

SADC-DF

Heloise K. P. Silva, José A .A. F. Filho, Leandro M. Rodrigues, Nínive H. H. Silva

Instituto Federal de Educação, Ciência e Tecnologia de Brasília (IFB) - Campus
Taguatinga

Ciência da Computação

Resumo. *Com o avanço da pandemia percebemos a importância de acompanhar os dados sobre o avanço do vírus no Brasil e no mundo. O objetivo do nosso artigo é realizar o tratamento e a análise de dados da Covid-19 no Brasil e disponibilizar essas análises em um site utilizando principalmente as seguintes tecnologias: Django, Python e MongoDB.*

Abstract. *With the sending of the pandemic. The purpose of our article is to carry out the treatment and analysis of Covid data in Brazil and provide them with the following technologies on a website using mainly: Django, Python and MongoDB.*

1. Introdução

Desde o início da pandemia em 11 de março de 2020 o governo tem estudado os dados para tomar medidas como lockdown com o objetivo de diminuir a transmissão do vírus SARS-CoV-2. Os dados podem ser acessados pelo site do SUS (Sistema Único de Saúde). O projeto AADC-DF tem como objetivo realizar o tratamento e a análise dos dados da vacinação de COVID-19 no Distrito Federal e criar um site que possa disponibilizar a análise desses dados de forma intuitiva para os usuários.

2. Projeto SACDC-DF

O SACDC-DF é um projeto web para visualização de dados de vacinados no DF, com o intuito de entender como está a situação de vacinados no distrito federal, classificando a quantidade de vacinados por gênero, etnia, comorbidade, faixa etária e por profissionais da saúde. No desenvolvimento do projeto SADC-DF foi realizado um levantamento do problema, levantamento de

requisitos para a elaboração do projeto. Nas etapas requisitos foram realizadas pesquisas de qual linguagem seria mais eficiente e prática para o desenvolvimento do projeto, qual paradigma seria mais adequado para a linguagem e para o projeto. O projeto AADC-DF utilizou o framework Django, e a linguagem python que é a linguagem oficial do django para desenvolvimento de aplicações web. O paradigma funcional, um paradigma que possibilita o uso de funções, que podem ser reutilizadas e facilita a manutenção e a organização do projeto.

3. Paradigmas de Programação

Paradigmas de programação é a modelagem do código, como a estrutura do código vai ser aplicada na linguagem de programação. Cada linguagem tem sua regra de paradigma e como deve ser seguida, cada projeto ou problema que vai definir qual paradigma é mais adequado para resolver ou desenvolver um projeto. Cada paradigma tem sua vantagem e desvantagem, e cada um consegue suprir determinada necessidade no desenvolvimento do código, pode se utilizar de vários paradigmas aplicados a uma linguagem, que são as linguagens chamadas multiparadigmas.

Os paradigmas de programação são muito importantes para o desenvolvimento de algum projeto, cada linguagem de programação está associada a algum paradigma, entender qual linguagem e qual paradigma de programação é mais adequado para resolver determinado problema, facilita na resolução de um problema ou na construção de um projeto. O paradigma aplicado da maneira correta traz uma eficiência e um melhor desenvolvimento do projeto. Conhecer-los também ajuda a ter um melhor uso da linguagem escolhida e uma confiabilidade. Uma escolha errada do paradigma pode trazer um mau uso da linguagem e não ter uma eficiência da aplicabilidade, pois para um projeto está bem estruturado precisa ter o paradigma definido corretamente e utilizando a linguagem adequada. Projetos que utilizam paradigmas e linguagens adequadas

para cada tipo de problema ou projeto que são desenvolvidos, têm uma aplicabilidade e confiabilidade mais eficiente.

Para o desenvolvimento deste projeto utilizamos a linguagem Python e os paradigmas funcional e imperativo. A linguagem Python também suporta o paradigma orientado a objetos, mas esse paradigma não foi utilizado no projeto. O paradigma imperativo trabalha com uma sequência de instruções que devem ser passadas ao computador na sequência em que devem ser executadas. O paradigma funcional utiliza o uso de funções que são pequenos trechos de códigos que podem ser executados na sequência em que são chamados.

4. Framework Django

Este projeto foi desenvolvido utilizando o Django. Django é um framework para desenvolvimento web escrito na linguagem Python que utiliza o padrão modelo-template-view. O Django é utilizado para criar a interface visual onde os gráficos da vacinação seriam renderizados, Django é uma framework bastante popular, pois é um framework que facilita o desenvolvimento web, seu padrão de modelo facilita a integração com o banco de dados e a criação de páginas para a visualização na web.

5. Linguagem Python

Utilizamos a linguagem de programação Python para desenvolver o projeto. Python é uma linguagem multiparadigma e que possibilita codificar com menos linhas de códigos, pois sua sintaxe é muito simples. Para a construção dos gráficos foi utilizada a biblioteca Pandas, uma biblioteca da linguagem Python que tem como foco fornecer estruturas de dados com alto desempenho e ferramentas para a análise de dados, possibilitando selecionar tabelas e limpar dados.

6. Paradigma Funcional

Um dos paradigmas utilizado no projeto foi o paradigma funcional, esse paradigma permite a criação de funções e codificação mais complexa. Esse paradigma permite dividir as etapas do desenvolvimento do código em pequenos trechos de códigos que podem ser reutilizados, dependendo apenas dos dados de entrada e saída, sem ter alteração do código, as funções que são operações dentro do código podem ser chamadas a qualquer momento, também permite uma manutenção melhor de código. No projeto SADC-DF o paradigma funcional foi utilizado para criar as funções para cada gráfico receber seu código de limpeza e criação do gráfico.

