pd.read_csv(DATA_PATH.joinpath("vacina_atibaia.csv"),
sep=';', low_memory=False)

df = df.rename_axis('doses').reset_index()# define o indice
original como uma coluna chamada doses e cria um novo indice

doses = df['doses'].iloc[-1]# le o ultimo valor da coluna
doses

app = Dash(__name__)#cria a aplicação com o dash

app.title = "Vacinação Atibaia"

server = app.server

# DADOS POR DATA

data = df.vacina_dataAplicacao.value_counts().to_frame()

data = data.rename_axis('Data').reset_index()

data.rename(columns={'vacina_dataAplicacao': 'Doses
aplicadas'}, inplace=True)

data.Data = pd.to_datetime(data.Data, format="%Y-%m-%d") #
converte os dados para o formato de data
```

```

data = data.sort_values("Data") # ordena por ordem de data

final = data['Data'].iloc[-1]

final = ("{}/{}/{}".format(final.day, final.month,
final.year))

fig3 = px.line(data, x="Data", y="Doses aplicadas",
color_discrete_sequence=["green"],

                title="Quantidade de doses aplicadas por dia no
município de Atibaia-SP")

fig3.update_layout({'paper_bgcolor': 'rgba(0, 0, 0, 0)'})

fig3.update_layout(font_color="white",
title_font_color="white", font=dict(family='Andale Mono,
monospace'))

# DADOS POR SEXO

sexo =
df.paciente_enumSexoBiologico.value_counts(normalize=True).to_
frame() # converte os dados da coluna sexo para o df sexo
como porcentagem

sexo = sexo.rename_axis('Sexo').reset_index() # define o
índice original como uma coluna e cria um novo índice

sexo.rename(columns={'paciente_enumSexoBiologico':
'Porcentagem'}, inplace=True)

sexo.loc[sexo['Sexo'] == 'F', 'Sexo'] = 'Mulheres' # altera o
item F da coluna sexo

sexo.loc[sexo['Sexo'] == 'M', 'Sexo'] = 'Homens' # altera o
item M da coluna sexo

fig1 = px.pie(sexo, values="Porcentagem", names="Sexo",

                title="Porcentagem de vacinados por sexo no
município de Atibaia-SP",

                color="Sexo", color_discrete_map={'Homens':
'steelblue', 'Mulheres': 'indianred'}, hole=.5)

```

```

fig1.update_layout({'paper_bgcolor': 'rgba(0, 0, 0, 0)'})

fig1.update_layout(font_color="white",
title_font_color="white", font=dict(family='Andale Mono,
monospace'))

# DADOS POR NOME DA VACINA

vacina = df.vacina_nome.value_counts().to_frame()

vacina = vacina.rename_axis('Vacina/Fabricante').reset_index()

vacina.rename(columns={'vacina_nome': 'Quantidade'},
inplace=True)

fig2 = px.pie(vacina, values="Quantidade",
names="Vacina/Fabricante",

               title="Porcentagem de doses aplicadas por
fabricante no município de Atibaia-SP", hole=.3)

fig2.update_layout({'paper_bgcolor': 'rgba(0, 0, 0, 0)'})

fig2.update_layout(font_color="white",
title_font_color="white", font=dict(family='Andale Mono,
monospace'))

# DADOS POR UBS

posto = df.estalecimento_noFantasia.value_counts().to_frame()

posto = posto.rename_axis('Posto').reset_index()

posto.rename(columns={'estalecimento_noFantasia': 'Doses'},
inplace=True)

fig4 = px.bar(posto, y='Posto', x='Doses',
color_discrete_sequence=["darkslategray"],

               title="Quantidade de doses aplicadas por UBS no
município de Atibaia-SP")

fig4.update_layout({'paper_bgcolor': 'rgba(0, 0, 0, 0)'})

```

```

fig4.update_layout(font_color="white",
title_font_color="white", font=dict(family='Andale Mono,
monospace'))

app.layout = html.Div([ # layout da página html do dash

    html.Div(

        className="header-container",

        children=[

            html.Div(

                className="app-header",

                children=[html.Div('Vacinação Covid-19',
className="app-header-title"),

                        html.Div('Atibaia-SP',
className="app-header-title")

                ]

            )

        ]

    ),

    html.Div(

        className="header-container",

        children=[

            html.Div(

                className="app-box",

                children=[html.Div('Total de doses aplicadas:',
className="app-box-text"),

                        html.Div(doses,
className="app-box-text"),

```

```

        html.Div('Última atualização em:',
className="app-box-text"),

        html.Div(final,
className="app-box-text"),

        html.Div('Fonte dos dados:',
className="app-box-text"),

html.A(href="https://opendatasus.saude.gov.br/dataset/covid-19-
-vacinacao",

        children=[

            html.Img(alt="Datasus",
src="/assets/OpenDatusus.png",

                style={'width':
"100px", 'height': "30px", 'margin-left': "-3px"})

        ],

    ),

    ],

)

],

),

html.Div(

    dcc.Graph( # Gráfico 1

        id="sexo",

        figure=fig3

    )

),

```

```
html.Div(  
    dcc.Graph( # Gráfico 2  
        id="fabricante",  
        figure=fig2  
    )  
,  
html.Div(  
    dcc.Graph( # Gráfico 3  
        id="data",  
        figure=fig4  
    )  
,  
html.Div(  
    dcc.Graph( # Gráfico 4  
        id="ubs",  
        figure=fig1  
    )  
,  
html.Footer(  
    html.Div(  
        children=[  
            html.A("Desenvolvido por Reginaldo G. Santos -  
2022"),
```



```

        html.A(href="https://github.com/reginaldogalli",
                children=[
                    html.Img(alt="Github",
src="/assets/github.png")
                ]
            ),

html.A(href="https://www.linkedin.com/in/reginaldogalli/",
        children=[
            html.Img(alt="Linkedin",
src="/assets/linkedin.png")
        ]
    )

    )

    ])

if __name__ == '__main__':
    app.run_server(debug=False)

```

APÊNDICE B - style.css

```

body {

    font-family: Andale Mono, monospace;

    background-color: #21262d

}

div {

    margin: 10px;

```

```
        border-radius: 10px
    }

    footer {

        background-color: teal;

        color: white;

        text-align: center;

        border-radius: 5px

    }

    a {

        color: white

    }

    img{

        width:15px;

        height:15px;

        border-radius:5px;

        margin-left: 15px;

        margin-top: 3px;

    }

    .app-box-text{

        text-align:center;

        color:white

    }

    .app-header {

        width: 1900px;
```

```
    height: 80px;

    background-color: teal;

    position: relative;

    border-radius: 10px;

    margin: 0px
}

.app-box {

    width: 1900px;

    height: 200px;

    background-color: dimgray;

    position: relative;

    border-radius: 10px;

    margin: 0px

}

.app-header-title {

    font-size: 22px;

    font-weight: bold;

    text-align: center;

    color: white;

}

.header-container {

    display: flex;

    flex-direction: row;

    justify-content: center;
```

```
        text-align:center
    }

    .container {

        display: flex;

        flex-direction: row;

        justify-content: center;

        padding-top: 250px;

        margin-bottom: 100px;

    }

    .smcontainer {

        background-color: #0e4c5614;

        width: 1000px;

        height: 420px;

        position: relative;

        border-radius: 30px

    }

    .box { height: 50px;

        width: 400px;

        position: relative;

        left: 50%;

        top: 50%;

        transform: translate(-120%, 40%);

        background-color: yellow

    }
```