7. Paradigma Imperativo

O paradigma imperativo utilizado é um dos mais antigos, ele utiliza de uma sequência lógica para executar os códigos, diferente do funcional que não segue uma sequência, podendo ter chamadas de funções em diferentes estados no código. O paradigma imperativo mantém suas variáveis e valores atribuídos a elas, o valor atribuído sempre permanece o mesmo na execução do código, o paradigma imperativo na compilação segue a sequência em que o código foi escrito.

É utilizado em algoritmos que e projetos em que a seguem uma lógica sequencial no seu desenvolvimento. No projeto SADC-DF foi utilizado na criação dos gráficos, pois cada gráfico foi criado em uma ordem pré definida para sua visualização.

8. Metodologia

Utilizando a linguagem Python para criar as funções que são responsáveis pela chamada dos gráficos. Na linguagem Python foi utilizado o paradigma funcional, um paradigma que possibilita a criação de funções que podem ser modificadas e reutilizadas, facilitando a construção e a manutenção do código.

Cada função era responsável pelo código do gráfico, os gráficos foram desenvolvidos usando outra biblioteca do Python e Matplotlib, uma biblioteca que possibilita a visualização de dados através de gráficos. As etapas de criação de função foram criadas usando o Python, o tratamento de dados utilizando a biblioteca pandas, e a visualização utilizando a biblioteca Matplotlib. O Python é uma das linguagens multiparadigma mais utilizadas na área de análise e ciência de dados e por isso foi a linguagem escolhida para o desenvolvimento do projeto SADC-DF. (FIGURA 1)

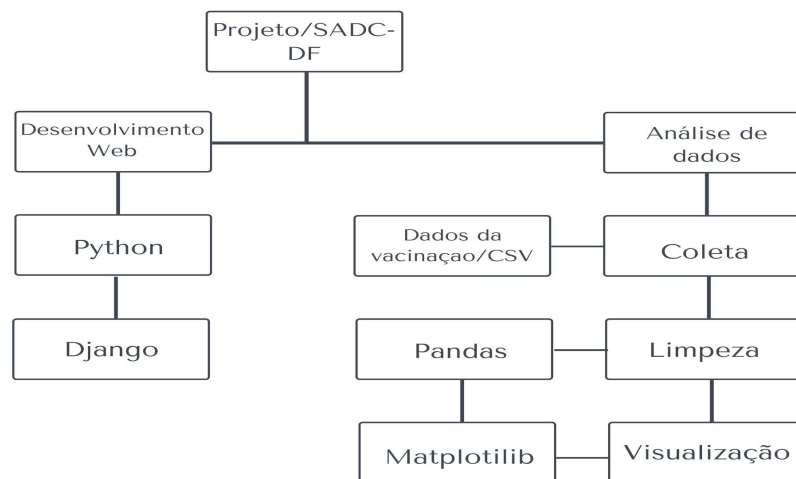


Figura 1 - Diagrama do projeto SADC-DF

O projeto teve a criação de views e rotas para a criação da página inicial, para a visualização dos gráficos e foi criado uma página chamada analytics que era responsável pelas funções das criações dos gráficos. O Django utilizando a views e rotas acessa o arquivo analytics e pela página inicial eram mostrados os gráficos das vacinas no DF.

A organização do projeto foi através da plataforma Trello, uma plataforma bastante utilizada para organização e divisão de tarefas envolvendo planejamento e desenvolvimento de software, onde as tarefas são divididas em

cartões em etapas de desenvolvimento e conclusão de cada tarefa, e cada etapa do desenvolvimento do projeto é associada ao responsável para aquela tarefa. A equipe toda consegue acompanhar as etapas de todos os envolvidos e também colabora na conclusão.

9. Conclusão

Podemos concluir que a utilização do framework Django foi de extrema importância para a parte web do projeto e facilitou a integração com o banco de dados possibilitando que o foco fosse direcionado para o tratamento e análise dos dados da Covid-19 no DF. A biblioteca Pandas do Python otimizou o tratamento e análise dos dados tendo como principal característica a facilidade em lidar com os dados, a biblioteca Matplotlib atendeu aos requisitos e nos possibilitou criar os gráficos de forma rápida e simples.

10. Referências

Sistema Único de Saúde. SUS. Disponível em: < <https://opendatasus.saude.gov.br/> > Acesso em: 3 de fev. de 2022.

Django. Disponível em: < <https://www.djangoproject.com/> > Acesso em: 17 de dez. de 2021.

Python. Disponível em: < <https://docs.python.org/pt-br/3//> > Acesso em: 06 de jan. de 2021.

Medium Disponível em: :< <https://sergiocosta.medium.com/paradigma-funcional-3194924a8d20/> > Acesso em: 06 de jan. de 2022.

Medium Disponível em: :< <https://medium.com/zeroeumas/paradigmas-de-programacao-845ecf7b4aaf/> > Acesso em: 06 de jan. de 2022.

Medium Disponível em: :<
<https://medium.com/@jt93synyster/como-os-paradigmas-de-programa%C3%A7%C3%A3o-influenciam-no-desenvolvimento-de-programas-523ebd5af7db/> >
Acesso em: 06 de jan. de 2022.